

**PROBIT BASED METHODS IN TRAFFIC
ASSIGNMENT AND DISCRETE CHOICE MODELLING**

Andrea Rosa

A thesis submitted in partial fulfilment of the requirements
of Napier University for the degree of Doctor of Philosophy

School of the Built Environment

Napier University

April 2003

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Mike Maher, for his advice, help and patience during the production of this thesis.

Many thanks should also go to my second supervisor Dr. Wafaa Saleh and to the people in the School of the Built Environment.

I also would like to thank all those who have been close to me, my parents and my friends, who helped me and supported me during these years.

My parents should be thanked, amongst many reasons, for their moral and financial support throughout the time spent on the thesis and before and for getting used to their only son living at such a long distance away.

A really heartfelt thank you must go to my partner Melanie, whose love and support have been invaluable.

ABSTRACT

Many phenomena studied in transportation are the result of choices made by individuals, such as the choice of route through a road network or the choice of mode to travel with, and are therefore modelled using choice models. This thesis addresses some aspects of the application of the multinomial probit (MNP) choice model, a well known model that, although theoretically sound, is not widely used as it is difficult to calculate since its choice function cannot be written in closed form. In particular, this thesis investigates the solution of the multinomial probit choice function by means of analytical approximations (that allow us to overcome the computational difficulties of the model) and addresses the application of MNP analytical approximations to the solution of the traffic assignment problem and some of its extensions. The use of MNP approximations to solve the model calibration problem is also addressed.

The feasibility of solving the multinomial probit choice function by analytical approximation, rather than by simulation as typically done in applications, is explored by reviewing a number of approximation methods, some already used in transportation and some borrowed from other disciplines. Their precision is investigated by comparing their results and those obtained by numerical integration on a large number of test cases designed to replicate choice situations encountered in traffic assignment. Their computational cost is assessed by comparing the calculation times needed to solve the same set of test cases.

The application of analytical approximations to solve the Stochastic User Equilibrium (SUE) traffic assignment problem is studied by describing and testing solution algorithms based on solving analytically the choice model and on the link-based formulation of the problem due to Sheffi and Powell (1982) but keeping track of the solution both in terms of path and of link flows. A first series of algorithms considered use the search direction of the method of successive averages (MSA) and are based on those developed by Maher and Hughes (1997a). A second series of algorithms employ alternative search vectors devised by considering the MSA search direction as a preconditioned steepest descent direction. Numerical tests, carried out on fixed sets of paths using two of the MNP approximations investigated, show that

some of the algorithms of the second series outperform consistently the previous ones and those in the literature.

The inclusion of multiple classes of users (MUC) with different perceptions of the network costs is treated using the formulation put forward by Daganzo (1982) and extending the algorithms investigated for the SUE case. The cases of SUE with elastic demand and with both multiple user classes and elastic demand are studied proposing formulations of the problems that are extensions of those for SUE and MUC SUE and allow us to develop efficient algorithms that are extensions of those proposed for SUE.

Finally, the solution of the multinomial probit calibration problem is investigated by describing and applying a method to obtain analytical derivatives of the log-likelihood function by exploiting the structure of the approximation employed. The work reported focuses on the approximation of Mendell and Elston but the calculation structure suggested can be used also with other approximations. The efficiency of the approach proposed is tested by comparing the computational effort it requires against that required using numerical derivatives.

CONTENTS

1. INTRODUCTION.....	1
2. RANDOM UTILITY DISCRETE CHOICE MODELS IN TRANSPORTATION.....	7
2.1 Introduction.....	7
2.2 The Random Utility Framework.....	7
2.3 A Survey of Random Utility Discrete Choice Models.....	10
2.3.1 Introduction.....	10
2.3.2 The Deterministic Model.....	12
2.3.3 Stochastic Models.....	12
2.3.3.1 The Probit Model.....	12
2.3.3.2 The Multinomial Logit Model.....	15
2.3.3.3 The Nested Logit Model.....	17
2.3.3.4 The Generalised Extreme Value Model.....	19
2.3.3.5 The Cross-Nested Logit Model.....	20
2.3.3.6 The Paired Combinatorial Logit Model.....	20
2.3.3.7 The C-Logit Model.....	21
2.3.3.8 The Mixed Logit Model.....	22
2.3.4 Discussion.....	23
2.4 Using Discrete Choice Models.....	25

2.4.1	Introduction.....	25
2.4.2	Choice Set Definition and Specification of a Choice Model.....	25
2.4.3	Calibration of Discrete Choice Models.....	28
2.4.4	Validation of Discrete Choice Models.....	29
2.4.5	Simple Forecasting and Equilibrium Forecasting.....	29
2.4.6	Aggregation of Discrete Choice Model Results	30
2.5	Conclusions.....	31
3.	METHODS FOR THE EVALUATION OF THE MULTINOMIAL PROBIT CHOICE FUNCTION.....	33
3.1	Introduction.....	33
3.2	The Multinomial Probit Choice Function.....	33
3.3	Calculation of the MNP Choice Function by Simulation	42
3.4	Calculation of the MNP Choice Function by Numerical Integration.....	46
3.5	Calculation of the MNP Choice Function Using Analytical Approximations	50
3.5.1	Introduction.....	50
3.5.2	The Approximation of Clark.....	51
3.5.3	The Approximation of Mendell and Elston	55
3.5.4	The Separated Split Approximation of Langdon.....	57
3.5.5	The Approximations of the FOMN Group	61
3.5.6	The Taylor Approximation	67

3.5.7	The Solow-Joe Approximation.....	71
3.5.8	The Extended Approximation of Joe.....	74
3.6	Investigations on Accuracy and Computational Cost of Approximations for Solving the MNP Choice Function	76
3.6.1	Introduction.....	76
3.6.2	Methodology.....	77
3.6.3	Test Networks	79
3.6.4	Results on Approximation Accuracy.....	86
3.6.4.1	Accuracy of the Simple Clark Approximations.....	86
3.6.4.2	Accuracy of the Improved Clark Approximation	93
3.6.4.3	Accuracy of the Mendell-Elston Approximation.....	102
3.6.4.4	Accuracy of the Separated Split Approximation of Langdon.....	115
3.6.4.5	Accuracy of the FOMN Family of Approximations.....	122
3.6.4.6	Accuracy of the Taylor Series Approximations.....	136
3.6.4.7	Accuracy of the Solow-Joe Approximation.....	145
3.6.4.8	Accuracy of the Extended Joe Approximation	155
3.6.5	Summary and Comparison of Accuracy and Calculation Time Results.....	164
3.7	Issues in the Practical Application of the MNP Choice Model	182
3.7.1	Semidefinite Covariance Matrices.....	182
3.7.2	The Evaluation of the Satisfaction Function.....	185
3.7.2.1	Introduction.....	185

3.7.2.2	Evaluation of the Satisfaction Using the Approximation of Clark	186
3.7.2.3	Evaluation of the Satisfaction Using the Approximation of Langdon	187
3.7.2.4	Evaluation of the Satisfaction Using the Normal Approximation.....	187
3.8	Conclusions.....	189
4.	MULTINOMIAL PROBIT STOCHASTIC USER EQUILIBRIUM PATH-BASED TRAFFIC ASSIGNMENT	193
4.1	Introduction.....	193
4.2	The Traffic Assignment Problem.....	193
4.2.1	Generalities	193
4.2.2	A Review of Traffic Assignment Models in the Literature	198
4.2.2.1	A Discussion on the Specification of Path Sets.....	198
4.2.2.2	Link-Based Models and Algorithms for UE.....	202
4.2.2.3	Path-Based Models and Algorithms for UE	203
4.2.2.4	Link-Based Models and Algorithms for SNL and SUE	204
4.2.2.5	Path-Based Models and Algorithms for SNL and SUE.....	210
4.3	Algorithms for Multinomial Probit Path-Based Stochastic User Equilibrium	211
4.3.1	Introduction.....	211
4.3.2	Test Methodology and Test Bed	213

4.3.2.1	Algorithm Test Methodology	213
4.3.2.2	Algorithm Test Bed	215
4.3.3	Application of Algorithms in the Literature	217
4.3.3.1	The Method of Successive Averages (MSA).....	217
4.3.3.2	Quadratic and Cubic Interpolation Methods.....	218
4.3.3.3	Conclusions.....	229
4.3.4	Algorithms with Alternative Step Calculation Methods.....	230
4.3.4.1	Introduction.....	230
4.3.4.2	The Method of Weighed Averages (MWA)	230
4.3.4.3	The <i>Regula Falsi</i> Method	233
4.3.4.4	The Rational Interpolation Method.....	238
4.3.4.5	Partial Step Quadratic Interpolation Methods.....	244
4.3.4.6	Conclusions.....	251
4.3.5	Algorithms with Alternative Search Directions.....	252
4.3.5.1	Introduction.....	252
4.3.5.2	Preconditioning Optimisation Problems.....	252
4.3.5.3	An Interpretation of the Traditional Search Direction	254
4.3.5.4	Preconditioned Conjugate Gradient Search Directions	260
4.3.5.5	Application of the Precise Preconditioned Fletcher-Reeves and Polak-Ribiere Search Directions	266
4.3.5.6	Application of the Approximate Preconditioned Fletcher-Reeves Search Direction	271

4.3.5.7	Application of the Approximate Preconditioned Polak-Ribiere Search Direction	280
4.3.5.8	Comparison of Approximate Preconditioned Conjugate Gradient Algorithms and Conclusions.....	290
4.4	Conclusions.....	291
5.	EXTENSIONS OF SUE: MULTIPLE USER CLASSES AND ELASTIC DEMAND	294
5.1	Introduction.....	294
5.2	Stochastic User Equilibrium Models with Multiple User Classes (MUC SUE)	295
5.2.1	Introduction.....	295
5.2.2	MUC SUE Models in the Literature	296
5.2.3	The MUC SUE Model of Daganzo (1982).....	296
5.2.3.1	The Gradient of the MUC SUE Objective Function.....	299
5.2.3.2	Shape of the MUC SUE Objective Function and Uniqueness of Flows and Costs.....	302
5.2.4	Algorithms for the Probit Path-Based MUC SUE Problem.....	304
5.2.4.1	Introduction.....	304
5.2.4.2	The MSA Algorithm for MUC-SUE	304
5.2.4.3	MUC SUE Algorithms with Traditional Search Direction and Optimised Line Search.....	305
5.2.4.4	MUC SUE Algorithms with Alternative Search Directions	307
5.2.5	Performance of the Algorithms for the MUC SUE Problem.....	310

5.2.5.1	Introduction.....	310
5.2.5.2	MUC SUE Algorithm Test Methodology.....	310
5.2.5.3	MUC SUE Algorithm Test Bed.....	312
5.2.5.4	Test Results for the Chen and Alpha Network	314
5.2.5.5	Test Results for the Sioux Falls Network	320
5.2.5.6	Test Results for the Headingley Network	325
5.2.6	Conclusions.....	330
5.3	Stochastic User Equilibrium Models with Elastic Demand (SUE ED)	331
5.3.1	Introduction.....	331
5.3.2	Elastic Demand Models in the Literature	333
5.3.3	A New Objective Function for SUE ED.....	338
5.3.3.1	Relationship Between SUE ED Objective Functions	342
5.3.4	Algorithms for the Probit Path-Based SUE ED Problem	343
5.3.4.1	Introduction.....	343
5.3.4.2	SUE ED Path-Based Algorithms with BDA Search Direction....	344
5.3.4.3	SUE ED Path-Based Algorithms with Alternative Search Directions.....	346
5.3.5	Conclusions.....	347
5.4	Stochastic User Equilibrium with Multiple User Classes and Elastic Demand (MUC SUE ED)	348
5.4.1	Introduction.....	348
5.4.2	MUC SUE ED Models in the Literature.....	348

5.4.3	An Objective Function for MUC SUE ED	349
5.4.4	An Alternative Objective Function for MUC SUE ED	353
5.4.5	Algorithms for the Probit Path-Based MUC SUE ED Problem	357
5.4.5.1	Introduction.....	357
5.4.5.2	The MSA Algorithm for MUC SUE ED	357
5.4.5.3	MUC SUE ED Algorithms with MUC BDA Search Direction and Optimised Line Search.....	358
5.4.5.4	MUC SUE ED Algorithms with Alternative Search Directions..	359
5.4.6	Performance of the Algorithms for the MUC SUE ED Problem.....	360
5.4.6.1	Introduction.....	360
5.4.6.2	MUC SUE ED Algorithms Test Methodology.....	360
5.4.6.3	MUC SUE ED Algorithms Test Bed.....	362
5.4.6.4	Test Results for the Chen and Alpha Network	364
5.4.6.5	Test Results for the Sioux Falls Network	368
5.4.6.6	Test Results for the Headingley Network	371
5.4.7	Conclusions.....	374
5.5	Conclusions.....	376
6.	CALIBRATION OF MULTINOMIAL PROBIT CHOICE MODELS	379
6.1	Introduction.....	379
6.2	Specification of the Multinomial Probit Model.....	381
6.3	Multinomial Probit Calibration Methods in the Literature	387

6.4	Multinomial Probit Calibration Using the Mendell-Elston Approximation: Method	389
6.4.1	Introduction.....	389
6.4.2	Analytical Derivatives for the Mendell-Elston Log-Likelihood	391
6.4.2.1	Introduction.....	391
6.4.2.2	Calculation of the Derivatives of the Probability of the Option Actually Chosen w.r.t. the Parameters Appearing at the Beginning of the Application of the Mendell-Elston Approximation.....	393
6.4.2.3	Calculation of the Derivatives of the Probability of the Option Actually Chosen w.r.t. the Control Variables of the Optimisation	401
6.4.2.4	Calculation of the Derivatives of the Log-Likelihood Function w.r.t. the Control Variables of the Optimisation.....	406
6.4.2.5	Extension of the Method to Consider Probabilities Normalised to Their Sum	407
6.4.2.6	Analytical Derivatives with Other Approximations	408
6.5	Multinomial Probit Calibration Using the Mendell-Elston Approximation: Numerical Tests	409
6.5.1	Introduction.....	409
6.5.2	Test Methodology	409
6.5.3	Test Bed	411
6.5.4	Calculation Time Results.....	415
6.5.5	Parameter Retrieval Results.....	417
6.5.6	Conclusions.....	434

6.6	Conclusions.....	435
7.	SUMMARY AND CONCLUSIONS.....	437
	REFERENCES.....	445
	APPENDIX A.....	467
A1	Gradient and Hessian of the MUC SUE Objective Function in Terms of Standardised Link Flows	467
A2	Gradient and Hessian of the MUC SUE Objective Function in Terms of Common Link Costs.....	470
A3	Gradient and Hessian of the SUE ED Objective Function in Terms of Link Flows	472
A4	Gradient and Hessian of the SUE ED Objective Function in Terms of Link Costs	475
A5	Gradient and Hessian of the MUC SUE ED Objective Function Derived from the Formulation of Maher <i>et al.</i> (1999).....	478
A6	Gradient and Hessian of the Short MUC SUE ED Objective Function in Terms of Standardised Flows	484
A7	Gradient and Hessian of the Short MUC SUE ED Objective Function in Terms of Common Link Costs.....	487
	APPENDIX B: PEER REVIEWED PUBLICATION	490

1. INTRODUCTION

Transportation models are used in the planning and management of transportation facilities to support decision making and analysis of transportation systems. Their results are used by planners to appraise the effects of implementing different possible alternative actions in pursuing some set objectives and to decide which actions are the best to plan and undertake.

Decisions made on the basis of a transportation model's results have important impacts on people's everyday life and on its quality as well as on the economy and the environment of the areas involved. They also often entail the use of large amounts of resources.

Therefore, although any model's results are approximate in nature since a model can describe reality only to a certain extent, it is important to develop and use models which are capable of capturing real phenomena as accurately as possible thus providing decision makers with sound and significant data to make decisions on.

Many transportation models relate the usage of a transportation system to its characteristics and to those of a population of potential users with cause-effect relationships.

The phenomena investigated in transportation are often the result of choices made by a number of individuals. An example of such phenomena is the choice of mode of travel amongst a number of available ones in a certain area (*e.g.* in an urban context: foot, bicycle, private car as driver, private car as passenger, taxi, bus, LRT, underground), that results in the overall mode choice pattern for that area. A further example is the choice of route through a road network, that results in the traffic pattern on that road network. The cause-effect relationships used in transportation models dealing with such phenomena are therefore choice models that are mathematical models able to describe how people choose amongst alternative

options. Thus the analyst can investigate the behaviour of potential users of transportation facilities in different situations and evaluate the effectiveness before implementation and, more generally, the effect of possible schemes and changes to transportation systems: for instance the effect of possible modifications to some characteristics of a public transport service on its patronage or how the road traffic would redistribute, and perhaps vary in quantity, if a motorway were to be added to the available roads between two cities or if a city centre were to be pedestrianised. In the same way it is also possible to assess how a transportation system would respond to changes in the demand for its usage as in the case of an important traffic generator, as a main commercial centre, were built.

A key factor for obtaining sound results is the use of an appropriate choice model. A number of choice models and methods for their application to specific transport problems have been devised and applied in the literature over the years and the research efforts to obtain more detailed and realistic but also practically usable models continue.

Rather than working on new models, this thesis focuses on the use of the multinomial probit (MNP) model, a well-known and theoretically sound but not widely used choice model, and attempts to address some issues related to its practical implementation focussing on analytical methods to solve it and on their efficient use in traffic assignment models (that describe how traffic distributes itself on a road network), for which it is particularly suitable, and for the solution of the calibration stage (in which the parameters of the models are retrieved from surveyed data) of more general choice modelling problems in which the MNP is interesting for its flexibility to represent different choice mechanisms.

The main reason behind the relatively limited use of the multinomial probit choice model in practical applications is that it is not easily tractable mathematically: its choice function, that is the function defining the result of the model, cannot be written in closed form and approximate solution methods need be used.

Traditionally, this has led researchers and practitioners to prefer for many applications other choice models that can be solved directly and more easily, although they are not necessarily as suitable for the modelled phenomena or as flexible as the MNP.

Alternatively, most of the applications and research works using the MNP model have relied on its solution by simulation.

The approach taken in the present work is to consider methods for the analytical solution of the MNP model, as opposed to simulation based methods, on the basis of the efficiency and of the repeatability of the results that they allow.

The analytical approach to the solution of the MNP model and, in particular, the use of analytical approximations has been considered in a rather limited number of research works presenting several different methods (for traffic assignment see *e.g.* Daganzo and Sheffi, 1977; Maher, 1992; Maher and Hughes, 1997a; for model calibration see *e.g.* Daganzo *et al.*, 1977; Hausman and Wise, 1978; Kamakura, 1989) but it has not found its way into modelling practice. This is possibly because of the limited applicability of some of the methods proposed and of the lack of commercial software.

The present study builds on the results obtained in previous work trying to establish a set of data showing that the approach is actually efficient and competitive.

The **aims of the study** are therefore to explore different possible analytical solutions to the probit model, refining them where possible, establishing a set of data on their accuracy and cost and characterising those most suitable for transportation applications (for accuracy and computational efficiency) as well as to investigate efficient ways to apply such analytical solution methods to traffic assignment and model calibration, also by exploiting the advantages that they offer.

The remainder of the thesis is organised as follows.

Chapter 2 introduces choice models and outlines their application to transportation problems. It reviews the main characteristics and the applicability of the multinomial probit model and of the main alternative choice models used in transportation (with particular focus on those used for mode choice and route choice) and compares them remarking the high degree of flexibility of the MNP. The stages in the practical use of a choice model are outlined and the concepts of simple and equilibrium choice problems are discussed.

Whilst the issues related to the evaluation of the MNP choice function (the base function for the application of the model) are mentioned in chapter 2, **chapter 3** deals with them in detail. The three kinds of evaluation methods that can be used are introduced: simulation, numerical integration and approximation. The main simulation methods are described and, although they are at present the most widely used solution methods in research and practical applications, arguments are given for preferring analytical methods. The rest of chapter 3 is thus devoted to describing and testing some analytical MNP solution methods. A brief overview of the numerical approximation techniques devised in the past precedes the description of the recently proposed method of Genz (1992), the first one to be suitable for the large dimensional MNP problem. Then a number of analytical approximation methods, some of which already used in transportation and some taken from other disciplines, are described in detail and tested for accuracy and computational cost to assess their suitability for transportation problems. The tests are carried out in the context of route choice on a large number of small artificial networks. The most precise and best-performing approximations are characterised. Finally, the satisfaction function is defined and alternative methods for evaluating it when using the MNP choice model are described.

The problem of devising efficient algorithms for stochastic user equilibrium (SUE) path-based traffic assignment is studied in **chapter 4**. First, the traffic assignment problem is defined, some of the models for solving it are introduced and a review of link-based and path-based traffic assignment models and algorithms is given with a survey of the techniques for forming the sets of used paths between the origins and

destinations on a network. Several different SUE solution algorithms incorporating some of the approximations investigated in chapter 3 are then described and tested for performance on real size networks. The algorithms include techniques already used in the literature and others obtained by building on them. The discussion is based on path-based algorithms applying MNP approximations but similar algorithms could also be used in link-based cases and with other choice models whose choice function can be written in closed form or approximated analytically.

Chapter 5 considers two extensions of SUE path-based traffic assignment: the inclusion of multiple user classes and of elastic demand. The former allows us to account for differences across the population of travellers of the factors affecting route choice whilst the latter is an aggregate way to include variable demand in traffic assignment models, that means extending the models for route choice on a road network to include other choices, hierarchically higher (as *e.g.* the choice of mode or the choice of travelling or foregoing a trip).

Methods for solving path-based SUE problems including multiple user classes and elastic demand, separately and together, are developed and discussed, building on the work presented in chapter 4. In particular, new formulations for SUE with elastic demand and SUE with multiple user classes and elastic demand are put forward. Then, algorithms for SUE with multiple user classes and for SUE with multiple user classes and elastic demand are introduced and tested on the same real size networks considered in chapter 4.

Chapter 6 deals with a different problem related to the practical use of the MNP model: its calibration, that is the process of retrieving the parameters governing the choice model from surveyed choice data. A review of the literature on the subject and a discussion of the relevant issues (difficulties for the correct specification of the model, complexity of the calibration problem) introduce the chapter. Then the methods used to develop a program for the calibration of MNP models using one of the approximation methods analysed in chapter 3 are discussed. In particular, it is explained how, in some cases, the recursive structure of the calculations for the MNP approximations considered can be exploited to obtain analytical derivatives of the likelihood function. The computer program developed using the method described is

then applied to some artificial choice situations to confirm the efficiency of using analytical derivatives and investigate the quality of the retrievable results for a number of model structures and sample sizes.

Chapter 7 summarises the work reported in the thesis, drawing conclusions and suggesting further possible research work on the problems studied.

2. RANDOM UTILITY DISCRETE CHOICE MODELS IN TRANSPORTATION

2.1 Introduction

The present chapter introduces choice models, on which the models and algorithms considered in the remainder of the thesis are based, by summarising the results of a literature survey on the subject. In particular, it outlines the characteristics of the multinomial probit model, on which the research work focuses, and those of the main alternative choice models currently available in the literature.

The stages in the practical application of a choice model are outlined to introduce the work on equilibrium route choice models and on model calibration presented in the following chapters.

2.2 The Random Utility Framework

Transportation models relate the usage of transportation facilities to their characteristics and to those of a population of potential users.

They can be used, for instance, to describe how a population of users will choose at what time or by which transport mode to travel to their destination amongst different possible ones, to model the traffic pattern resulting from the routes that people drive along or from the public transport services that people use. All these phenomena, and others studied in transportation, can be described as the results of choices of individuals: in the examples above the choice of the time to travel at, of the mode to travel with and of the route to travel along. In fact, many transportation models are actually applications of choice models.

Choice models replicate how a decision maker (in transportation, typically, an individual or an household) chooses amongst several mutually exclusive options. Since in transportation the options considered normally form a discrete set, discrete choice models are used. A sound and effective theoretical framework for modelling

discrete choices is provided by the random utility theory. This is not the only possible framework (see *e.g.* Ben-Akiva and Lerman, 1985, for alternative theoretical settings) but it is the most established one in transportation as well as in econometrics, to which transportation modelling is tightly related.

The random utility theory assumes that each decision maker has a set of available choice options, and attaches to each of them a quantity known as utility which is not directly observable but depends on the user's and option's characteristics. Then the decision maker ranks the alternatives according to their perceived utilities and chooses the one with the highest utility.

Whilst it is not possible to observe the perceived utilities, the characteristics of the alternatives and of the choice makers can be measured and the choices recorded. Therefore, the utilities are modelled as random variables whose distributions are functions of the observed characteristics and the choice models give the probability that each available option is chosen for the given user's and options' characteristics.

In this framework, the utility U_i associated by a decision maker to an alternative i in their choice set containing J alternatives can be written as the sum of a systematic or deterministic term V_i and a random term ε_i (also called error or disturbance). In general both terms can be functions of the vector of parameters of the model θ and of the vector \mathbf{a} of the characteristics of the alternative and socio-economic attributes of the decision maker:

$$U_i(\theta, \mathbf{a}) = V_i(\theta, \mathbf{a}) + \varepsilon_i(\theta, \mathbf{a}) \quad (2.1)$$

The systematic part of the utility is usually expressed as a linear combination of the relevant attributes \mathbf{a} or of their transformations as *e.g.*:

$$V_i(\theta, \mathbf{a}) = \sum_E \theta_e a_e \quad (2.2)$$

where e is the subscript associated to each of the E elements entering the systematic utility expression. The coefficients θ_e can be either considered fixed or some or all of them can be assumed randomly distributed across the population according to a

suitable distribution. In the latter case the model is said to account for taste variations across the population and (2.2) can be rewritten as:

$$U_i(\boldsymbol{\theta}, \mathbf{a}) = \sum_E \bar{\theta}_e a_e + \eta_i(\boldsymbol{\theta}, \mathbf{a}) + \varepsilon_i(\boldsymbol{\theta}, \mathbf{a}) \quad (2.3)$$

where $\bar{\theta}_e$ are the means of the distributions of the coefficients $\boldsymbol{\theta}$ and η_i is the additional random term deriving from the taste variations (for simplicity taste variations are not normally considered hereon).

The random part ε_i of the utility is assumed to account for unobserved attributes (*i.e.* not included in the systematic part), unobserved taste variations (that is accounting for coefficients of the attributes included in the systematic part considered as fixed whilst they should be more correctly considered as randomly distributed), measurements errors and imperfect information and instrumental variables (Ben-Akiva and Lerman, 1985).

Since it is assumed that the alternative chosen is the one with the highest perceived utility and that the utilities are random variables, the probability that an alternative i is chosen is equal to the probability that its utility is greater than those of the other options in the choice set:

$$P_i(\boldsymbol{\theta}, \mathbf{a}) = P(U_i(\boldsymbol{\theta}, \mathbf{a}) > U_j(\boldsymbol{\theta}, \mathbf{a})); \quad \forall j \neq i; \quad i, j \in J \quad (2.4)$$

Substituting a compact form of the formula (2.1) for the alternatives' utilities, (2.4) can also be written:

$$P_i(\boldsymbol{\theta}, \mathbf{a}) = P(V_i + \varepsilon_i > V_j + \varepsilon_j) = P(\varepsilon_j - \varepsilon_i < V_i - V_j); \quad \forall j \neq i; \quad i, j \in J \quad (2.5)$$

It is sometimes useful to expand (2.5) into the following integral:

$$P_i(\boldsymbol{\rho}) = \int_{-\infty}^{V_i - V_1} \int_{-\infty}^{V_i - V_2} \dots \int_{-\infty}^{V_i - V_{i-1}} \int_{-\infty}^{V_i - V_{i+1}} \dots \int_{-\infty}^{V_i - V_n} f(\boldsymbol{\rho}) d\boldsymbol{\rho}; \quad i, j \in J \quad (2.6)$$

where $\boldsymbol{\rho}$ is the multivariate random variable of the differences $\varepsilon_j - \varepsilon_i, \forall j \neq i$, $f(\boldsymbol{\rho})$ is its density function and the upper limits of integration result from considering (2.5) for each alternative option different from i .

The actual functional form of (2.5) or (2.6) depends on the assumed distribution of the systematic utility coefficients and of the random terms which also determine the different choice models.

This is an important point as the error term distribution allowed by each different model, and therefore the pattern of substitution it can accommodate, determines the model's suitability to represent different choice problems.

For instance, if the coefficients of the systematic part of the utility are fixed and the error terms are independently and identically distributed type I extreme value (Gumbel) variates the multinomial logit (MNL) model results, which is perhaps the simplest and most used stochastic choice model. The multinomial probit (MNP), which is the subject of this work, results when the coefficients of the systematic utility are fixed or jointly multivariate Normally distributed and the error terms are specified as jointly multivariate Normally distributed with zero mean and arbitrary covariance matrix.

Both the MNL and the MNP are introduced in more detail in the next section along with other random utility choice models used in transportation.

2.3 A Survey of Random Utility Discrete Choice Models

2.3.1 Introduction

A number of different discrete choice models have been developed and applied in the literature, ranging from simple and numerically convenient ones to more sophisticated models able to capture the mechanism of complex choice situations.

The development of new models, or of techniques to make feasible and efficient the use of existing models, is motivated by the need to use in practical studies models that are as realistic as possible, and often by the needs arising in each particular application. The remainder of this thesis focuses on route choice problems and on the calibration of more general choice problems so it is worth mentioning which are the general requirements of models suitable for these applications.

The choice of route through a network should be represented with models able to capture the different variability and the correlation between the alternatives. The alternatives in route choice are the paths between each two points of origin and destination of a journey. The utility of paths of different characteristics (*e.g.* length or travel time) should be assumed as perceived with a different variation. Moreover when the paths between two points on the network partially overlap their utility must be represented as partially correlated. These are actually quite demanding requirements that can be fully met only by some very flexible model, as the following survey shows.

In the case of model calibration that is, as explained in section 2.4.3, when the parameters of a model specification are retrieved from field data and very likely different model specifications are tested for the ability to fit the choice situation under study, the structure of the model is not known a priori although the analyst will have some information on it. When the choice structure is well defined a model directly fitting it can be used but, more in general, flexible models, able to represent simple choice situations as particular cases, should be used to check which model specification best fits the data and with what significance. Hence the need for models able to accommodate a rather wide range of choice situations, the simpler ones being “nested” as particular cases in the more complex and general ones.

The sort of data used in model calibration is a further reason to have models able to account for different variation and correlation of the alternatives’ utilities. For instance when panel data are used, that is when a model is fitted using data on the choices of the same individual monitored at different points in time, the correlations across time need to be taken into account. Also when actual and stated choices (the latter are choices stated by a choice maker confronted with hypothetical situations set up by the analyst; see *e.g.* Ortúzar and Willumsen, 1994) are considered together there is a need to account for different variability of the utility of the two types of data as there is when taste variations are included and the model needs to capture their correlation as well as their variability.

The following brief survey of choice models reviews the multinomial probit model, used in the rest of this work, and some of the most important alternative models

presently in the open literature. Although the focus here is on stochastic models the deterministic model is introduced first both as a base model and because it is largely applied, mainly for traffic assignment.

2.3.2 The Deterministic Model

The deterministic model is obtained when the error terms in the general specification of random utility discrete choice models (see *e.g.* formula 2.1) are assumed to be zero.

The result is that the decision is modelled as made on the systematic part of the utility of each option in a choice set and the option with the highest utility has a probability 1 of being chosen whilst all the others have zero probability.

Using this model entails assuming that the decision makers perceive an utility equal to the systematic utility specified by the analyst. This is often stated saying that the decision maker has a perfect knowledge of the utility of each alternative (as it is accounted for in the model).

Although the deterministic model is used in route choice when it is believed that attributes included in the systematic utility would be dominant if the random terms were not assumed to be zero, thus making the latter of little importance, this is generally too simplistic a model for most applications and stochastic models, with random errors not equal to zero, are used instead.

2.3.3 Stochastic Models

2.3.3.1 The Probit Model

The probit model or multinomial probit model (MNP) results when the error terms in the general specification of random utility discrete choice models (formula 2.1) are assumed to be multivariate Normally distributed with zero mean and arbitrary covariance matrix. The choice model remains a MNP also if the coefficients of the systematic utility are multivariate Normally distributed rather than fixed. The utility

vector \mathbf{U} of dimension J is therefore $MVN(\mathbf{V}, \Sigma)$ with $\mathbf{V}=\mathbf{V}(\boldsymbol{\theta}, \mathbf{a})$ and $\Sigma=\Sigma(\boldsymbol{\theta}, \mathbf{a})$ and its probability density function is therefore:

$$f(\mathbf{V}, \Sigma) = \left[(2\pi)^J \cdot |\Sigma| \right]^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{U} - \mathbf{V}) \Sigma^{-1} (\mathbf{U} - \mathbf{V})^T \right] \quad (2.7)$$

Considering the utilities as Normally distributed is theoretically sound as it is backed up by the Central Limit theorem, as pointed out *e.g.* by Daganzo and Sheffi (1977), since it is assumed that the error terms derive from the effect of several unobserved factors, as mentioned in section 2.2.

A seminal version of the MNP, for a binomial case, was put forward in psychometrics by Thurstone (1927). The model was adopted in transportation and econometrics later (Hausman and Wise, 1978). A comprehensive description of the model can be found in Daganzo (1979) although for the issues related to its identification and specification the studies listed in section 6.2, which reviews these issues, should be referred to.

The multinormal distribution of the random terms makes the MNP model very flexible, and suitable to represent the different patterns of substitution arising in different applications. The flexibility of the MNP has been discussed *e.g.* by Daganzo (1979), Bouthelier and Daganzo (1979), Sheffi *et al.* (1982) and it is important to remark that the MNP is much more flexible than most other discrete choice models in the literature as the survey in the following paragraphs shows.

In general, the MVN distribution of utilities can be specified to accommodate homoscedastic and heteroscedastic independent utilities and also different patterns of correlation between homoscedastic or heteroscedastic utilities (although with some limitations when a model is being calibrated, as noted in section 6.2), such as correlation between all options or between groups of options in a choice set.

The MVN distribution of the utilities results also in the ability to deal with cross sectional choice data along with panel data, to allow for calibration with aggregate or mixed aggregate and disaggregate data and also with missing data or data containing errors; moreover, the aggregation of the result is particularly simple if the attributes over which they are aggregated are Normally distributed (see Daganzo, 1979;

Bouthelier and Daganzo 1979; Sheffi *et al.*, 1982). The MNP can accommodate random taste variations due to Normally distributed coefficients of the attributes entering the systematic utility.

With the multinomial probit model, the choice probability of an option i in a set of J options, that is (2.4), is expressed by:

$$P_i(\mathbf{V}, \Sigma) = \int_{u_1=-\infty}^{u_i} \int_{u_2=-\infty}^{u_i} \dots \int_{u_i=-\infty}^{\infty} \dots \int_{u_j=-\infty}^{u_i} [(2\pi)^J \cdot |\Sigma|]^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{U} - \mathbf{V})\Sigma^{-1}(\mathbf{U} - \mathbf{V})^T\right] du_1 du_2 \dots du_i \dots du_j \quad (2.8)$$

This integral cannot be written in closed form and calculated exactly. This lack of tractability has counterbalanced the theoretical appeal and the flexibility of the model hampering, at least to some extent, its development and use in research and practical applications.

However, the choice function (2.8) can be solved approximately with a number of analytical or simulation based methods. The analytical methods are based on numerical integration or on other analytical approximations to (2.8) but have enjoyed only limited interest in the literature. In the case of the numerical integration methods this is due to the limited practical applicability of the algorithms usually considered, since their implementations are time consuming and become increasingly and extremely so when the dimension of the MNP problem increases beyond 4 (see *e.g.* Munizaga *et al.*, 2000). The approximation methods have had scarce use perhaps after the limited accuracy of the approximation of Clark has been underlined in Horowitz *et al.* (1982). Monte Carlo simulation algorithms have typically been preferred to solve the MNP choice function and are the only methods that have found their way into commercial software: MNP traffic assignment methods are available *e.g.* in the software packages SATURN (Van Vliet and Hall, 1993) and TRIPS (MVA, 1994). More recently, improved MNP simulation methods to solve (2.8) have been proposed (see *e.g.* McFadden, 1989; Borsch-Supan and Hajivassiliou, 1993), helping to overcome the traditional objections to the model found in the literature concerning its tractability, bringing about some renewed

interest in the MNP for econometrics and transportation applications (see *e.g.* Geweke *et al.*, 1994; Bolduc, 1999; Munizaga *et al.*, 2000) and also providing practical methods to use the mixed logit model described in section 2.3.3.8.

Approximate methods for solving the MNP are introduced in detail in chapter 3 in which the possibility of solving analytically the MNP is reconsidered and the performance of some analytical methods is assessed.

2.3.3.2 The Multinomial Logit Model

The logit model (McFadden, 1973), or multinomial logit model (MNL) is the simplest and best known stochastic choice model in the literature (for a comprehensive description see *e.g.* Ben-Akiva and Lerman, 1985). It results when the error terms ε_i in the general specification of the random utility (formula 2.1) are assumed to be identically and independently Gumbel (type I extreme value) distributed *i.e.* with cumulative distribution function:

$$F(\varepsilon) = \exp(-\exp(-\mu(\varepsilon - \eta))) \quad (2.9)$$

where η is a location parameter and μ is a strictly positive scale parameter. The parameters η and μ are related to the mean of the Gumbel distribution by the formula:

$$mean = \eta + \gamma / \mu \quad (2.10)$$

where γ is the Euler constant (see also *e.g.* Ben-Akiva and Lerman, 1985).

This assumption entails that the covariance matrix of the joint distribution of the random terms (and of the utilities) is a fixed diagonal matrix with non-zero entries equal to σ^2 , the common variances of the Gumbel variates, that are given by the formula:

$$\sigma^2 = \frac{\pi^2}{6\mu^2} \quad (2.11)$$

As noted by Ben-Akiva and Lerman (1985) the use of the Gumbel distribution in the logit model can be justified as an approximation to the Normal distribution.

The main advantage of assuming the Gumbel distribution for the random terms is that it is closed under maximisation and therefore results in a simple and tractable choice function. In fact, in the MNL case, the choice function (2.4) expressing the probability P_i of choosing alternative i , whose systematic utility is V_i , from a choice set containing J alternatives can be rewritten as:

$$P_i = \frac{\exp(\mu V_i)}{\sum_{j \in J} \exp(\mu V_j)} \quad (2.12)$$

The distribution of random terms assumed in the MNL makes it comply with the independence from irrelevant alternatives (IIA) axiom. This, as reported by Ortúzar and Willumsen (1994), states that when any two alternatives have a non zero probability of being chosen, the ratio of one probability over the other is unaffected by the presence or absence of any additional alternative in the choice set. This implies that the reduction or increase of any alternative's utility will have the same proportional impact on the probability of choice of all other alternatives.

The compliance with the pattern of substitution assumed by the IIA axiom underlines that the logit model is suitable only for choice situations with independent options with the same variance of the utility and therefore also makes the model unable to deal correctly with situations, such as those mentioned in the introduction to this section, where these assumptions are not correct. This has often been disregarded in practical applications although doing so gives biased choice results as discussed in the traffic assignment case *e.g.* by Florian and Fox (1976) and by Daganzo and Sheffi (1977).

The convenient closed functional form of the logit model and its applicability limitations have motivated research work that has led to richer logit-related models retaining the closed functional form while expanding the modelling capabilities. The most important ones are introduced in the following paragraphs.

2.3.3.3 The Nested Logit Model

The nested logit (NL), or hierarchical or tree logit, is the most used and longest established “extension” of the logit model. An account of its origins and of the studies on this model is given in Ortúzar (2001) whilst detailed information on its characteristics and use can be found *e.g.* in Ben-Akiva and Lerman (1985).

The NL can account for correlation of the utility within mutually exclusive groups of alternatives. In the NL model the alternatives can be seen as organised in a hierarchical tree-like structure (hence its name) where correlated alternatives are grouped in “nests”. Fig. 2.1 sketches the structure of an example with one level of nests only.

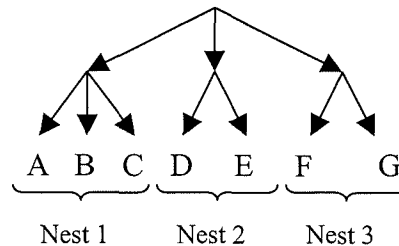


Fig 2.1 – Example of correlation structure amongst alternatives accommodated by a nested logit model with one level of nests. The letters A to G denote the alternatives.

The random terms of an option’s utility can be seen as resulting from a part relative to the nest it belongs to and a part relative to the option itself. The two parts of the error term are independent and are both Gumbel distributed. This results in the utilities of the alternatives within a nest having the same variance of the error term and a common correlation (determined by the coefficient of the nest) whilst alternatives belonging to different nests may have different variances and are uncorrelated. This utility structure corresponds to a joint distribution of the utilities of all the alternatives with a fixed block diagonal covariance matrix whose off diagonal entries can be calculated with the formula obtained by Daganzo and Kusnic (1993).

The choice probability of an alternative is expressed as the product of the marginal probability of choosing the nest to which it belongs and the conditional probability of choosing that alternative in the nest. Marginal and conditional probabilities are obtained with closed form functions similar to (2.12). For instance, the choice

probability of option i amongst the possible J belonging to nest m (out of N) in a nested logit model with one level of nests as in fig. 2.1, is given by:

$$P_C = P(m) * P(i|m) = \frac{\exp \mu(V_m + L_m)}{\sum_{n \in N} \exp \mu(V_n + L_n)} \frac{\exp(\beta_m V_i)}{\sum_{j \in J} \exp(\beta_m V_j)} \quad (2.13)$$

where $L_n = \frac{1}{\beta_n} \ln \sum_{j \in J} \exp(\beta_n V_j)$ is the expected maximum utility of the J options belonging to nest n .

Nested logit models can also include multiple levels of nests to represent more complex grouping of choice alternatives but the nests at each level can only include mutually exclusive alternatives or groups of alternatives. Alternatives not belonging to the same nest at any level are not correlated.

The wide use of the NL model is due to its ability to suit several choice situations. For instance, the patterns of correlation it accommodates can be used to model the choice between different transport options some of which have similar characteristics not accounted for in the systematic part of the utility as in the choice between several public and private transport modes.

Furthermore, the possibility of using correlated alternatives (although with some limitations) with a common part of the error terms can be used in mixed revealed stated preference studies where stated and observed choice data have a different variability and must be included in separate exclusive nests.

The possibility of introducing heteroscedasticity amongst options in different nests has also been used by Munizaga *et al.* (2000) to model a situation with heteroscedastic independent options with the Single Element Nested Logit (SENL) model: a nested logit model with only one alternative in each nest.

However the utility structure the NL allows is not completely general: for instance correlation structures such as those amongst routes between origin and destinations encountered in traffic assignment cannot be generally modelled. Moreover, since its errors terms cannot be function of the options' attributes, the NL model cannot treat taste variations.

2.3.3.4 The Generalised Extreme Value Model

The Generalised Extreme Value (GEV) model, proposed by McFadden (1978), is used in the literature to derive choice models that are particular cases of its functional formulation rather than directly as a choice model in its own right. The MNL and the NL models can be seen as special cases of the GEV model, although they have not been initially obtained as such. In fact, the GEV model has been included in this survey because it has been used to derive the Cross Nested Logit (CNL) and the Paired Combinatorial Logit (PCL), reported in the next two sections, that extend the modelling capabilities of the logit-like models, although not to the extent of including taste variations as, in general, GEV models cannot include randomly distributed coefficients of the systematic part of the utility.

Assuming $Y_i = \exp(V_i)$ where V_i is the systematic component of the utility for alternative i , a GEV model (see *e.g.* Ben-Akiva and Lerman, 1985) can be derived from any function:

$$G(Y_1, Y_2, \dots, Y_n) \quad Y_1, Y_2, \dots, Y_n \geq 0 \quad (2.14)$$

which is non-negative, homogeneous of degree $\mu > 0$, approaches infinity with any Y_i , $i=1, 2, \dots, n$ and has k th cross partial derivatives which are non-negative for odd k and non-positive for even k . Given the function G , the choice function for alternative i in the resulting GEV model can be written as:

$$P_i = \frac{Y_i G_i(Y_1, Y_2, \dots, Y_n)}{\mu G(Y_1, Y_2, \dots, Y_n)} \quad (2.15)$$

where G_i is the first derivative of G with respect to Y_i .

The actual pattern of substitution obtained, and therefore the particular model that results, depends on the function G used. For instance, taking $G = \sum_{i=1}^n (Y_i)^\mu$ gives the MNL model with scale parameter μ .

2.3.3.5 The Cross-Nested Logit Model

The cross-nested logit (CNL) model has been recently obtained by Vovsha (1997) as a particular case of the GEV model and has been applied by Vovsha and Bekhor (1998) in traffic assignment.

The CNL has a structure similar to the NL but the options may belong to several nests of the same hierarchical level and may have a different degree of inclusion to each of the nests they belong. Vovsha (1997) developed the model to study a mode choice situation. In traffic assignment it has been applied to capture the even more complex pattern of correlations amongst the routes between an origin-destination pair.

As noted by Koppelman and Sethi (2000), although alternatives can be included in different proportions in each nest, the parameter related to each nest, a parameter similar to μ in (2.13), is the same for all nests.

The choice function for each option is given by the summation over all the nests to which it belongs of the product of the marginal probability of choosing the nest by the conditional probability of choosing that option in the nest. Marginal and conditional probabilities are obtained similarly to the NL case, therefore with a closed form function similar to (2.13), although the exponential related to each option in each nest is weighted by the relevant degree of inclusion.

2.3.3.6 The Paired Combinatorial Logit Model

The paired combinatorial logit (PCL) model was proposed by Chu (1981), reconsidered by Koppelman and Wen (2000) for general choice and used in traffic assignment by Prashker and Bekhor (1998). It can be derived from the GEV and is able to capture complex patterns of correlation since it considers the similarity between all the possible pairs of alternatives in a choice set.

Koppelman and Sethi (2000) noted that, although different nests can have different parameters, each alternative has the same degree of inclusion into each nest it belongs to and this limits the correlation with other alternatives.

The choice probability of an option in a choice set is the sum, over all the pairs to which it belongs, of the products of the marginal probability of choice of that particular pair by the conditional probability that the option is chosen in the pair. Marginal and conditional probabilities are obtained similarly to the NL case, therefore with a closed form function similar to (2.13), although the exponential related to each option in each nest is weighted by the fixed degree of inclusion.

2.3.3.7 The C-Logit Model

The C-logit model has been derived by Cascetta *et al.* (1996) specifically for route choice applications. It cannot be derived from the GEV model and Koppelman and Sethi (2000) presented it as a form of the Mother Logit model. This is a rarely used model proposed by McFadden (1975) that can deal with similarity among alternatives by including in the utility of each option the attributes of the alternative options (see Koppelman and Sethi, 2000; Ben-Akiva and Lerman, 1985). In fact, the C-logit model accounts for the similarity between the options by adding to the utility of each option as considered in the MNL a term, known as the commonality factor, that is a non-linear function of a measure of similarity amongst the option under consideration and the others in the choice set. In the path choice case the degree of similarity is established considering a fixed attribute of the paths between the same origin destination pair (as the length). For instance, one of the specifications of the commonality factor suggested by Cascetta *et al.* (1996) is:

$$CF_i = \beta \ln \sum_{j \in J} \left(\frac{L_{ij}}{L_i^{0.5} L_j^{0.5}} \right)^\gamma \quad (2.16)$$

where β and γ are coefficients, L_i and L_j are respectively the lengths of paths i and j of the set of J paths between an origin-destination pair and L_{ij} is their common length. This device makes the model able to capture the topology of the network and therefore the effect of path overlapping.

The C-logit choice function is as (2.12) but with the relevant commonality factors subtracted from the utilities V_i to account for an option's similarities to other options:

$$P_i = \frac{\exp \mu(V_i - CF_i)}{\sum_{j \in J} \exp \mu(V_j - CF_j)} \quad (2.17)$$

2.3.3.8 The Mixed Logit Model

The mixed logit (ML) model, also known as the error component logit or the hybrid logit model, had first been applied by Boyd and Melman (1980) and Cardell and Dunbar (1980) (both cited by Brownstone and Train, 1999) but only very recently has enjoyed increasing attention and application (see *e.g.*: McFadden and Train, 1996; Train, 1998; Brownstone and Train, 1999).

It has a random component of the utility (2.1) given by the sum of two terms: one independently and identically Gumbel distributed as in a MNL model and the other having a distribution across the J alternatives in the choice set tailored to the problem at hand. Sometimes, but not necessarily, the second error term is multivariate Normally distributed. In other cases it is formed by a mixture of different distributions such as *e.g.* Normal and log-Normal distributions.

The expression of the utility (2.1) for the ML model, assuming the systematic part of the utility as in (2.2), results:

$$U_i = \sum_E \theta_e a_e + \varepsilon_i + \delta_i \quad (2.18)$$

where ε_i is the Gumbel distributed term and δ_i is the remaining random effect. The actual choice model is defined by the distribution of the δ_i terms.

Writing the choice function (2.5) for the ML with the utilities as in (2.18) yields:

$$P_i = \int_{\delta} \frac{\exp(\sum_E \theta_e a_{ei} + \delta_i)}{\sum_J \exp(\sum_E \theta_e a_{ej} + \delta_j)} f(\delta) d\delta \quad (2.19)$$

where $f(\delta)$ is the distribution function of δ and the integral is over the distribution function domain.

The choice function (2.19) and the ML model can be derived in different ways. For instance considering the inclusion in a logit model of taste variations resulting in the non-logit random effect. The choice function (2.19) is also analogous to the expression for the aggregation of logit results over a population with parameters distributed as the error terms δ .

The integral (2.19) expressing the ML choice function cannot be written in closed form and calculated exactly but it is solved in the literature using simulation with methods originating from the probit simulator of McFadden (1989) that obtains the value of the choice function taking draws of the non-logit random term and averaging the results of the resulting logit choice functions over the sample of random terms.

The ML cannot be obtained from the GEV model but it turns out to be more flexible than GEV models thanks to the allowed random term structure that can accommodate taste variations and other random effects of several sorts so to approximate several different substitution patterns. This is the reason for which the ML is encountering increasing favour. McFadden and Train (1996) proved that it can approximate arbitrarily closely any random utility model. For instance, Brownstone and Train (1999) discussed how it can approximate the nested logit model and the cross-nested logit model. This is similar to the substitution patterns that the probit models allow but the main difference is that the ML allows also for not-Normally distributed δ_i .

2.3.4 Discussion

The MNL choice model, although imposing a number of limitations on the choice pattern it can represent, is still widely used and one of its simpler extensions, the NL, is also very widely used (so much so that in the literature it is sometimes referred to as the “workhorse” of choice modelling).

However, as shown in the survey reported above, recently there has been a significant research effort that has resulted in choice models able to represent complex substitution patterns (and include simple ones as particular cases) being presented in the literature and applied in practice.

This evolution is still ongoing and is following different lines of work like the development of closed form models, whether referable to the GEV models (see *e.g.* Wen and Koppelman, 2001; Koppelman and Sethi, 2000) or otherwise, as the C-logit, or like the development of a new and very general model like as the ML.

One other line of development is the refinement of the techniques to solve a well known and general model like the MNP. In this line of work can be included the simulation techniques mentioned in 2.3.3.1 that also led to the development of the ML, and the validation, refinement and application of analytical techniques as in the work carried out in this thesis.

If, on the one hand, logit related models are being expanded to keep a closed form but approximate patterns of substitutions allowed *e.g.* by the MNP, applying analytical techniques for the solution of the MNP can be seen as a different way to propose flexible analytical methods for choice modelling. Moreover the MNP is more flexible than GEV models, which cannot include taste variations, and includes patterns of correlation in a perhaps more intuitive way through the representation of random terms using directly the covariance matrix.

The ML model, which has very recently received increasing attention in the literature, has a modelling flexibility similar to that of the MNP. In fact both MNP and ML can cover the substitution patterns of the other models in the survey reported in this chapter, at least approximately. As suggested by Brownstone and Train (1999), the difference between the MNP and the ML with a MVN mixing distribution is practically negligible. In fact, they support the use of ML on the basis of the possibility of using different mixing distributions which is when the ML offers actually an advantage over the MNP. This is useful, for instance, when random terms, perhaps deriving from taste variations, need to be strictly of one sign which makes the Normal distribution less appropriate than, for example, the log-Normal.

MNP and ML, the two more flexible models in the literature, share the issue of lack of a closed form solution for the choice function. However, the possibility of approximating analytically the MNP is particularly interesting because it results directly in repeatable results (critical in scenario comparisons) and, for traffic assignment models, in the possibility of obtaining the optimised solution methods

discussed in chapter 4 and 5 which could not be developed, for instance, with the MNP model solved by simulation.

2.4 Using Discrete Choice Models

2.4.1 Introduction

This part of the chapter outlines the stages and problems in using discrete choice models to place in this process the work described in the remainder of the thesis.

Whichever is the actual model chosen by the analyst, the final use of a choice model is for predicting choices: actual or hypothesised choice situations are defined by the attributes of the options and of the decision makers and by the relevant parameters (parameters are usually assumed not to change for the same population facing different choice situations). Then the model, with known functional form and parameters, is used to obtain the probability of choice for each option for the population of choice makers under study: the choice pattern.

However, before this forecasting or prediction stage is reached, a choice model must be defined through the specification of the choice set, the functional specification of the model, its calibration and its validation. The definition of the model and the values of its calibrated parameters constitute also a result as they are of help in understanding the relevance of each attribute considered in the choice process.

Furthermore, the forecasting stage is of a different nature depending on whether the utilities of the options can be considered fixed or they vary with the choices, thus representing an interaction amongst the choice makers. This is discussed before considering the aggregation of the choice model results that is the problem of obtaining the overall pattern of choice for a certain population from models predicting the probability of choice of an individual choice maker.

2.4.2 Choice Set Definition and Specification of a Choice Model

The choice set definition and specification stage consists of characterising a choice set, a particular choice behaviour and a functional specification of the model

representing the choice behaviour. It is carried out mainly on the basis of a priori knowledge of the modelled choice phenomenon, although confirmed by statistical information on the significance of the model obtained at the calibration stage, and on resource and data availability and is of major importance to obtain a sound model.

The choice set is often defined a priori, if it is small, by considering the problem under study and characterising the alternatives available to the decision makers.

However, in some cases the choice set is not straightforward to characterise. This is the case in traffic assignment where the choice set, that is the set of routes between each origin and destination pair, is typically formed with suitable algorithms (see chapter 4).

Employing a random utility choice model implies the relevant underpinning choice behaviour outlined in section 2.2.

Deciding the specification of a random utility choice model means specifying the functional form of the utilities and choosing a particular distribution of the error terms in (2.1). As seen above, the latter implies a description of the way variability and correlation of the alternatives' utilities are perceived by the decision maker and the substitution pattern modelled. Ultimately this results in the sort of choice model used.

The specification of the utilities and of the random errors involves selecting the relevant attributes of the users and of the facilities that will be included and how they will appear. The systematic part of the utility is commonly defined by a linear combination of suitable measures of the considered attributes or of their functional transformations as (2.2).

The issue of the specification of the models has different aspects depending on whether the model of interest is used directly for choice forecasting, as could happen in traffic assignment, or is being calibrated.

In traffic assignment the systematic term of the utilities is generally given by the performance function of each element of the network (see chapter 4). These are normally defined and calibrated separately. The variability and, if the model

accounts for it, the similarity of the alternatives are typically defined by some suitable characteristics of the network, although their scale needs to be defined.

When a model for a general choice situation is being specified prior to being calibrated the situation is more complex. Where possible (and sensible) different patterns of substitution and different models including also different sets of attributes are assumed. The calibration stage will return the statistics necessary to choose the pattern of substitution which best replicates the calibration data and the attributes which are actually important in the model.

In defining the possible specifications of the model it must be taken into account that, whilst the models are usually expressed in terms of utilities it is necessary to consider that the data to calibrate them (attributes and choice for each surveyed decision maker) give information about a process where the differences of the utilities rather than the utilities themselves determine the outcome (as results from the behaviour assumed in random utility models) (see *e.g.* Bunch, 1991). The utilities, in fact, are scalable as, according to the random utility theory, they are only used to rank the alternatives. Consequently, when specifying the functional form it is necessary to bear in mind that only the parameters that can be retrieved uniquely from the choice model written in difference with respect to a reference alternative can be estimated. This issue poses limitations on the possible specifications of both the deterministic and the random part of the utility that depend on the sort of model. Those for the MNP are briefly reviewed in section 6.2. Moreover when many choice options are considered the number of defined parameters related to the random term tends to increase very quickly and this applies particularly to models allowing for general choice structures such as the MNP and the ML. Synthetic though significant ways to capture the correlation amongst the options with few parameters, such as factor analytic structures for the covariance matrix, have been developed to deal with this issue (Bolduc, 1992; Ben-Akiva and Bierlaire, 1999).

2.4.3 Calibration of Discrete Choice Models

Once a model is specified, the parameters appearing in it have to be estimated or calibrated using relevant surveyed data and the statistical significance of the estimates evaluated.

Discrete choice models are most often calibrated using the maximum likelihood method, that produces the parameters with which the model best replicates the sample of data used. This method, in fact, consists of calculating the parameter values that maximise the likelihood function. Since the likelihood function is the probability density function of the calibration data as a function of the parameters of the model, maximising it results in producing the parameter values using which it is most likely that the model returns the calibration data. The maximum likelihood estimates of the parameters are consistent, asymptotically Normal and asymptotically efficient (see *e.g.* Ben-Akiva and Lerman, 1985).

The maximum likelihood problem is solved with optimisation algorithms chosen on the basis of the choice model used. For some choice models, as the MNL, the evaluation of the choice function and of its derivatives, and therefore of the likelihood function and of its derivatives, is straightforward so a Newton-Raphson algorithm can be used to obtain the maximum likelihood parameters.

In other cases the evaluation of the derivatives is more complex or possible only numerically. An example of this case is the MNP, whose derivatives evaluation is not straightforward, and which is usually calibrated as in the work reported in chapter 6 with quasi-Newton methods building second order information from function values and first derivatives only.

With most models there is also the issue of possible multimodality of the likelihood function, but it is often disregarded in practical applications.

The calibration of the choice model delivers also data on the statistical significance of the attributes and on the goodness of fit of the model to the data allowing comparisons between different model specifications, that are employed to choose the actual model to use.

2.4.4 Validation of Discrete Choice Models

The validation of a choice model consists of comparing the choice predicted by the model using the parameters obtained in calibration with those in a set of sample data (known as the “hold out” sample since these data are not used for calibration but saved for the validation stage). The attributes used for predicting the choices are those relative to the choice data in the hold out sample.

The validation is to confirm that the choices made in the specification stage and the parameters obtained in the calibration stage are effective and make sure that the model can produce reliable forecasts.

2.4.5 Simple Forecasting and Equilibrium Forecasting

Forecasting consists of applying the choice model, with the coefficients determined in the calibration stage, to real or hypothetical situations, described by sets of attributes, to determine the resulting pattern of choice. It can be carried out with fixed or variable attributes.

Forecasting with fixed attributes is referred to here as simple forecasting. In this case the application of the choice model using the given constant values of the attributes defining the choice situation yields directly the choice pattern. Simple forecasting is used in transportation when the attributes are not influenced by the choices or such influence is assumed to be of limited importance (in general or because of the time span of the model) as *e.g.* for modal split studies.

However when many decision makers compete for the use of options of limited availability there is an interaction between user’s choices that can be described by the utilities of the options being dependent on the options’ usage. Thus, when the utilities of the alternatives change, the demand may be reshaped because users respond to the changes modifying their choices according to the behavioural principle assumed by the model (utility maximisation, here). In turn, when choices change, options’ utilities may change. In transportation this interaction between demand (the users) and supply (the transportation system) is referred to as congestion.

Congestion is normally accounted for in traffic assignment models in which the network performance and sometimes also the overall demand (*i.e.* the number of trips made) are functions, respectively, of the demand on the elements of the network and of the network performance, rather than being fixed.

Modelling choices in such a case is more difficult than when alternatives' utilities are fixed. The models considered in the remainder of the thesis account for congestion within the equilibrium framework. This means assuming that, over the analysed period, utilities and demand are consistent. In practice, in equilibrium models the attributes that change with options' usage are calculated simultaneously to the usage. The concept of equilibrium in traffic assignment is at the basis of the models and algorithms investigated in chapter 4 and 5.

2.4.6 Aggregation of Discrete Choice Model Results

Whether applied to simple or equilibrium choice decisions, discrete choice models give the probabilities that an individual decision maker would choose each of the alternatives in a choice set. However in transportation what is of interest is the whole pattern of choices made by a known population of choice makers. This entails different cases depending on whether such a population is homogeneous or not.

A population of decision makers is homogeneous if it is made up of individuals with the same characteristics who, therefore, can all be described in the model by the same attributes (the possible variability of the coefficients of the models is accounted separately using taste variations). In this case the predicted probabilities of choice for each of the individuals in the population are the same and the overall pattern of choice is the same predicted by the model for the single individual.

When the population of choice makers is not homogeneous, different techniques can be employed to aggregate the results that is to account for how the attributes are distributed across the population and obtain the overall pattern of choice starting from information on the choice of the individuals.

If the choices, or the probabilities of choice, of all the individuals in the population were known, the overall pattern of choice could be obtained simply by considering them together. But usually, in practice, this is not the case. In general the part y_i of the population choosing an option i can be written as (see *e.g.* Daganzo, 1979):

$$y_i = S \int \int \dots \int_{a_1 a_2 \dots a_e} P_i(\boldsymbol{\theta}, \mathbf{a}) f(\mathbf{a}) d\mathbf{a} \quad (2.20)$$

where $P_i(\boldsymbol{\theta}, \mathbf{a})$ is the result of a choice model, the probability that an individual would choose the option i , \mathbf{a} is the vector of attributes with e elements distributed $f(\mathbf{a})$ across a population of size S for which the vector of parameters $\boldsymbol{\theta}$ can be assumed to be constant.

The integral (2.20) expresses the aggregate result of the choices but it is generally difficult to calculate directly. However it can be estimated approximately with aggregation methods such as the naive or average individual method, the market segmentation or classification method, the sample enumeration method and the artificial sample method (see *e.g.* Ben-Akiva and Lerman, 1985; Ortúzar and Willumsen, 1994).

Aggregation, in the reminder of this thesis, is relevant to the Multiple User Classes traffic assignment problems considered in chapter 5 where the market segmentation aggregation technique is applied within the equilibrium framework. The market segmentation aggregation method consists of dividing the population of choice makers in sets (market segments) with homogenous characteristics and summing the results of the choice model for each set after they have been referred to the size of the sets. In traffic assignment models this is done as a part of the equilibrium calculations, as detailed in chapter 5, so no further aggregation calculation is needed.

2.5 Conclusions

This chapter has introduced the random utility modelling framework, on which many choice models are based and has outlined the main characteristics of the Multinomial Probit model, the model on which the work developed in this thesis is based, as well

as those of the most important alternative choice models currently in the transportation literature.

The review has underlined the flexibility of the MNP model, which is also theoretically appealing as it is underpinned by the Central Limit Theorem, and in particular its suitability for route choice modelling and its ability to accommodate a wide range of substitution patterns. Other choice models have been reviewed (MNL, NL, GEV, CNL, PCL, C-logit, ML) amongst which there are several closed form choice models that have been recently devised and that extend the modelling capabilities of the models of the logit family. It has been noted that similar capabilities are also provided, and in some cases represented in a perhaps more intuitive way by the MNP.

The Mixed Logit model, which has recently received much attention in transportation and econometrics, provides high flexibility similarly to the MNP but can also accommodate non-Normal disturbances which is of advantage when random terms must be strictly of one sign.

Both the MNP and the ML model, along with the high flexibility, share the impossibility to write in a closed form their choice functions. In fact, their application has been recently favoured by developments in simulation methods. However, the MNP choice function can also be approximated analytically with numerical integration and approximation methods that are discussed in chapter 3. In fact the rest of the thesis explores the use of such MNP analytical solution methods, that provide repeatable results, and their use as the basis for optimised methods for traffic assignment and model calibration. The use of analytical approximations to non-closed form choice models can be seen as a different approach to obtaining tractable and very flexible choice models, as opposed to the approach of extending the capabilities of closed form logit-related models.

The chapter closed with a short review of the stages in the application of a choice model to introduce equilibrium forecasting, used in traffic assignment, and model calibration, which are the problems examined respectively in chapter 4 and 5 and in chapter 6.

3. METHODS FOR THE EVALUATION OF THE MULTINOMIAL PROBIT CHOICE FUNCTION

3.1 Introduction

The present chapter discusses the different methods that can be used to evaluate the Multinomial Probit (MNP) choice function. The problem is introduced expanding on the discussion of the MNP given in the previous chapter, showing the alternative formulations of the MNP choice function and mentioning some applications of the MNP integral and of the related MVN integral in disciplines different from transportation.

A brief account of the most important simulation methods for the solution of the MNP choice function is given but most of the chapter is devoted to analytical methods. The numerical integration method of Genz is introduced and a number of approximation methods are treated in detail explaining the principles on which they are based and assessing their accuracy and computational cost. Techniques to improve the accuracy of the results of some approximation methods are also investigated.

The next section, besides introducing the problem of evaluating the MNP choice function, sets out the notation used throughout the rest of the chapter.

3.2 The Multinomial Probit Choice Function

The choice function is the function expressing the probability that an option in the choice set is chosen according to a given choice model. The multinomial probit (MNP) choice model describes the choice amongst a set of J mutually exclusive options whose utility \mathbf{U} is assumed to be distributed $MVN(\mathbf{V}, \mathbf{\Sigma})$ with:

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_i \\ \dots \\ V_J \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1i}^2 & \dots & \sigma_{1J}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2i}^2 & \dots & \sigma_{2J}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sigma_{i1}^2 & \sigma_{i2}^2 & \dots & \sigma_{ii}^2 & \dots & \sigma_{iJ}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sigma_{J1}^2 & \sigma_{J2}^2 & \dots & \sigma_{Ji}^2 & \dots & \sigma_{JJ}^2 \end{bmatrix} \quad (3.1)$$

Given the distribution of the utilities (3.1), the MNP choice function can be written in several alternative but equivalent forms. It can be written using directly (3.1) as the probability P_i that the utility U_i of an option i is the largest among those of the J options in the choice set:

$$P_i(\mathbf{V}, \Sigma) = \quad (3.2)$$

$$= \int_{u_1=-\infty}^{u_1} \int_{u_2=-\infty}^{u_2} \dots \int_{u_i=-\infty}^{\infty} \dots \int_{u_j=-\infty}^{u_j} \left[(2\pi)^J |\Sigma| \right]^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{U} - \mathbf{V})^T \Sigma^{-1} (\mathbf{U} - \mathbf{V}) \right] du_1 du_2 \dots du_i \dots du_j$$

The choice function for any option i can also be written reducing of one the dimension of the problem by considering the $J-1$ dimensional distribution of the utilities in difference with its utility $\mathbf{Z}_{(i)} = \text{MVN}(\mathbf{W}_{(i)}, \mathbf{\Omega}_{(i)})$. $\mathbf{Z}_{(i)}$ is still MVN distributed as MVN distributions are invariant under summation (see *e.g.* Johnson and Kotz, 1972). Its mean vector and covariance matrix are:

$$\mathbf{W}_{(i)} = \Delta_{(i)} \mathbf{V} = \begin{bmatrix} W_{1(i)} \\ W_{2(i)} \\ \dots \\ W_{i-1(i)} \\ W_{i+1(i)} \\ \dots \\ W_{J(i)} \end{bmatrix} = \begin{bmatrix} V_1 - V_i \\ V_2 - V_i \\ \dots \\ V_{i-1} - V_i \\ V_{i+1} - V_i \\ \dots \\ V_J - V_i \end{bmatrix} \quad (3.3a)$$

$$\mathbf{\Omega}_{(i)} = \mathbf{\Delta}_{(i)} \mathbf{\Sigma} \mathbf{\Delta}_{(i)}^T = \begin{bmatrix} \omega_{1,1}^2 & \omega_{1,2}^2 & \dots & \omega_{1,j}^2 & \dots & \omega_{1,J-1}^2 \\ \omega_{2,1}^2 & \omega_{2,2}^2 & \dots & \omega_{2,j}^2 & \dots & \omega_{2,J-1}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega_{j,1}^2 & \omega_{j,2}^2 & \dots & \omega_{j,j}^2 & \dots & \omega_{j,J-1}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega_{J-1,1}^2 & \omega_{J-1,2}^2 & \dots & \omega_{J-1,j}^2 & \dots & \omega_{J-1,J-1}^2 \end{bmatrix} \quad (3.3b)$$

(the subscript referring to the reference alternative is omitted from the $\omega_{j,k}^2$ to simplify the notation). The matrix $\mathbf{\Delta}_{(i)}$ used above to obtain $\mathbf{W}_{(i)}$ and $\mathbf{\Omega}_{(i)}$ has been first proposed by Daganzo (1979) and is a $J-1 \times J$ matrix formed as an identity matrix of dimension $J-1 \times J-1$ with an added column of -1 in position i , the position of the utility used as reference, and whose choice probability is calculated.

Then the probability that the utility of option i is the largest amongst those of the J options in the choice set corresponds to the probability that the difference between each utility, except that of option i , and that of option i is not greater than 0. This is written expanding on (2.5) as:

$$P_i = P(V_i + \varepsilon_i > V_j + \varepsilon_j) = P(\varepsilon_j - \varepsilon_i < V_i - V_j) = P(V_j + \varepsilon_j - V_i - \varepsilon_i < 0); \quad (3.4)$$

$\forall j \neq i; i, j \in J$

Or writing the integral explicitly:

$$P_i(\mathbf{W}_{(i)}, \mathbf{\Omega}_{(i)}) = \Phi_{J-1}(\mathbf{0}, \mathbf{W}_{(i)}, \mathbf{\Omega}_{(i)}) = \quad (3.5)$$

$$= \int_{z_1=-\infty}^0 \int_{z_2=-\infty}^0 \dots \int_{z_{j-1}=-\infty}^0 \int_{z_j=-\infty}^0 \left[(2\pi)^{J-1} |\mathbf{\Omega}_{(i)}| \right]^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{Z}_{(i)} - \mathbf{W}_{(i)})^T \mathbf{\Omega}_{(i)}^{-1} (\mathbf{Z}_{(i)} - \mathbf{W}_{(i)}) \right] dz_1 dz_2 \dots dz_{j-1} dz_j$$

where $\Phi_n(\cdot)$ indicates the MVN integral of dimension n .

A further equivalent way of writing this integral, that is the starting point of some of the calculation methods reviewed in the next sections, is by using the MVN distribution of the utilities in difference w.r.t. that of option i in standardised form.

The standardised utility difference distribution $\mathbf{Y}_{(i)}$ is then distributed $MVN(\mathbf{0}, \mathbf{P}_{(i)})$ with:

$$\mathbf{P}_{(i)} = \begin{bmatrix} 1 & \frac{\omega_{1,2}^2}{\omega_{1,1}\omega_{2,2}} & \dots & \frac{\omega_{1,j}^2}{\omega_{1,1}\omega_{j,j}} & \dots & \frac{\omega_{1,J-1}^2}{\omega_{1,1}\omega_{J-1,J-1}} \\ \frac{\omega_{2,1}^2}{\omega_{2,2}\omega_{1,1}} & 1 & \dots & \frac{\omega_{2,j}^2}{\omega_{2,2}\omega_{j,j}} & \dots & \frac{\omega_{2,J-1}^2}{\omega_{2,2}\omega_{J-1,J-1}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\omega_{j,1}^2}{\omega_{j,j}\omega_{1,1}} & \frac{\omega_{j,2}^2}{\omega_{j,j}\omega_{2,2}} & \dots & 1 & \dots & \frac{\omega_{j,J-1}^2}{\omega_{j,j}\omega_{J-1,J-1}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\omega_{J-1,1}^2}{\omega_{J-1,J-1}\omega_{1,1}} & \frac{\omega_{J-1,2}^2}{\omega_{J-1,J-1}\omega_{2,2}} & \dots & \frac{\omega_{J-1,j}^2}{\omega_{J-1,J-1}\omega_{j,j}} & \dots & 1 \end{bmatrix} = \quad (3.6)$$

$$= \begin{bmatrix} 1 & \rho_{1,2} & \dots & \rho_{1,j} & \dots & \rho_{1,J-1} \\ \rho_{2,1} & 1 & \dots & \rho_{2,j} & \dots & \rho_{2,J-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \rho_{j,1} & \rho_{j,2} & \dots & 1 & \dots & \rho_{j,J-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \rho_{J-1,1} & \dots & \rho_{J-1,j} & \dots & \dots & 1 \end{bmatrix}$$

Due to the standardisation, the mean of each variate is 0 and the upper limits of integration are no longer 0 but are defined by the vector β :

$$\beta_{(i)} = \begin{bmatrix} \beta_{1,(i)} \\ \beta_{2,(i)} \\ \dots \\ \beta_{J-1,(i)} \end{bmatrix} = \begin{bmatrix} \frac{-W_{1,(i)}}{\omega_{1,1}} \\ \frac{-W_{2,(i)}}{\omega_{2,2}} \\ \dots \\ \frac{-W_{J-1,(i)}}{\omega_{J-1,J-1}} \end{bmatrix} = \begin{bmatrix} \frac{-V_1 + V_i}{\omega_{1,1}} \\ \frac{-V_2 + V_i}{\omega_{2,2}} \\ \dots \\ \frac{-V_J + V_i}{\omega_{J-1,J-1}} \end{bmatrix} \quad (3.7)$$

The MVN integral can then be written as:

$$P_i(\beta_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) = \Phi_{J-1}(\beta_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) = \quad (3.8)$$

$$= \int_{y_1=-\infty}^{\beta_{1(i)}} \int_{y_2=-\infty}^{\beta_{2(i)}} \dots \int_{y_{j-1}=-\infty}^{\beta_{j-1(i)}} \int_{y_j=-\infty}^{\beta_{j(i)}} [(2\pi)^{J-1} |\mathbf{P}_{(i)}|]^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{Y}_{(i)})^T \mathbf{P}_{(i)}^{-1}(\mathbf{Y}_{(i)})\right] dy_1 dy_2 \dots dy_j \dots dy_{J-1}$$

To explain some of the methods introduced in this chapter it is also useful to rewrite (3.5) or (3.8) in terms of independent Normal variates using the Cholesky factorisation of the covariance matrix of the integrand. In particular, focusing on (3.8), $\mathbf{P}_{(i)}$ can be rewritten as:

$$\mathbf{P}_{(i)} = \mathbf{C}\mathbf{C}^T \quad (3.9)$$

where \mathbf{C} is the Cholesky decomposition of $\mathbf{P}_{(i)}$, a uniquely defined lower triangular matrix.

The variate $\mathbf{Y}_{(i)}$ can then be written as a function of a vector \mathbf{X} of unidimensional standard Normal variates with the same dimension of $\mathbf{Y}_{(i)}$ as:

$$\mathbf{Y}_{(i)} = \mathbf{C}\mathbf{X} \quad (3.10)$$

Following *e.g.* Genz (1992) and considering (3.10), the product of matrices in the exponential of the integrand of (3.8) can be rewritten as:

$$\mathbf{Y}_{(i)}^T \mathbf{P}_{(i)}^{-1} \mathbf{Y}_{(i)} = \mathbf{X}^T \mathbf{C}^T \mathbf{C}^{-T} \mathbf{C}^{-1} \mathbf{C} \mathbf{X} = \mathbf{X}^T \mathbf{X} \quad (3.11)$$

and, substituting this expression, (3.8) can be rewritten as:

$$\begin{aligned} P_i(\boldsymbol{\beta}_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) &= \Phi_{J-1}(\boldsymbol{\beta}_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) = \\ &= \int_{x_1=-\infty}^{\beta'_1(\mathbf{x})} \int_{x_2=-\infty}^{\beta'_2(\mathbf{x})} \dots \int_{x_{j-1}=-\infty}^{\beta'_{j-1}(\mathbf{x})} \int_{x_j=-\infty}^{\beta'_j(\mathbf{x})} (2\pi)^{-\frac{J-1}{2}} \exp\left[-\frac{1}{2} \mathbf{X}^T \mathbf{X}\right] dx_1 dx_2 \dots dx_j \dots dx_{J-1} = \\ &= (2\pi)^{-\frac{J-1}{2}} \int_{x_1=-\infty}^{\beta'_1(\mathbf{x})} \exp\left(-\frac{x_1^2}{2}\right) \int_{x_2=-\infty}^{\beta'_2(\mathbf{x})} \exp\left(-\frac{x_2^2}{2}\right) \dots \int_{x_{j-1}=-\infty}^{\beta'_{j-1}(\mathbf{x})} \exp\left(-\frac{x_{j-1}^2}{2}\right) dx_1 dx_2 \dots dx_{j-1} = \end{aligned} \quad (3.12a)$$

where:

$$\beta'_j(\mathbf{X}) = \left(\beta_{j(i)} - \sum_{k=1}^{j-1} c_{jk} x_k \right) / c_{jj} \quad (3.12b)$$

and c_{jk} is the entry of position jk in the Cholesky matrix \mathbf{C} .

The main difficulty of working with the probit model is that the above equivalent integrals (3.2), (3.5), (3.8), (3.12) cannot be rewritten in closed form. Therefore their evaluation can only be approximate and is generally not straightforward.

A univariate Normal integral can be calculated with generally available computer routines (*e.g.* the methods in Abramowitz and Stegun, 1972). Bivariate Normal integrals can also be calculated with routines available in the computational literature (*e.g.* Donnelly, 1973). The test cases considered in the remainder of this chapter include MNP models with up to 15 options but the description of the methods reported here is more general and refers to techniques for solving MNP problems and MVN integrals of even larger dimension.

There are a limited number of cases in which the evaluation of a MVN integral can be carried out rather simply. For instance, when the covariance matrix has a product correlation structure or when all the covariances or the correlations are equal, the multivariate integral reduces to a one-dimensional one that can be solved with standard numerical integration methods. These cases are illustrated *e.g.* in Johnson and Kotz (1972) and in Tong (1990) but, since they are particular instances of the problem that are not generally encountered in transportation, they are not considered further here.

Several methods to solve approximately the probit choice function in general cases have been put forward in the literature, and they can be broadly divided into simulation and analytical methods. Analytical methods can be further divided in methods using numerical integration and analytical approximations.

Simulation reduces the problem to the evaluation of the frequency with which a suitable randomly drawn vector satisfies the conditions defining the problem. Numerical integration calculates the hypervolume defining an integral, the MVN integral here, by approximating the surfaces with suitably defined polynomial functions and refining the determination of such functions until a satisfactory precision is reached. Analytical approximation methods reduce the problem to a

simpler one, or more specifically for the present case, to a series of simpler ones obtained making some approximate assumptions.

Methods of each type have been used in transportation but, recently, simulation methods have received most of the interest both in research and in practical applications. This could partly be due to the availability of commercial software for simulated MNP traffic assignment and to the availability of MNP calibration code with one of the most recent simulation techniques, the GHK simulator (Borsch-Supan and Hajivassiliou, 1993), illustrated in the following.

Numerical integration of (3.5) or (3.8) is generally disregarded for practical applications since it is usually considered feasible only for small dimensions of the problem (3-4 options) within which it has had limited use in the past (see *e.g.* Andrews and Langdon, 1976; Hausman and Wise, 1978). Mention should also be made of a tabulation method, based on numerical integration, proposed for up to trinomial probit models by Sparmann *et al.* (1983) (see also Sheffi *et al.*, 1982).

Approximation methods have had limited use perhaps as an effect of the lack of information about their accuracy or as an effect of the low precision expected by the best known method, due to Clark (1961), after the results published by Horowitz *et al.* (1982).

Because of their different nature, each type of method, and each particular method, has a different performance in terms of accuracy of the results and computational time necessary to solve the problem.

The accuracy of simulation methods may depend, amongst other factors, by the suitability of the method for the problem at hand (see *e.g.* the comment in the literature and reported in section 3.3 on the crude frequency simulator method for the MNP choice problem) and from the number of repetitions of the random vector sampling. The result will also depend, at least to some extent, on the seed used to generate pseudo-random numbers.

The accuracy of numerical integration methods depends on the suitability of the particular method for the problem to solve (see *e.g.* the points on the difficulty encountered by many integration methods in evaluating MVN integrals, made by Genz (1992) and reported in section 3.4) and on the ability of the chosen polynomial approximation to replicate the surface of the integrand function.

The accuracy of the approximations depends on the validity of the assumptions on which they are based.

Generally the accuracy of a method varies with the problem at hand. For instance with the magnitude of the result (*e.g.* in crude frequency simulation, see section 3.3) or with the complexity of the problem (*e.g.* with the correlation of the variates in approximation methods).

The computational expense depends in all cases on the dimension of the problem and on the complexity of the operations required. Moreover, for simulation methods it increases with the number of replications of the random vector sampling and for numerical integration methods with the precision required.

The focus of this thesis is on analytical methods, and particularly on approximations, on the grounds that they give repeatable results and allow the development of the optimised methods considered in chapter 4 and 5 for traffic assignment. Simulation implicitly does not ensure repeatability of results (in fact, simulation results depend, to an extent, on the simulation convergence and on the pseudo-random numbers employed). Van Vuren (1994), discussing the effect of using simulated probit in traffic assignment (with a method related to crude frequency simulation, mentioned in the remainder of this chapter), remarked that since transportation models' results are typically used in comparisons amongst different scenarios, "noise" and non-repeatability typical of simulation should be avoided. Also, trying to reduce these influences by averaging the results of several runs may entail long computational times, longer than those usually accepted in transportation practice.

Integrals of the form of the MNP choice function (3.2) or the corresponding MVN integrals (3.5) and (3.8), expressing the multivariate Normal cumulative distribution, recur in a number of applications in different disciplines. For instance, the MNP integral (3.2) is used to express choices in transportation (see *e.g.* Daganzo, 1979; Ben-Akiva and Lerman, 1985), econometrics (see *e.g.* Lerman and Manski, 1981; Geweke *et al.*, 1994), political studies (see *e.g.* Alvarez and Nagler, 1994, Alvarez *et al.*, 1999), policy assessment (see *e.g.* Bolduc *et al.*, 1996; Frölich *et al.*, 2000). It is also used in activity scheduling (PERT network programming: see Clark, 1961; Guo *et al.*, 2001). The MVN integral (3.5) or (3.8) is used *e.g.* in civil engineering to study limit states of structures (see *e.g.* Melchers, 1999), in communications for signal processing (see Thompson, 1974; Pattison and Gossink, 1999), in biometrics for the study of inheritance of traits or diseases (see *e.g.* Rice *et al.*, 1979), and, in general, in statistical applications it is used for parameter estimation, hypothesis testing, classification and discriminant analysis (Tong, 1990).

There is a rather good exchange of methods between transportation and econometrics. However, in other fields where the MVN integral is used, besides being sometimes dealt with using well known simulation methods, it is solved in some cases with analytical techniques that have received little or no attention for transportation applications. Some of those, and also some methods already used in transportation, are considered and tested in the remainder of this chapter to assess their suitability for transportation applications.

First the simulation methods are briefly reviewed with particular focus on the crude frequency simulator (Lerman and Manski, 1981) and the GHK probability simulator (Borsch-Supan and Hajivassiliou, 1993). Then, a survey of some numerical integration methods appeared in the literature for the solution of the MVN integral precedes an introduction to the method of Genz (1992, 1993), used in this work. A detailed description of the several approximation methods investigated follows.

3.3 Calculation of the MNP Choice Function by Simulation

In recent years most of the research and applications of the MNP model have been based on simulation methods. Research oriented calibration software is available on the internet (see the code for the GHK method, developed by Vassilis Hajivassiliou, used *e.g.* by Munizaga *et al.*, 2000). Simulation is also the only MNP calculation method included in commercial software for traffic assignment (*e.g.* SATURN, Van Vliet and Hall, 1993; TRIPS, MVA, 1994).

The simplest way to solve the MNP by simulation has been put forward by Lerman and Manski (1981), and is actually a method that can be adapted to choice models with any distribution of utilities. It has become generally known in econometrics as “crude frequency simulation” (CFS). The principle on which it relies is to draw the utilities of the options in a choice set from their distribution a number of times, recording each time which option has the largest utility and finally approximating the choice probability of each option by its resulting relative choice frequency. Alternatively, only the times that the alternative of interest is chosen are recorded to obtain its choice frequency.

This frequency simulator is applied to (3.2) or (3.5) using the Cholesky factorisation of the covariance matrix as in (3.10). For instance to solve (3.2), at each simulation draw, an instance of the vector \mathbf{X} of independent univariate Normal variates is drawn, then it is multiplied by the lower triangular matrix resulting from the Cholesky factorisation of Σ , thus being transformed in the random vector of the utilities \mathbf{U} . This, summed to their deterministic part \mathbf{V} , gives the utilities. These are compared and the option with the higher utility is recorded as chosen.

Whilst simple to implement, the CFS has some drawbacks, remarked on in Daganzo *et al.* (1977), Sheffi *et al.* (1982), Geweke *et al.* (1994), due to difficulties with low probability alternatives and to the fact that the frequencies are not continuous. The first sort of problem is due to the possibility of recording no successes, that is no cases in which an option with actual low probability is chosen, during the simulation process (see Sheffi *et al.*, 1982). Continuing the simulation until a satisfactory value is obtained for all options and using an adjustment to account for low choice probabilities introduce bias in the results (Sheffi *et al.*, 1982). Also, the error of the

results depends on the number of successes, making the method less precise in case of small probabilities. The lack of continuity of the derivatives w.r.t. the parameters of the choice model originates from the frequency of successes being a step function and thus having points of discontinuity. This latter problem is relevant especially to the use of the method in model calibration. As any simulation method, the CFS gives results that are not strictly repeatable unless the same seed for pseudo-random number generation is used.

Notwithstanding these drawbacks, the CFS still enjoys much attention (see *e.g.* Liu and Mahmassani, 2000, and Garrido and Mahmassani, 2000, who used the program of Lam, 1991). It should also be mentioned that the crude frequency simulation method had independently been used also for solving MVN integrals outside transportation and econometrics: Thompson (1974), for instance, put it forward for signal processing problems for communications applications. The method is still used nowadays in the same field (see Pattison and Gossink, 1999).

The CFS is also the method used in traffic assignment to solve the MNP choice function but in that case the procedure to simulate the utilities is different from that reported above as it exploits the network structure of the problem and the assumption that the utilities of the routes between an origin and a destination, which are the choice options, are given by the sum of the utilities of the links that they traverse. With the assumption that the link utilities are independently Normally distributed and with the additivity assumption just mentioned, the utilities of the routes between an origin and a destination are, as a result, MVN distributed. In practice a simulation draw is obtained by drawing the utilities of the links of the network. The utilities of the routes do not need to be calculated explicitly: a shortest route algorithm characterises the route with the highest utility (in the assignment case actually the one with minimum cost or disutility) that is then the option chosen in the simulation draw. This process is repeated a number of times and the traffic between an origin-destination pair is loaded according to the frequency of choice of the paths. On this concept are based the approximated probit stochastic network loading method of Burrell (1968), that uses uniform, rather than Normal, link utilities, and the probit stochastic network loading method by Daganzo and Sheffi (1977) and Sheffi and Powell (1981). Methods based on this principle (although not performing a total

solution of the choice function, as illustrated in chapter 4) have been used by Sheffi and Powell (1981, 1982) and Nielsen (1996) for the calculation of the stochastic user equilibrium traffic pattern. Further versions of these methods have more recently been used in research (see *e.g.* Nielsen *et al.* 1998; Cantarella and Binetti, 1998) and in practical applications as they are available in commercial software for assignment.

A number of alternative MNP simulation methods have been developed in econometrics or other disciplines. Several such methods are due to Genz (1992), and use the transformation he used also to propose his numerical integration method (see section 3.4). Deak (1980, 1986) proposed a method based on the problem expressed using the Cholesky factorisation of the covariance matrix and on a transformation to spherical coordinates. Somerville (1998) proposed a method partially based on simulation and partially involving numerical integration. He built on a previous simulation method (Somerville and Wang, 1994) that obtained the integral of a MVN distribution over a convex integration region including the origin by working in the space of uncorrelated variates and sampling the distance of the integration limit in a number of randomly chosen directions. The method proposed by Somerville (1998) uses numerical integration to calculate the integral of a function of the distribution of the distance of the origin of the reference system and the limits of the integration region, obtained by sampling such distance in a number of random directions. None of these methods has been used in transportation.

A notable attempt to produce a simulator with results continuous in the parameters is due to McFadden (1989) who developed a smooth MNP probability simulator that processes the utilities sampled as in the CFS through a logit function. The development of that simulator has led to the resurgence of interest in the Mixed Logit model, introduced in chapter 2.

At present the most important, and widely used, MNP simulation method is the Geweke-Hajivassiliou-Keane (GHK) probability simulator (see *e.g.* Borsch-Supan and Hajivassiliou, 1993). It is based on expressing the distribution of the utilities in difference with respect to the option of interest, as in (3.8). The resulting n

dimensional MVN integral is then transformed as in (3.12), employing the Cholesky factorisation of the covariance matrix, and the integral (3.12) is solved by applying simulation to the n conditional univariate Normal integrals into which it can be decomposed.

To explain how the GHK simulator works, it is useful to reconsider (3.12) and its limits of integration, whose general expression is in (3.12b). Rewriting them explicitly it is clear that the first one is independent of the variate \mathbf{X} :

$$\beta'_1(\mathbf{X}) = \beta'_1 = \beta_1/c_{11} \quad (3.13)$$

whilst the limit of integration of any other variate, say j , depends on the variates of the vector \mathbf{X} up to that of place $j-1$. Thus for instance β_2 depends on X_1 :

$$\beta'_2(\mathbf{X}) = \beta'_2(X_1) = (\beta_2 - c_{21}X_1)/c_{22} \quad (3.14)$$

Similarly β_3 depends on X_1 and X_2 :

$$\beta'_3(\mathbf{X}) = \beta'_3(X_1, X_2) = (\beta_3 - c_{31}X_1 - c_{32}X_2)/c_{33} \quad (3.15)$$

and so on.

Considering this, the integral (3.12) can also be written:

$$\begin{aligned} P_i(\boldsymbol{\beta}_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) &= \\ &= (2\pi)^{-\frac{J-1}{2}} \int_{x_1=-\infty}^{\beta'_1(\mathbf{X})} \exp\left(-\frac{x_1^2}{2}\right) \int_{x_2=-\infty}^{\beta'_2(\mathbf{X})} \exp\left(-\frac{x_2^2}{2}\right) \dots \int_{x_{J-1}=-\infty}^{\beta'_{J-1}(\mathbf{X})} \exp\left(-\frac{x_{J-1}^2}{2}\right) dx_1 dx_2 \dots dx_{J-1} = \end{aligned} \quad (3.16)$$

$$= P(X_1 < \beta'_1; X_2 < \beta'_2(X_1); \dots; X_j < \beta'_j(X_1, X_2, \dots, X_{j-1}); X_{J-1} < \beta'_{J-1}(X_1, X_2, \dots, X_{J-2}))$$

and can also be expressed as a product of marginal conditional probabilities:

$$\begin{aligned} P_i &= P(X_1 < \beta'_1) \cdot P(X_2 < \beta'_2(X_1) | X_1 < \beta'_1) \cdot P(X_3 < \beta'_3(X_1, X_2) | X_1 < \beta'_1; X_2 < \beta'_2) \cdot \\ &\quad \cdot P(X_{J-1} < \beta'_{J-1}(X_1, X_2, X_{J-2}) | X_1 < \beta'_1; X_2 < \beta'_2; X_{J-2} < \beta'_{J-2}) \end{aligned} \quad (3.17)$$

Since the variates in \mathbf{X} are Normally distributed, this can be further re-written as:

$$P_i = \Phi(\beta'_1) \cdot \Phi(\beta'_2(X_1)) \cdot \Phi(\beta'_3(X_1, X_2)) \cdot \dots \cdot \Phi(\beta'_{J-1}(X_1, X_2, X_{J-2})) \quad (3.18)$$

The GHK simulator solves (3.8) by taking a number N of draws from \mathbf{X} for which (3.18) is satisfied and assuming that P_i is:

$$P_i \cong \frac{1}{N} \sum_{n=1}^N f_{(n)i} \quad (3.19)$$

where $f_{(n)i}$ is the value assumed by (3.18) for the n th draw of the random vector \mathbf{X} :

$$f_{(n)i} = \Phi(\beta'_1) \cdot \Phi(\beta'_2(x_{(n)1})) \cdot \Phi(\beta'_3(x_{(n)1}, x_{(n)2})) \cdot \dots \cdot \Phi(\beta'_{J-1}(x_{(n)1}, x_{(n)2}, \dots, x_{(n)J-2})) \quad (3.20)$$

The conditional integration limits are obtained by drawing from an inverse truncated Normal. For instance β'_2 conditional on $X_1 < \beta'_1$ is obtained by substituting in (3.14) a value x_1 drawn from the distribution of X_1 truncated at β'_1 that is computed as $\Phi^{-1}(u\Phi(\beta'_1))$ where u is drawn from a uniform variate defined over $[0,1]$.

The GHK, being a probability simulator rather than a frequency simulator, does not give the problems with small probabilities encountered with the CFS. Moreover, as the result of the simulation is not a step function but a continuous one, it is continuous in the parameters. There remains, as in any simulation based method, the issue of result repeatability.

The GHK probability simulator has been used in a number of recent transportation studies. Munizaga *et al.* (1997) and Munizaga *et al.* (2000) used it in a calibration program to investigate the ability of the probit model to represent heteroscedasticity in the utilities' distribution. Bolduc (1999) proposed an improved calibration program based on the GHK simulator including for the first time the analytical calculation of the MNP likelihood function derivatives, obtained exploiting the structure of the simulator.

3.4 Calculation of the MNP Choice Function by Numerical Integration

Numerical integration methods for calculating the MNP choice function are discussed separately from approximation methods here although, strictly, both these sorts of methods are analytical as opposed to simulation ones. The methods in the literature are mainly from outside transportation and, in particular, no application of this type of method has been proposed for traffic assignment.

An example of an early attempt to solve MVN integrals by numerical integration can be found in Milton (1972) who used the Simpson rule. Andrews and Langdon (1976) considered the trinomial probit case and the associated bivariate Normal integral reducing it, with a series of variable transformations, to an integral in one dimension, that they solved numerically. Hausman and Wise (1978) also worked on the trivariate probit problem and, starting from the related bivariate Normal integral, suggested a transformation to write it as an integral in one dimension, solved by numerical integration.

Other numerical integration methods have been developed more recently for the evaluation of the MVN integral (*e.g.* Schervish, 1984, who used locally adaptive integration based on a Simpson rule; Drezner, 1992, who used a Gaussian integration rule). They have been reviewed *e.g.* in Genz (1992) and appear to be of little interest for transportation applications as they require long calculation times and can be applied only to problems of limited dimension. For instance in Genz (1992), where comparison of different MVN evaluation methods are carried out on problems of dimension up to 20, the method of Schervish (1984) is not used for more than 4 dimensions because it requires too long a calculation time.

Although for trinomial problems the numerical integration calculations for the MNP choice function have been reduced to a univariate integral, for larger dimensions of the problem there are difficulties for applying multidimensional numerical integration routines that are sometimes referred in the literature as the “curse of dimensionality”. Genz (1992) explained that the difficulty of applying numerical integration to the MVN integral is due to the peaked shape of the integrand function and to the infinity limits of integration. The former requires suitable integration methods and the latter either a suitable truncation of the integration domain or a transformation of the integrand.

On the basis of these considerations, Genz has proposed a new numerical integration method (Berntsen *et al.*, 1991; Genz, 1992 and 1993), not yet introduced in transportation, that can solve a MVN integral for a large dimension of the problem and to a specified degree of precision in relatively short calculation times. Genz’s

method relies on a functional transformation presented in Genz (1992, 1993) that changes the MVN integral expressed as in (3.8) into one of one less dimension over a unit hypercube, so that the infinite integration limits are eliminated. This new formulation of the problem is then solved using an algorithm for subregion adaptive numerical integration well known in the relevant literature and developed by Berntsen *et al.* (1991) that can recognise the peakedness of the integrand. Adaptive numerical integration algorithms evaluate the integrand at points whose number and position depends on the integrand shape, as opposed to non-adaptive integration methods, where only the number of evaluation points is related to the nature of the integrand while their position is determined with a fixed rule.

In the method of Genz (1992, 1993), the integral (3.8) is rewritten using the Cholesky decomposition of the integrand covariance matrix as in (3.12) that is reported here:

$$P_i(\boldsymbol{\beta}_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) = \Phi_{J-1}(\boldsymbol{\beta}_{(i)}, \mathbf{0}, \mathbf{P}_{(i)}) = \quad (3.12a)$$

$$= (2\pi)^{-\frac{J-1}{2}} \int_{x_1=-\infty}^{\beta_1(\mathbf{x})} \exp\left(-\frac{x_1^2}{2}\right) \int_{x_2=-\infty}^{\beta_2(\mathbf{x})} \exp\left(-\frac{x_2^2}{2}\right) \dots \int_{x_{J-1}=-\infty}^{\beta_{J-1}(\mathbf{x})} \exp\left(-\frac{x_{J-1}^2}{2}\right) dx_1 dx_2 \dots dx_{J-1}$$

Genz suggested that (3.12) should be further transformed by letting:

$$x_j = \Phi^{-1}(z_j) \text{ for all } j \quad (3.21)$$

Therefore:

$$\Phi(x_j) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_j} \exp\left(-\frac{s^2}{2}\right) ds \quad (3.22)$$

From which it can be derived that:

$$dz_j = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_j^2}{2}\right) dx_j \quad (3.23)$$

Considering (3.23), (3.12a) can be rewritten with a change of variable as:

$$P_i = \int_0^{e_1} \int_0^{e_2(z_1)} \dots \int_0^{e_{j-1}(z_1, z_2, \dots, z_{j-2})} dz_1 dz_2 \dots dz_{j-1} \quad (3.24)$$

where:

$$e_j(z_1, \dots, z_{j-1}) = \Phi \left(\left(\beta'_j - \sum_{k=1}^{j-1} c_{jk} \Phi^{-1}(z_k) \right) / c_{jj} \right) \quad (3.25)$$

Genz applied a further transformation to simplify the integration region for use with numerical integration software by letting:

$$z_j = e_j w_j \quad \text{for all } j \quad (3.26)$$

therefore:

$$dz_j = e_j dw_j \quad (3.27)$$

$$P_i = e_1 \int_0^1 e_2(\mathbf{w}) \dots \int_0^1 e_{j-1}(\mathbf{w}) \int_0^1 dw_1 dw_2 \dots dw_{j-1} \quad (3.28)$$

where:

$$e_j(\mathbf{w}) = \Phi \left(\left(\beta'_j - \sum_{k=1}^{j-1} c_{jk} \Phi^{-1}(e_k(\mathbf{w}) w_k) \right) / c_{jj} \right) \quad (3.29)$$

The dimension of the original MVN integral (3.12) is reduced by one since the innermost integral in (3.28) is equal to one and the problem becomes:

$$P_i = \int_0^1 \int_0^1 \dots \int_0^1 e_1 e_2(w_1) \dots e_{j-2}(w_1, w_2, \dots, w_{j-3}) dw_1 dw_2 \dots dw_{j-2} \quad (3.30)$$

It should be noted that this presentation of the transformation of Genz (1992, 1993) refers to single sided integrals. The more general case of double sided integrals, possibly including also some infinite integration limits, is illustrated in Genz (1992).

Following the sorting of variables proposed by Schervish (1984), Genz suggested that the integration is more easily and quickly computed if the variates are ordered so that the innermost ones have the largest upper integration limits (referred to the integration limits appearing in 3.8) so that their values are closer to one. This way the overall variation of the integrand is reduced and the integration algorithm works more efficiently (Genz 1992, 1993).

Applying Genz's method to the calculation of the MNP model, the choice probability of each option in a choice set is obtained by calculating the relevant MVN integral in difference (3.8).

3.5 Calculation of the MNP Choice Function Using Analytical Approximations

3.5.1 Introduction

This section considers a number of numerical approximations methods for solving the MNP choice function by transforming the MNP or MVN integral in a series of simpler operations not based, or not completely based, on numerical integration.

The approximations considered resulted from a literature search in different disciplines, where the MNP or the MVN integral is employed. The paragraphs describing each approximation give a brief account of the use of the approximations in the disciplines where they have been developed along with the relevant references. A short description of the principles at the basis of the approximation precedes a full description, formulae and calculation details. All methods rely on established algorithms for calculating univariate Normal integrals that are readily available in the literature (see *e.g.* Abramowitz and Stegun, 1972).

It should be noted that some approximations calculate the choice probabilities separately for each option, therefore they might not sum exactly to one. In these

cases the choice probabilities are normalised to their sum. Alternatively, the value of the choice probabilities for all options except one could be calculated and the difference from 1 could be assigned to the remaining one but this alternative method would arbitrarily treat differently one of the options whilst normalising is preferred since it redistributes the inaccuracies proportionally amongst all the options. When only the choice probability of one alternative is required, as in model calibration, this is sometimes used directly as obtained from the approximation without normalisation to save on the computational cost of calculating the other choice probabilities.

3.5.2 The Approximation of Clark

The approximation of Clark (1961) was originally developed for the solution of PERT programming problems and has been introduced in transportation by Daganzo and Sheffi (1977) and Daganzo *et al.* (1977) respectively for use in traffic assignment and in MNP calibration. It has also recently been used by Maher (1992) and Maher and Hughes (1997a) in traffic assignment.

Clark obtained the formulae for the first four moments of the maximum of two Normal variates, which he suggested to approximate as Normally distributed, and the formula expressing the correlation between that maximum and any third Normal variate. The application to choice models of his method uses only the formulae for the first two moments of the maximum and the formula for the correlation.

If the utility distribution under study is written as in (3.1) and therefore the original utility U_i has mean V_i and variance σ_{ii}^2 and the correlation with a different utility U_j is denoted by ρ_{ij} the maximum of two Normal utilities U_i and U_j can be approximated as a Normal having mean V_{MAX} and variance σ_{MAX}^2 given by:

$$V_{MAX} = v_1 \tag{3.31}$$

$$\sigma_{MAX}^2 = v_2 - v_1^2 \tag{3.32}$$

where:

$$v_1 = V_i \Phi(\gamma_{ij}) + V_j \Phi(-\gamma_{ij}) + \omega_{ij} \varphi(\gamma_{ij}) \quad (3.33)$$

$$v_2 = (V_i + \sigma_{ii}^2) \Phi(\gamma_{ij}) + (V_j + \sigma_{jj}^2) \Phi(-\gamma_{ij}) + (V_i + V_j) \omega_{ij} \varphi(\gamma_{ij}) \quad (3.34)$$

where $\Phi(\cdot)$ is the cumulative standard Normal distribution function, $\varphi(\cdot)$ is the standard Normal density function and:

$$\omega_{ij}^2 = \sigma_{ii}^2 + \sigma_{jj}^2 - 2\sigma_{ii}\sigma_{jj}\rho_{ij} \quad (3.35)$$

$$\gamma_{ij} = \frac{V_i - V_j}{\omega_{ij}} \quad (3.36)$$

Moreover, the expressions:

$$\Phi(\gamma_{ij}) = \Phi\left(\frac{V_i - V_j}{\omega_{ij}}\right) \quad (3.37)$$

$$\Phi(-\gamma_{ij}) = \Phi\left(\frac{V_j - V_i}{\omega_{ij}}\right) \quad (3.38)$$

can be seen as the marginal probabilities of choice respectively of option i and option j .

The correlation between $Max(U_i, U_j)$ and any third variate U_k is then calculated with the following formula:

$$\rho[U_k, Max(U_i, U_j)] = \frac{\sigma_{ik}\rho_{ik}\Phi(\gamma_{ij}) + \sigma_{jk}\rho_{jk}\Phi(-\gamma_{ij})}{\sqrt{(v_2 - v_1^2)}} \quad (3.39)$$

This set of formulae can be used recursively by taking into account one more variate at a time to approximate the maximum of n jointly Normally distributed variates. In fact, they can be used to write (see *e.g.* Sheffi, 1985):

$$Max(U_1, U_2, U_3, \dots, U_n) = Max\{Max[Max(U_1, U_2), U_3], \dots, U_n\} \quad (3.40)$$

The formulae of Clark can be used to solve the MNP integral (3.2) directly or as MVN integral of the utilities in difference (3.4). The description reported here refers to the direct solution of the MNP integral.

There are two possible ways to implement the Clark approximation to solve the MNP choice problem: one is referred to here as “improved Clark” and is correct, whilst the other, called here “simple Clark”, is simpler but is a heuristic.

The improved Clark method consists of approximating the MNP choice function (3.2) for each alternative in the choice set using recursively the formulae of Clark. As the probabilities for each option are calculated separately they might not sum exactly to one. In such a case they are normalised to their sum.

The calculation of each option’s probability is started by finding the mean and variance of the maximum of the utilities of a pair of options in the choice set, that is approximated as Normal, and obtaining its correlation with the utilities of all the remaining options. Then such maximum is considered along with a further utility, their maximum is again approximated as Normal using the formulae of Clark and its correlation with the remaining utilities is calculated. The calculations proceed in this fashion including one more option at a time as in (3.40) above, until the maximum of all the utilities except that of the option whose probability of choice is being calculated is obtained. The probability that the utility of the remaining option, say i , is larger than the maximum of the others, that is equivalent to its MNP integral, is obtained as an unidimensional Normal integral:

$$P_i = \Phi\left(\frac{V_i - V_M}{\omega_{iM}}\right) \quad (3.41)$$

where V_i is the mean utility of option i , V_M is the mean utility of the maximum of all the other options and ω_{iM} is the standard deviation of the difference of the utilities whose mean values appear in the numerator.

Carrying out one further maximum calculation it is possible to approximate as Normal the distribution of the maximum of the utility of all the options in the choice

set thus obtaining also the expected maximum utility of the choice set, known as the satisfaction, which is particularly useful *e.g.* in equilibrium calculations (see chapters 4 and 5).

Alternatively, the simple Clark procedure consists of calculating the MNP integral as in the improved Clark method for one option only and obtaining the choice probabilities for the other options by using the intermediate results in the recursive calculations as conditional probabilities that each option is chosen. In fact, each time a new option is considered in the application of (3.40) its marginal probability of choice and the marginal probability that it is not chosen are calculated according to (3.37) and (3.38). Thus the probability that the last option accounted for is not chosen may be divided between the probability that the penultimate option accounted for is chosen and the alternative event that the maximum of the previous ones is chosen. The latter is in turn similarly divided and the calculations proceed in a similar way until all the options have been assigned a choice probability.

However, as noted by Langdon (1981), the simple Clark method is a heuristic. In fact, when calculating the marginal split between two options any choice whose probability has already been obtained should be accounted for as it conditions the distribution of the utilities of the remaining options (similarly to the procedure in the method of Langdon, see section 3.5.4). Thus, for instance, when splitting the probability that the last option is not chosen, the split calculated as discussed in the previous paragraph is not the correct one to use as it is obtained with utilities that are not conditional on the fact that the last option is not chosen. Similar considerations apply all the times the probability that an option is not chosen is divided. Therefore in the simple Clark method the probability of choice of the option entered last in the calculations is calculated correctly but the method to obtain the probabilities for the others is a heuristic and the results should not be expected to be necessarily accurate.

Also in this case it is possible to approximate the distribution of the maximum and obtain the value of the satisfaction function as explained for the improved Clark

case. The results obtained with the simple Clark method do not need normalisation as they are obtained together and sum to one.

The order in which the variates are processed in either implementation of the Clark approximation might have an effect on the results. This has been considered in the literature and investigations on such matter have concerned the improved Clark method but have only suggested that the results vary little with the order of calculation and no preferable calculation order has been determined (see Daganzo *et al.*, 1977; Lerman and Manski, 1981; Langdon, 1984a,b).

3.5.3 The Approximation of Mendell and Elston

The method of Mendell and Elston solves MVN integrals and, whilst originally developed in biostatistics (Mendell and Elston, 1974, Rice *et al.*, 1979), it has been introduced in transportation by Kamakura (1989) for the calibration of MNP models. Substantially the same approximation has been developed independently by Terada and Takahashi (1988) for bivariate or trivariate Normal integrals in structural reliability applications, and has been employed by Pandey (1998 a, b) to solve larger problems still in the structural reliability context.

Given an n dimensional MVN integral, whose integrand is expressed in the normalised form as in (3.8), and given a truncation of one of the correlated Normal variates of the integrand, the formulae of the Mendell-Elston approximation calculate the conditional variance and correlation of the remaining $n-1$ variates, whose conditional distribution is approximated as standardised MVN, and their conditional limits of integration. Thus, if the truncation of the variate Y_i is considered, the conditional variance σ_{ji}^2 of one of the remaining variates Y_j results:

$$\sigma_{ji}^2 = 1 - \rho_{ji}^2 a_i (a_i + \beta_i) \quad (3.42)$$

where ρ_{ji} is the correlation between Y_j and option Y_i , β_i is the upper limit of integration for Y_i and a_i is given by:

$$a_i = \frac{\varphi(\beta_i)}{\Phi(\beta_i)} \quad (3.43)$$

The conditional correlation $\rho_{mn|i}$ between two variates Y_m and Y_n different from i is given by:

$$\rho_{mn|i} = \frac{\rho_{mn} - \rho_{im}\rho_{in}a_i(a_i + \beta_i)}{\sigma_{m|i}\sigma_{n|i}} \quad (3.44)$$

The distribution defined by these parameters is approximated as MVN and its conditional standardised limits of integration result:

$$\beta_{j|i} = \frac{(\beta_j + a_i\rho_{ji})}{\sigma_{j|i}} \quad (3.45)$$

A MVN integral can be calculated using the approximation formulae recursively until it is reduced to the product of a series of unidimensional Normal integrals. At each stage the marginal integral of the truncation of one variate is calculated and, using the approximation formulae, the conditional distribution of the remaining variates is approximated as MVN and the conditional integration limits obtained. Such approximated conditional MVN integral is treated similarly at the next stage. These operations are repeated until all the variates have been considered and the value of the MVN integral is given by the product of the univariate Normal integrals. This procedure can be summarised as:

$$\begin{aligned} \Phi_n(\boldsymbol{\beta}, \mathbf{0}, \mathbf{P}) &= \\ &= \Phi(Y_n < \beta_n | Y_{n-1} < \beta_{n-1}, \dots, Y_2 < \beta_2, Y_1 < \beta_1) \dots \Phi(Y_2 < \beta_2 | Y_1 < \beta_1) \Phi(Y_1 < \beta_1) \end{aligned} \quad (3.46)$$

The MNP choice probability of each option in the choice set is calculated by obtaining the value of the relevant MVN integral in difference. As they are calculated separately, the choice probabilities might not sum exactly to one, thus they are normalised by dividing each by their sum.

The possible effect on the results of the order in which the variates are introduced in the calculations of the MVN integral has not been considered in the paper of Mendell and Elston (1974). Rice *et al.* (1979), however, noticed the importance of the variate processing order. Working with positively equicorrelated variates and common integration limits they noticed that the accurate results obtained when integration limits giving small marginal integrals were used, could be explained by the reduction of the conditional correlation amongst the remaining variates which, in that case, they bring about. They suggested, as a conjecture, that considering the variates giving the smallest marginal integral first could be of advantage also for more general MVN integrals with different correlation amongst the variates.

Kamakura (1989) did not mention the order suggested by Rice *et al.* (1979) and proposed, instead, that to obtain more precise results, the variates should be considered in the order devised by Langdon (1984a,b) for his approximation: that is the variates with the smallest variance should be considered first. This particular calculation order is such that skewness and kurtosis of the distribution approximated as Normal are as close as possible to those of the Normal distribution.

Terada and Takahashi (1988) and Pandey (1998a,b) did not discuss the calculation order issue.

3.5.4 The Separated Split Approximation of Langdon

Langdon (1984 a, b) put forward expressly for use in transportation the “separated split” method. It relies on calculations for the conditional distributions similar in principle to those of the Mendell-Elston method, but the problem can be analysed directly in terms of utility distributions thus solving directly (3.2). There are also equivalent formulae to work in terms of utility differences. The structure of the calculations is such that all the choice probabilities are calculated at the same time. This makes the separated split calculations more complicated but assures that the probabilities of the different options sum exactly to one.

The basic operation of the method consists of considering the marginal choice between two options in a choice set and splitting the population of choice maker into those who have chosen each option and for whom the other option becomes irrelevant. For each of these two parts of the population, the mean, the variance and the correlation of the utility of all the options in the choice set (except the discarded one) conditional on the choice made are obtained using the formulae devised by Langdon and the distribution of such utilities is approximated as MVN. For instance if two options, say i and j , are considered in a set of J whose utilities are MVN distributed and assuming that i is chosen over j , its marginal probability of choice is:

$$P_{i(j)} = \Phi(\gamma_{ij}) \quad (3.47)$$

where γ_{ij} is:

$$\gamma_{ij} = \frac{V_i - V_j}{\omega_{ij}} \quad (3.48)$$

and

$$\omega_{ij}^2 = \sigma_{ii}^2 + \sigma_{jj}^2 - 2\sigma_{ii}\sigma_{jj}\rho_{ij} \quad (3.49)$$

The conditional mean of the utility of i $V_{i|i}$, results:

$$V_{i|i} = V_i + (\sigma_{ii}^2 - \sigma_{ij}^2)R_1 / \omega_{ij} \quad (3.50)$$

where R_1 , is:

$$R_1 = \frac{\phi(\gamma_{ij})}{\Phi(\gamma_{ij})} \quad (3.51)$$

The conditional variance of option i is:

$$\sigma_{ii|i}^2 = \sigma_{ii}^2 - \frac{(\sigma_{ii}^2 - \sigma_{ij}^2)^2}{\omega_{ij}^2} R_1 (R_1 + \gamma_{ij}) \quad (3.52)$$

Moreover, the conditional mean of any other remaining utility m (except that of j that is discarded as not chosen) results:

$$V_{m|i} = V_m + (\sigma_{im}^2 - \sigma_{jm}^2)R_1 / \omega_{ij} \quad (3.53)$$

and its variance is:

$$\sigma_{mm|i}^2 = \sigma_{mm}^2 - \frac{(\sigma_{im}^2 - \sigma_{jm}^2)^2}{\omega_{ij}^2} R_1 (R_1 + \gamma_{ij}) \quad (3.54)$$

The covariance between i and any of the remaining conditional utilities can be obtained as:

$$\sigma_{im|i}^2 = \sigma_{im}^2 - \frac{(\sigma_{ii}^2 - \sigma_{ij}^2)(\sigma_{im}^2 - \sigma_{jm}^2)}{\omega_{ij}^2} R_1 (R_1 + \gamma_{ij}) \quad (3.55)$$

whilst the covariance between any two other utilities (not including that of the chosen option i) is:

$$\sigma_{mn|i}^2 = \sigma_{mn}^2 - \frac{(\sigma_{im}^2 - \sigma_{jm}^2)(\sigma_{in}^2 - \sigma_{jn}^2)}{\omega_{ij}^2} R_1 (R_1 + \gamma_{ij}) \quad (3.56)$$

The formulae for the case in which j is chosen over i can be obtained by symmetry.

A MNP choice problem is solved by applying recursively to pairs of options' utilities these operations until all possible marginal choices are explored. Each time a choice between a pair of options is considered two separate sets of conditional utilities are calculated and approximated as MVN, one for each of the options in turn considered as chosen. As mentioned above, this can also be seen as a split of the population into two groups according to the choice made. At the next stage each of the groups is then faced with the choice between the option previously chosen and one of the remaining ones. The two groups are thus further split according to the new choices and the calculations proceed repeating choice, group split and utility updating for all the population groups until all the alternatives have been considered. The final choice probability of each option is the sum of the probabilities that each of the population groups resulting in the last stage of the calculations would choose it.

Fig. 3.1 depicts an example of the structure of the separated split calculations in a case with 4 options labelled 1, 2, 3 and 4 (a similar graph can be found for a case with 5 options in Langdon, 1984a). Each square with single border represents the conditional marginal choice between the options indicated (the top square, however, represents only a marginal choice) and the calculation of two conditional sets of utilities and the consequent split of the population in two subgroups. The final probability of choice for each option is obtained by summing the resulting probabilities at the bottom of the calculation “tree”. These are in turn obtained by multiplying all the marginal conditional probabilities on the squares leading to the final square considered. For instance, the total choice probability of the option 1 is given by p_1 which is in turn given by $p_{1(1,2)} \cdot p_{1(1,3)} \cdot p_{1(1,4)}$ (the indices in brackets in the subscripts indicate the relevant conditional marginal choice set whilst the index outside the brackets indicates the chosen option). The total probability of choice of 2 is obtained similarly whilst that for 3 is the sum of $p_{1(1,2)} \cdot p_{3(1,3)} \cdot p_{3(3,4)}$ and $p_{3(1,2)} \cdot p_{3(3,2)} \cdot p_{3(4,3)}$.

It is evident that the structure of the calculations tends to get complicated as the number of options increases.

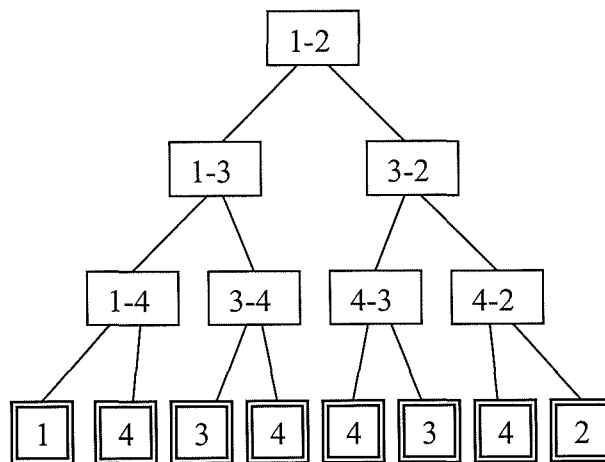


Fig. 3.1 – Example of structure of the calculation with the method of Langdon in a case with 4 options.

One of the variables used in the calculations is the utility of each option conditional on it being chosen (3.50). Continuing the calculations one step beyond what is necessary to obtain the utilities, it is possible to calculate the final utility of each option conditional on it having been chosen over each of the others. As noted by Langdon (1981), summing these conditional utilities weighted by the relevant probability of choice gives the value of the satisfaction of the choice set, the expectation of the maximum of the utilities.

Langdon (1984 a, b) also investigated the effect of the order in which the alternatives are included in the calculations by considering the effect of the characteristics of the utilities on the skewness and the kurtosis of the conditional distribution. As the conditional distribution is approximated as Normal, the best calculation order (in terms of results' accuracy) is such that the skewness and kurtosis of the distribution being approximated as Normal are as close as possible to those of the Normal distribution. Langdon suggested that more precise results are obtained by including at each stage the pair of utilities with the smallest variance of the difference and, in case of a tie and with a minor effect on the results, the pair containing the largest utility.

3.5.5 The Approximations of the FOMN Group

The FOMN (First Order MultiNormal) is a group of three approximations developed in civil engineering for use in structural reliability analysis (see *e.g.* Melchers, 1999). The approximation of this group devised first is known as “crude FOMN” and is due to Hohenbichler and Rackwitz (1983). Hohenbichler refined it proposing the “improved FOMN” approximation (as reported in Tang and Melchers, 1987) and, more recently, Tang and Melchers (1987) built on those previous methods putting forward the “generalised FOMN” method. None of these approximations has been previously considered for use in transportation.

All the FOMN approximations have a similar calculation structure and differ only in a part of the procedure that, however, is important for the quality of the results and for the computational effort involved in obtaining them. A first difference between the methods is that the crude and improved FOMN involve the use of unidimensional Normal integral calculations only whilst the generalised FOMN includes also the use of bivariate Normal integral calculations. The direct calculation of bivariate Normal integrals is possible thanks to computer routines from the literature (*e.g.* Donnelly, 1973; see also Genz, 2001).

All the FOMN methods calculate a MVN integral like (3.8) as the product of univariate Normal conditional marginal integrals, and a bivariate Normal conditional marginal integral in the generalised FOMN case, obtained in stages.

At each stage the marginal integral of the truncation of one of the variates in the MVN distribution is calculated and the conditional distribution of the remaining variates is approximated as a standardised MVN, and the relevant new integration limits obtained. Then the MVN approximating the conditional distribution is considered and the process is repeated. The calculations proceed in this way considering one variate at a time. In the generalised FOMN method, when only a bivariate Normal integral remains, it is solved directly with a suitable algorithm. The J dimensional MVN integral is then calculated as the product of J univariate marginal Normal integrals in the crude and improved FOMN cases or, in the generalised FOMN case, as the product of the $J-2$ univariate Normal integrals and of a bivariate Normal integral.

To approximate the conditional MVN distribution and the new integration limits, the original n dimensional MVN integral $\Phi_n(\boldsymbol{\beta}, \mathbf{0}, \mathbf{P})$ is expressed in terms of n independent standard Normal variates X_j using the Cholesky factorisation of the correlation matrix $\mathbf{P}=\mathbf{C}\mathbf{C}^T$, as specified also in (3.9). The region of integration of (3.8) can therefore be written in vector form as:

$$\mathbf{Y} = \mathbf{C}\mathbf{X} \leq \boldsymbol{\beta} \tag{3.57}$$

or, in expanded form, as:

$$Y_j = \sum_{k=1}^n c_{jk} X_k \leq \beta_j \quad \text{for each line } j \quad (3.58)$$

where c_{jk} is the entry of position jk of \mathbf{C} .

This expression characterises a semispace in the space of the n independent Normal variates delimited by hyperplanes, each one defined by each equation (3.58) when only the equality sign is considered.

Considering the truncation of a variate X_1 , its conditional distribution $X_{1|1}=(X_1 | X_1 \leq \beta_1)$ can be written as:

$$X_{1|1} = \Phi^{-1}(\Phi(\beta_1)\Phi(X_1)) \quad (3.59)$$

The region of integration conditional on the truncation of X_1 can be written substituting (3.59) for X_1 in each of the (3.58) above. This introduces a non-linearity in the (3.58) and the surfaces delimiting the semispace are no longer hyperplanes. The idea of the crude FOMN approximation consists of linearising each of these expressions by replacing it with that of the hyperplane tangent to the surface it describes at the point closest to the origin, which is also the point with highest probability density along that limiting surface.

This point can be found by minimising a program whose objective function is the distance between the origin and the point sought and the point is constrained to be on the surface.

Following Tang and Melchers (1987) the problem of linearising one of this n -dimensional surfaces can be reduced to a problem in two dimensions as the conditioning on X_1 introduces a non-linearity only in one of the terms of (3.58). So if (3.58) for line j is rewritten:

$$c_{j1} \Phi^{-1}(\Phi(\beta_j)\Phi(X_1)) + \sum_{k=2}^j c_{jk} X_k \leq \beta_j \quad (3.60)$$

letting:

$$h_j(X_1) = \Phi^{-1}(\Phi(\beta_j)\Phi(X_1)) \quad (3.61)$$

(3.60) becomes:

$$c_{j1}h(X_1) + \sum_{k=2}^j c_{jk}X_k \leq \beta_j \quad (3.62)$$

the part that remains linear can be compacted in a single variable by letting:

$$\alpha_{j1} = c_{j1} \quad (3.63)$$

$$\alpha_2 = (1 - c_{j1}^2)^{1/2} \quad (3.64)$$

$$V_j = \frac{\left(\sum_{k=2}^j c_{jk}X_k \right)}{\alpha_2} \quad (3.65)$$

and rewriting (3.60) as:

$$g = \alpha_{j1}h(X_1) + \alpha_2V_j \leq \beta_j \quad (3.66)$$

Thus the original problem defined for each line of (3.60) or (3.62) is recast into a two dimensional problem. The closest point to the origin β_j^* can then be calculated solving the program:

$$\text{Min } \beta_j^{*2} = X_1^2 + V_j^2 \quad (3.67)$$

subject to (3.66) re-written with the equality sign.

Keeping on following the method suggested by Tang and Melchers (and coded in the program used in the experiment reported later in the chapter) if (3.66) is solved for V_j and substituted in (3.67), the program becomes:

$$\text{Min } \beta_j^{*2} = X_1^2 + (\beta_j + \alpha_1h(X_1))^2 / \alpha_2^2 \quad (3.68)$$

where:

$$h(X_1) = \Phi^{-1}(\Phi(\beta_1)\Phi(X_1)) \quad (3.69)$$

The solution is then given by the point where the first derivative of (3.68) w.r.t. X_1 is zero, which is when:

$$\alpha_2^2 X_1 + (\beta_j + \alpha_1 h(X_1)) \alpha_1 h'(X_1) = 0 \quad (3.70)$$

where h' is the first derivative of h w.r.t. X_1 . The derivatives of h and of (3.70) can be calculated analytically as detailed in Tang and Melchers (1987) that suggested to find the solution for (3.70) with a Newton-Raphson algorithm.

Once β_j^* is determined, the linearisation of the surface at that point is obtained by calculating the direction cosines of the outward Normal vector to the surface at the point of tangency. These determine the new linear combination coefficient $c_{jk|l}$ that are also the Cholesky factors giving the correlation matrix of the conditional $n-1$ dimensional standardised distribution approximated as MVN.

The direction cosines are obtained as the vector of components:

$$\frac{\partial g_j}{\partial X_k} \quad \forall k \quad (3.71)$$

normalised to its length.

To summarise, the components of the unit vector parallel to (3.71) form the new linear combination coefficient $c_{jk|l}$ for the approximating hyperplane and the β_j^* gives the β_j for the new hyperplane expression.

This procedure is carried out for each line in (3.58). Finally, the new Cholesky coefficients give the covariance matrix of a standardised distribution of one less dimension than the starting one and such distribution is approximated as Normal.

The improved FOMN method (Hohenbichler, 1981, cited in Tang and Melchers, 1987) uses the direction cosines determined with the procedure outlined above but introduces a modification to β_j^* . The β_j^* of the crude FOMN method approximates the probability content behind the non-linear curve (when the problem is re-written in two dimensions as above) resulting from the conditioning with that contained

behind the straight line tangent to it at the point closer to the origin. The correction introduced by the improved FOMN considers the asymptotes to the actual non-linear limit of integration and corrects the β_j^* so that it represents the union of the probability contents behind the approximating straight line and behind the asymptotes. The direction cosines are not changed so the approximating straight line is simply shifted parallel to itself to adapt its distance from the origin to the new β_j^* .

The modified β_j^* is obtained by calculating the union of the probability content behind the approximating line and the two asymptotes. If the approximating straight line, the right hand side asymptote and the left hand side asymptote are respectively written as:

$$g_{j,*} = 0 \qquad g_{j,R} = 0 \qquad g_{j,L} = 0 \qquad (3.72 \text{ a, b, c})$$

the union of the probability content of the space behind them is:

$$P_j = P(g_{j,*} \leq 0) + P(g_{j,*} \geq 0 \cap g_{j,L} \leq 0) + P(g_{j,*} \geq 0 \cap g_{j,R} \leq 0) \qquad (3.73)$$

The first term in this expression is simply the univariate Normal integral with integration limit β_j^* and the others can be obtained as bivariate Normal integrals, reduced to univariate ones in the original method, according to the formulae detailed in Tang and Melchers (1987). The improved β_j^* is then simply obtained as the inverse Normal cumulative distribution of P_j in (3.73):

$$\beta_{j|1,impr} = \Phi^{-1}(P_j) \qquad (3.74)$$

Tang and Melchers (1987) suggested a different modification of the crude FOMN algorithm to improve the determination of the conditional integration limits. Their generalised FOMN method entails calculating exactly the conditional probability:

$$P_{j,EX} = P\left(\sum_{k=1}^n c_{jk} X_k \leq \beta_j | X_1 < \beta_1\right) = P(Y_j \leq \beta_j | X_1 < \beta_1) \qquad (3.75)$$

thus working directly with the original variables, their integration limits β_j and β_1 and their correlation ρ_{1j} , to write (3.75) as:

$$P_{j,EX} = \frac{P[(Y_j \leq \beta_j) \cap (Y_1 \leq \beta_1)]}{P(Y_1 \leq \beta_1)} = \frac{\Phi_2(\beta_1; \beta_j; \rho_{1j})}{\Phi(\beta_1)} \quad (3.76)$$

where the numerator is calculated using a routine for the bivariate Normal integral. The new limit of integration for each row j is then obtained as:

$$\beta_{j|1,GEN} = \Phi^{-1}(P_{j,EX}) \quad (3.77)$$

With any of the FOMN approximations, the probit choice function for each alternative in a choice set is calculated by obtaining the integral (3.8) for each option and normalising the resulting probabilities to their sum if they do not sum exactly to one.

The literature on the FOMN methods (Hohenbichler and Rackwitz, 1983; Tang and Melchers, 1987; Melchers, 1999) does not seem to discuss the possible effects related to the calculation order although Tang and Melchers pointed out that the approximations are less precise when the surface approximated with an hyperplane is closer to the origin, because of the high probability density in such a region.

3.5.6 The Taylor Approximation

The use of the first, second or higher order Taylor series approximation for evaluating MVN integrals has been suggested by Cox and Wermuth (1991) who considered bivariate and trivariate Normal integrals only, and by Olson and Weissfeld (1991) that independently but focusing on biostatistics applications as well, considered mainly bivariate and trivariate integrals but investigated the method also for the evaluation MVN integrals of up to 20 dimensions. In fact, the basic method they proposed can be generalised to any number of variates and, as noted by Olson and Weissfeld, is also applicable to integrals of other functional forms.

The method is based on reducing a MVN integral to a univariate Normal integral and a sequence of factors that are Taylor series approximations of expectations of marginal conditional probabilities. These reduce to univariate Normal integrals in the first order series case.

Starting from a MVN integral as in (3.8), the marginal integral of one of the variates is obtained, then the contribution of the integral of a second variate is accounted as the Taylor series approximation of the expectation of its marginal integral conditional on the truncation of the first variate. Considering one more variate at a time and obtaining similarly their contribution as the expectation of the marginal integral of each of them conditional on the truncation of the variates previously included, the MVN integral is reduced to a series of factors.

The formulae used can be explained by considering the first two variates in a MVN distribution. The marginal integral of the first variate will be:

$$P(Y_1 < \beta_1) = \Phi(\beta_1) \quad (3.78)$$

The contribution of the second variate is obtained considering that:

$$P(Y_2 < \beta_2 | Y_1 < \beta_1) = E\left[P\left((Y_2 | Y_1 = y) < \beta_2 | Y_1 < \beta_1\right)\right] \quad (3.79)$$

To elaborate on this expression it should be recalled that the variate Y_2 conditional on $Y_1=y$ will be Normally distributed with mean and variance respectively:

$$\mu_{Y_2|Y_1=y} = \rho_{12}y \quad (3.80)$$

$$\sigma^2_{Y_2|Y_1=y} = 1 - \rho^2_{12} \quad (3.81)$$

The mean and variance of any third variate can be obtained similarly, and their correlation will be:

$$\rho_{ij|1} = \frac{\rho_{ij} - \rho_{i1}\rho_{j1}}{\sqrt{1 - \rho_{i1}^2}\sqrt{1 - \rho_{j1}^2}} \quad (3.82)$$

(these are general results for multivariate Normal distributions, see *e.g.* Johnson and Kotz, 1972).

Moreover, the distribution of Y_1 conditional on its truncation has mean and variance that are, respectively:

$$\mu_{Y_1|Y_1 < \beta_1} = -\frac{\phi(\beta_1)}{\Phi(\beta_1)} \quad (3.83)$$

$$\sigma^2_{Y_1|Y_1 < \beta_1} = 1 - \beta_1 \mu_{Y_1|Y_1 < \beta_1} - \left(\mu_{Y_1|Y_1 < \beta_1}\right)^2 \quad (3.84)$$

Finally, the Taylor series approximation to the expectation of a function can be written (in this case writing the terms up to the second order) as:

$$E[g(X)] = g(\mu) + \frac{1}{2} g''(\mu)E(X - \mu)^2 + o(X - \mu)^2 \quad (3.85)$$

where μ is the mean of X .

Returning to (3.79), in the present case $g(X)$ is the conditional integral of the second variate that can be written as:

$$\left[\Phi\left(\frac{\beta_2 - \rho_{12}Y_1}{\sqrt{1 - \rho_{12}^2}}\right) \middle| Y_1 < \beta_1 \right] \quad (3.86)$$

Thus, using the first order Taylor approximation, (3.79) can be rewritten:

$$\begin{aligned} P(Y_2 < \beta_2 | Y_1 < \beta_1) &= E\left[P((Y_2 | Y_1 = y) < \beta_2 | Y_1 < \beta_1)\right] = \\ &E\left[\Phi\left(\frac{\beta_2 - \rho_{12}Y_1}{\sqrt{1 - \rho_{12}^2}}\right) \middle| Y_1 < \beta_1\right] = \Phi\left(\frac{\beta_2 - \rho_{12}\mu_{Y_1|Y_1 < \beta_1}}{\sqrt{1 - \rho_{12}^2}}\right) \end{aligned} \quad (3.87)$$

that can also be seen as the Normal integral for the distribution conditional on the first variate being equal to its mean when its truncation is considered.

Considering one more variate at a time in this fashion, a MVN integral is reduced to the product of a sequence of univariate Normal integrals.

This method appears similar, numerically, to that proposed by Mendell and Elston, but, without considering the different rationale, they differ in the way the covariance matrix of the distribution of the variate conditional on the truncation of one of them is obtained.

The Taylor series of the contribution of each variate can also be expanded beyond the first term to improve the accuracy of the method as suggested by Olson and Weissfeld (1991). This can be carried out at little additional computational expense as few added quantities are necessary for writing additional terms.

Here it is considered also the approximation derived by using the second term expansion, already used by Cox and Wermuth for evaluating bivariate Normal integrals. The expression (3.79) can be re-written with its second term Taylor series as:

$$\begin{aligned}
 P(Y_2 < \beta_2 | Y_1 < \beta_1) &= E\left[P((Y_2 | Y_1 = y) < \beta_2 | Y_1 < \beta_1)\right] = E\left[\Phi\left(\frac{\beta_2 - \rho_{12}Y_1}{\sqrt{1 - \rho_{12}^2}}\right) \Big| Y_1 < \beta_1\right] = \\
 &= \Phi\left(\frac{\beta_2 - \rho_{12}\mu_{Y_1|Y_1 < \beta_1}}{\sqrt{1 - \rho_{12}^2}}\right) + \\
 &\quad - \frac{1}{2}\sigma_{Y_1|Y_1 < \beta_1}^2 \left[\frac{\rho_{12}^2}{1 - \rho_{12}^2} \left(\frac{\beta_2 - \rho_{12}\mu_{Y_1|Y_1 < \beta_1}}{\sqrt{1 - \rho_{12}^2}}\right) \varphi\left(\frac{\beta_2 - \rho_{12}\mu_{Y_1|Y_1 < \beta_1}}{\sqrt{1 - \rho_{12}^2}}\right) \right]
 \end{aligned} \tag{3.88}$$

Thus in the second order case the contribution of each variate other than the first is obtained with a expression similar to (3.88).

The MNP choice function of each option in a choice set can be obtained by calculating the relevant MVN integral in difference (3.8) for each option with the approximation extended to the relevant order and normalising the results to their sum, as they are not calculated simultaneously and therefore their sum could be not exactly one.

Both Olson and Weissfeld (1991) and Cox and Wermuth (1991) recognised the importance of the sequence in which the variates are considered on the accuracy of the final results. Olson and Weissfeld (1991) suggested a simple rule for determining the calculation sequence that can be used with MVN integrals of any dimension (and is applicable with any Taylor series order): they suggest to use at each stage as conditioning variate the one giving the smaller marginal integral, that is the one with the smaller integration limit in the case considered here. The rationale for this order is that the smaller the conditioning integral the smaller will be the second and higher terms of the Taylor series expansion. In other words, assuming that the Taylor series expansion converges, it does so more quickly than with a different calculation order, and disregarding higher order terms has a lower impact on the final result.

Cox and Wermuth (1991) suggested the same calculation order for trivariate Normal integrals and a more elaborate rule for bivariate ones. The rationale suggested by Cox and Wermuth (1991) for their order rule is that it introduces lower non-linearity in the conditional terms than other possible orders, which coincides with the rationale of Olson and Weissfeld (1991).

3.5.7 The Solow-Joe Approximation

The method of Solow (1990), as extended by Joe (1995), is based on a formula to approximate the conditional integral of a variate in a MVN distribution given a truncation of all the others. Solow proposed the method for one-sided MVN integrals with equal integration limits. Joe extended it to two sided integrals, different integration limits and considered the calculation order issue. No further application of this method resulted from a literature survey.

As described by Joe (1995), the approximation is based on a formula analogous to the exact formula to obtain the expected value of a Normal variate given the truncation of a correlated one. In the bivariate case the latter formula is:

$$E(Y_2|Y_1 = y_1) = \mu_2 + \rho_{21}(y_1 - \mu_1) \quad (3.89)$$

where μ_j is the mean of the variate j and the other symbols are as for (3.8). This is the formula used in the approximation of Taylor. The (3.89) can be generalised to the multivariate case, that is to the case when the expected value of one variate conditional on a number of others having a fixed value with the following:

$$E(\mathbf{Y}_2|\mathbf{Y}_1 = \mathbf{y}_1) = \boldsymbol{\mu}_2 + \mathbf{P}_{21}\mathbf{P}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1) \quad (3.90)$$

where \mathbf{P}_{11} \mathbf{P}_{21} are partitions of the correlation matrix \mathbf{P} delimited by the conditioning and conditioned vectors

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \quad (3.91)$$

The formula proposed by Solow and by Joe, instead of the variables taking a fixed value, considers the event of the truncation of each conditioning variate and each pair of conditioning variates. Thus it uses the marginal probability of each conditioning variate and the bidimensional marginal probabilities of each pair of conditioning variates. Considering in the notation the unilateral truncation, these can be written respectively:

$$E(I_j) = \Pr(Y_j < \beta_j) = \Phi(\beta_j) \quad (3.92)$$

$$E(I_j, I_m) = \Pr(Y_j < \beta_j; Y_m < \beta_m) = \Phi_2(\beta_j; \beta_m; \rho_{jm}) \quad (3.93)$$

According to this approximation method, the conditional marginal integral of the k th variate in a k -dimensional MVN distribution given the truncation of all the others can be approximated as:

$$\Pr(X_k < \beta_k | X_1 < \beta_1, \dots, X_{k-1} < \beta_{k-1}) = E(I_k) + \Omega_{21} \Omega_{11}^{-1} (1 - E(I_1), \dots, 1 - E(I_{k-1}))^T \quad (3.94)$$

where Ω_{21} is the vector of the $k-1$ covariances between the conditional and the conditioning events:

$$\Omega_{21} = \begin{bmatrix} \text{cov}(I_k, I_1) \\ \text{cov}(I_k, I_2) \\ \dots \\ \text{cov}(I_k, I_{k-1}) \end{bmatrix} = \begin{bmatrix} E(I_k, I_1) - E(I_k)E(I_1) \\ E(I_k, I_2) - E(I_k)E(I_2) \\ \dots \\ E(I_k, I_{k-1}) - E(I_k)E(I_{k-1}) \end{bmatrix} \quad (3.95)$$

and Ω_{11} is a $k-1 \times k-1$ matrix of covariances between the conditioning events:

$$\Omega_{11} = \begin{bmatrix} \text{cov}(I_1, I_1) & \text{cov}(I_1, I_2) & \dots & \text{cov}(I_1, I_{k-1}) \\ \text{cov}(I_2, I_1) & \text{cov}(I_2, I_2) & \dots & \text{cov}(I_2, I_{k-1}) \\ \dots & \dots & \dots & \dots \\ \text{cov}(I_{k-1}, I_1) & \text{cov}(I_{k-1}, I_2) & \dots & \text{cov}(I_{k-1}, I_{k-1}) \end{bmatrix} = \quad (3.96)$$

$$= \begin{bmatrix} 1 & E(I_1, I_2) - E(I_1)E(I_2) & \dots & E(I_1, I_{k-1}) - E(I_1)E(I_{k-1}) \\ E(I_2, I_1) - E(I_2)E(I_1) & 1 & \dots & E(I_2, I_{k-1}) - E(I_2)E(I_{k-1}) \\ \dots & \dots & \dots & \dots \\ E(I_{k-1}, I_1) - E(I_{k-1})E(I_1) & E(I_{k-1}, I_2) - E(I_{k-1})E(I_2) & \dots & 1 \end{bmatrix}$$

Thus the integral of an n -dimensional MVN distribution can be reduced to the product of a bivariate marginal integral, calculated directly with a bivariate Normal integration routine, and of the remaining $n-2$ conditional univariate marginal integrals, each of them obtained with the approximation formula (3.94). First the bivariate marginal integral is calculated, then the (3.94) is applied to compute the marginal integral of a third variate conditional on the truncation of the first two. Next, the (3.94) is applied to the calculation of the marginal integral of a fourth variate conditional on the first three, and so on, until all the variates are accounted for.

With the Solow-Joe approximation, the value of the MNP choice function for each option in a choice set is obtained by calculating the relevant MVN integral of the

utility distribution in difference, and normalising the final results to their sum if this is not exactly one.

Joe (1995) remarked on the existence of the calculation order issue: the result of the approximation depends on the order in which the variates are considered. He also claimed that the mixed quality of the results previously obtained by Solow was due to the lack of consideration of this issue and suggested to circumvent the problem by obtaining the final value of a MVN integral by averaging the results obtained considering a number of the possible permutations of calculation orders. In particular for a MVN integral of up to 6 dimension he suggested averaging over all the possible permutations and averaging the results from a sample of them (100-10000) for higher dimensions. The size of such a sample should be determined by the user of the approximation based on information on the standard deviation of the mean of the probability resulting from different sample sizes.

Joe (1995) pointed out that his method does not use approximations to the conditional distributions and therefore can be used not only in the MVN case but also with other distributions whose multivariate integral is difficult to evaluate but for which univariate and bivariate integrals can be readily calculated.

3.5.8 The Extended Approximation of Joe

Joe (1995) extended the idea on which the Solow-Joe approximation, illustrated in the previous section, is based to calculate MVN integrals using the values of one-dimensional to four-dimensional marginal integrals of the conditioning variates to write the approximating expression of a marginal conditional integral in a MVN distribution. Such an extended method thus solves by numerical integration the MVN integrals of dimension up to four and employs both numerical integration and analytical approximation for integrals of larger dimension. No further use of such approximation resulted from a literature survey.

This approximation, called by Joe (1995) second order improved approximation, is based on a formula similar to (3.94) but including as events the truncation of each conditioning variate and of groups of 2, 3 and 4 of them. Such a formula allows us to obtain the conditional marginal integral of a variate in a MVN distribution given the truncation of at least four others, which includes marginal MVN integrals of dimension up to 4 calculated directly with a suitable routine or with a numerical integration program.

The integral of an n -dimensional MVN distribution (with n greater than four) is approximated by reducing it to the product of a quadrivariate marginal integral, obtained directly with a numerical integration routine (Joe included in his program the numerical integration method of Schervish (1984) since he found it quicker than the method of Genz for quadrivariate integrals), and of the remaining $n-4$ conditional univariate marginal integrals obtained with the approximation's formula. First the approximation is applied to compute the marginal integral of a fifth variate conditional on the truncation of the first four. Then the approximation is applied to obtain the marginal integral of a sixth variate conditional on the first five, and so on, until all the conditional marginal integrals of all the variates are accounted for. Thus rather than an approximation method similar to those previously illustrated, the extended approximation of Joe can be seen as a hybrid between numerical integration and analytical approximation.

As in other methods approximating the MVN integral rather than directly the MNP, the value of the MNP choice function for each option in a choice set is obtained by calculating the relevant integral of the utility distribution in difference (3.8), and normalising the final results to their sum if they do not sum to one.

Whilst the order of calculation for the quadrivariate and trivariate integrals is dealt with inside the numerical integration routine the order in which the variates are selected to calculate their conditional marginal integral can affect the final result.

Also in this case Joe suggested to circumvent the problem by averaging the results obtained by considering a number of the possible permutations of calculation orders. In particular for MVN integral with up to 6 variates he suggested averaging over all the possible permutations and averaging the results from a sample of them for higher dimensions, whose size should be determined by the user of the approximation based on information on the standard deviation of the mean of the probability resulting from different sample sizes.

The remark of Joe (1995) reported in the previous section, about the applicability of that method to other multivariate distributions with univariate and bivariate integrals being easy to calculate and higher order integrals difficult to obtain is valid also for this approximation. However, in this case the integrals of dimension up to four need to be easy to obtain to make viable the use of the method.

Before closing the illustration of the approximation methods of Joe, it should be mentioned that Joe (1995) proposed a third approximation for calculating MVN integrals based on the moment generating function of the truncated MVN distribution. This approximation, however, is not described here as it has not been included in the following series of investigations.

3.6 Investigations on Accuracy and Computational Cost of Approximations for Solving the MNP Choice Function

3.6.1 Introduction

This section presents the results of a series of numerical tests carried out to assess the accuracy and the computational cost of some of the probit approximation methods introduced in the previous part of this chapter.

For practical applications, approximations are required to be as precise as possible but the computational time they take to solve the MNP choice function is a rather

important element in their applicability: choice calculations are carried out a large number of times in algorithms for traffic assignment and for model calibration. Therefore, the overall calculation time, and, consequently, the practical applicability, of such algorithms depends strongly on the calculation time of the MNP method employed. A trade-off between accuracy and computational time might be required and the investigations carried out here aim at assessing the suitability of the methods for transportation applications by weighting those two elements.

The tests have been carried out focussing on the application of MNP for traffic assignment and therefore representing test choice situations as choices of paths through networks. The probabilities calculated are those of each of the path's cost being minimum amongst all the costs of the paths between the same origin-destination pair. This is a problem equivalent to maximising options' utilities: the costs, in fact, are negative utilities, or disutilities.

The methodology and the test networks used in the investigations are described in the next two sections and graphical and numerical results are reported in the following ones.

3.6.2 Methodology

The accuracy of the approximation methods has been tested by comparing the choice probabilities they give in a large series of choice situations represented as choices of paths through a network with the probabilities obtained with the numerical integration method of Genz (1992, 1993). The numerical integration method of Genz has been chosen as reference method because of the published results guaranteeing its good accuracy (see Genz, 1992 and 1993) and because of the possibility of setting the required accuracy of the results at the outset of the calculations. To obtain the reference probabilities used in the tests, the calculations were carried out requiring an absolute precision for each path probability of 0.00005.

The data on the computational costs have been obtained from the calculation times on the same series of test cases working on a Pentium II 350MHz desktop computer.

Although the results are relative to the computer and programs used, they should give a fair comparison of relative calculation efforts involved.

To carry out the tests, FORTRAN routines for each approximation method have been included in a path-based assignment program. The routines for most methods have been specially written during the research work. The only exceptions were the routine for the numerical approximation methods of Genz (1992) and the routines for the Solow-Joe (Joe, 1995) and the extended approximation of Joe (1995).

The code for the numerical integration method of Genz has been adapted (only to the extent needed to include it in the programs used for this project) from that publicly available on Alan Genz's web page: <http://www.math.wsu.edu/math/faculty/genz/homepage>. Such code can solve MVN integrals with up to 20 dimensions. The subregion adaptive numerical integration method supplied with the MVN integration program is an intermediate program between ADAPT of Genz and Malik (1980) and DCUHRE of Berntsen *et al.* (1991) but it is closer to the latter (Genz, 2000).

The codes for the Solow-Joe and the extended Joe approximation have been obtained from those publicly available at <ftp://stat.ubc.ca/pub/hjoe/mvnapp>. The changes made to the original codes were mainly to include the programs in the path-based assignment program and for allowing different calculation orders.

For uniformity, the approximations coded use the routine for the calculation of the univariate Normal probability distributed with the Numerical integration program of Genz which provides results accurate to 10^{-15} and is based on an algorithm by Hart *et al.* (1968) (Genz, 2001). A limited series of tests carried out with less precise routines (*e.g.* with precision 10^{-7}) showed no substantial influence on the quality of the results. Bivariate Normal integrals calculated directly within the analytical approximation routines are obtained with the algorithms of Donnelly (1973).

The only cases in which a different univariate Normal routine is used are the Solow-Joe and the extended Joe approximations, in which the routine for univariate Normal integration due to Hill (1973) and, for the extended Joe approximation only, the

numerical integration method (Mulnor of Schervish, 1984), included in the original programs have been kept.

Two series of numerical tests have been carried out, using choice situations described by a large number of small artificial networks. Details on the networks used and on how they have been obtained are given in the following section.

For each choice situation the choice probabilities have been calculated and compared with the reference ones numerically and graphically. Two sorts of graphs have been used: one depicting the percentage errors against the actual choice probabilities and the other plotting the cumulative relative frequency of the percentage errors for values between -20% and $+20\%$ grouped in bins of width 1% . Numerical results include absolute and percentage errors and summary statistics (means and variances of the errors) both over the whole range of probabilities for each experiment and binned.

3.6.3 Test Networks

Two series of accuracy tests have been performed on small artificial networks whose topology has been designed so that 3, 6, 9, 12 and 15 paths could be enumerated.

The number of paths through the test networks has been chosen to be representative of the most likely dimensions of the choice sets in traffic assignment and in MNP calibration. Practical experiences suggest that the number of actually used paths between each origin-destination (OD) pair is generally limited. Cascetta *et al.* (1996) in a model of interurban assignment used choice sets with a maximum of 8 paths. Again Cascetta *et al.* (1997), calibrating a path enumeration model in an urban context and therefore a choice set building model for traffic assignment, reported obtaining the best results with their assignment program with 6-8 paths per OD and underlined that increasing the number of paths did not necessarily imply a an improvement in the assignment results. In a further urban assignment model Cantarella *et al.* (1999) reported having used only 5 paths per OD pair.

In calibration applications the number of options is generally limited as well, with often less than 10 option being considered. An example of a practical case with a

relatively large number of options is that in Bolduc (1999) who considered MNP with 9 options where, however, not all the options are available to all users and therefore the actual dimension of the choice problem considered for each sampled user is smaller than the overall choice set.

The topology of the networks used in the experiments has been designed to consider sets of paths with different degrees of overlapping. For instance, the networks for 9 paths include networks with 9 independent paths, with 6 independent and 3 partially overlapping paths, with 3 independent and 6 partially correlated paths and with 9 partially overlapping paths. A summary of the different degrees of overlapping considered is provided in table 3.1 whilst fig. 3.2 reports the sketches of the network topologies used in the cases of 6 paths. It should be noted that networks with independent paths and therefore MVN utility distributions with no correlation have however a MVN utility distribution in difference with correlated variates.

Each of the paths enumerated on a network is made up by the same number of links that is equal to the number of paths enumerated (for instance, on the networks designed to enumerate 9 paths, all the paths are made up of 9 links).

No. of paths	Topologies employed				
3	3 i.p.	3 p.c.p.			
6	6 i.p.	3 i.p.+ 3 p.c.p.	6 p.c.p.		
9	9 i.p.	6 i.p.+ 3 p.c.p.	3 i.p.+ 6 p.c.p.	9 p.c.p.	
12	12 i.p.	6 i.p.+ 6 p.c.p.	3 i.p.+ 9 p.c.p.	12 p.c.p.	
15	15 i.p.	9 i.p.+ 6 p.c.p.	6 i.p.+ 9 p.c.p.	3i.p.+ 12p.c.p.	15 p.c.p.

Table 3.1- Summary of the network topologies employed (legend: i.p.: independent paths; p.c.p.: partially correlated paths).

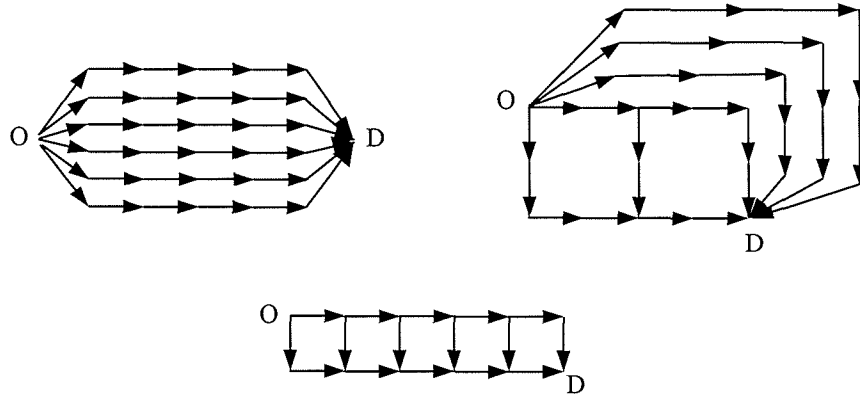


Fig. 3.2 – Sketches of the topologies of the networks with 6 paths (not to scale).

The networks used in the two series of experiments differ for the link and path costs used.

The networks for the first series of experiments have links whose mean costs are drawn from a uniform distribution with extremes 5 and 10. The variances of the link costs are obtained by multiplying the mean cost by a fixed coefficient of 0.5. The networks in this series of experiments can be thought as representing the situation of stochastic network loading on free flow costs at the beginning of the calculation of a stochastic user equilibrium assignment.

The networks of the second series of experiments are obtained from those for the first series. They retain the variances of the link costs used in the first series of experiments whilst each mean link cost is obtained by multiplying that used for the first series of networks by a coefficient drawn from an uniform distribution with extremes 1 and 2. These networks are intended to replicate the conditions encountered during a Stochastic User Equilibrium (SUE) calculation away from the start.

From each base network topology 30 actual networks have been obtained for the first series of experiments and 30 for the second series. The total number of networks for each series of experiments and each number of paths is reported in the following table 3.2.

	First series	Second series
3 paths	60	60
6 paths	90	90
9 paths	120	120
12 paths	120	120
15 paths	150	150
<i>Total</i>	<i>540</i>	<i>540</i>

Table 3.2 – Number of test networks for each number of paths and for each series of experiments.

Figures 3.3 to 3.7 depict the cumulative relative frequencies of the reference choice probabilities obtained by numerical integration for the two series of artificial networks. Except in the 3 option case, most choice probabilities are in the range 0-30%, although the probabilities for the second series of networks are spread over a larger range than those for the first series. This is due to the higher variability of the deterministic costs. Also, as the number of options increases the choice probabilities tend to have smaller values for both network series.

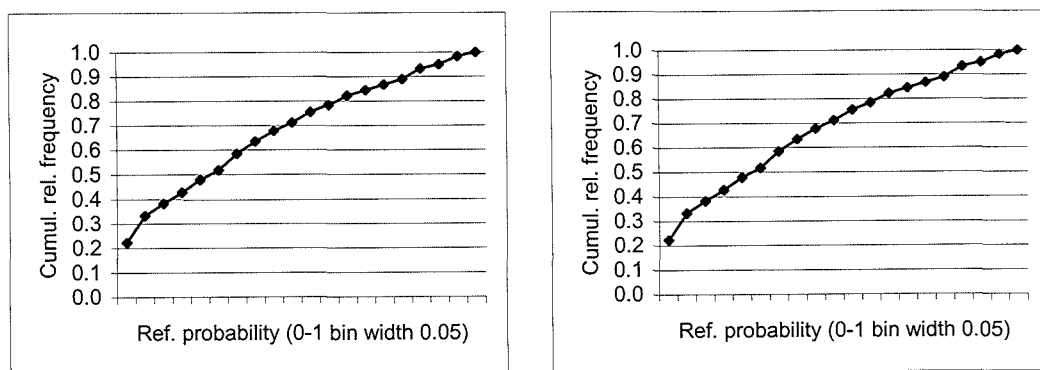


Fig. 3.3 - Cumulative relative frequencies of the reference choice probabilities for the 3 path networks of the first series (left) and for those of the second series (right).

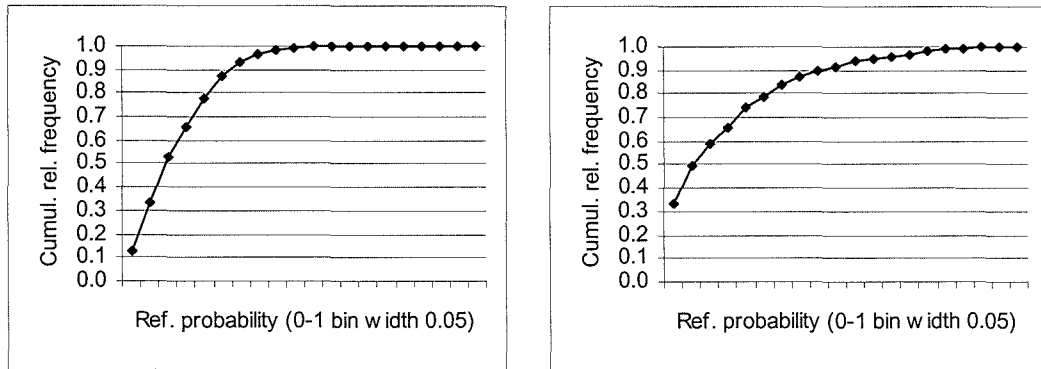


Fig. 3.4 – Cumulative relative frequencies of the reference choice probabilities for the 6 path networks of the first series (left) and for those of the second series (right).

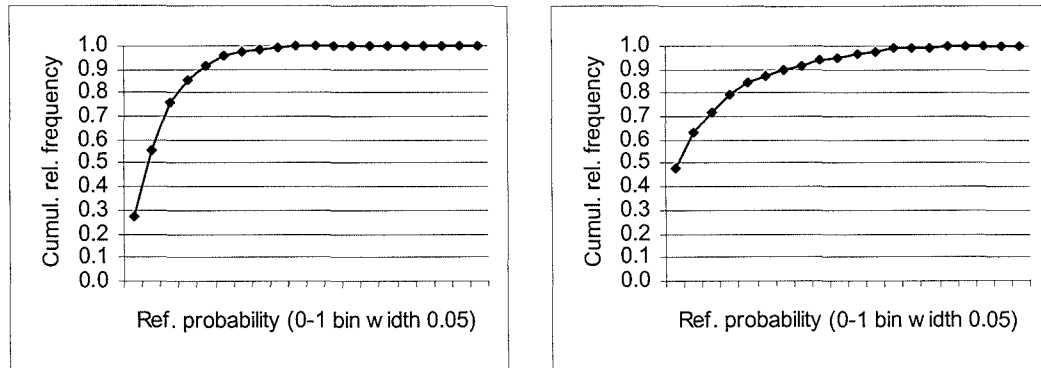


Fig. 3.5 – Cumulative relative frequencies of the reference choice probabilities for the 9 path networks of the first series (left) and for those of the second series (right).

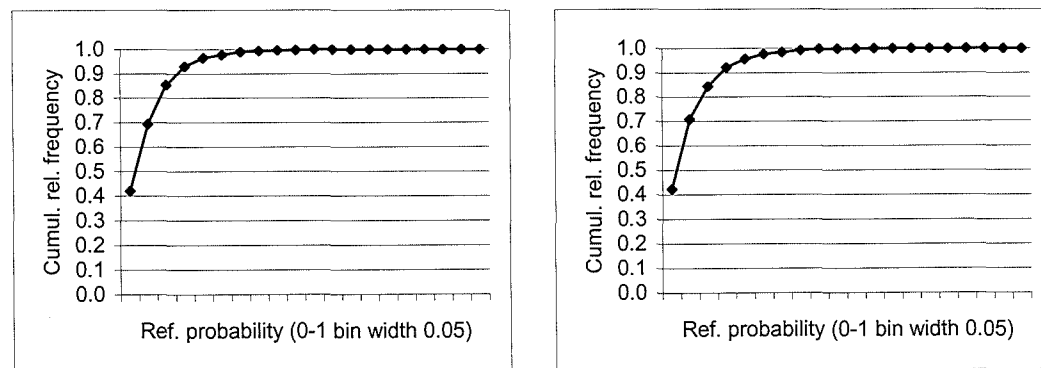


Fig. 3.6 – Cumulative relative frequencies of the reference choice probabilities for the 12 path networks of the first series (left) and for those of the second series (right).

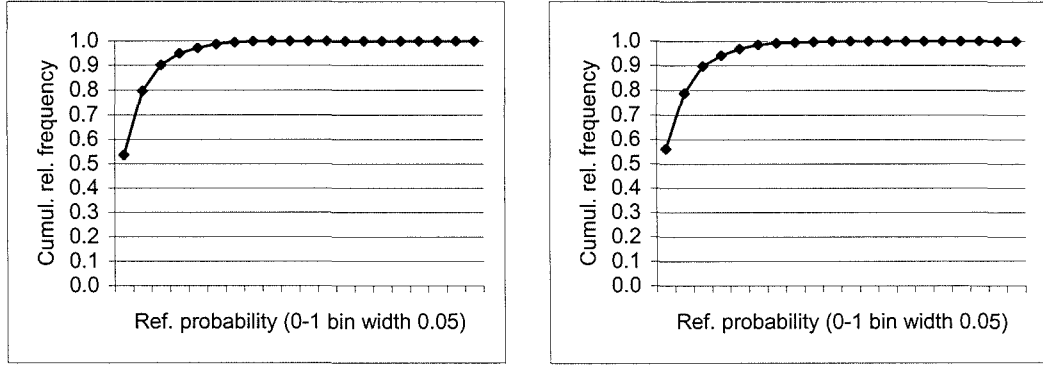


Fig. 3.7 – Cumulative relative frequencies of the reference choice probabilities for the 15 path networks of the first series (left) and for those of the second series (right).

The following sections report aggregate results together or separately for the first and the second series of networks. However, also the numerical results for 3 particular cases, respectively with 3, 6 and 9 choice options, taken from the second series of experiments are reported to give some disaggregate information on the approximations' quality.

The first case, with 3 partially correlated options, is defined by the vector \mathbf{C}_1 of costs and the covariance matrix Σ_1 , that give the vector \mathbf{P}_1 of numerical integration choice probabilities:

$$\mathbf{C}_1 = \begin{bmatrix} 15.86 \\ 17.97 \\ 14.41 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 5.35 & 1.97 & 0 \\ 1.97 & 5.49 & 1.52 \\ 0 & 1.52 & 5.12 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 0.3117 \\ 0.0365 \\ 0.6519 \end{bmatrix} \quad (3.97)$$

The second example considers the choice amongst 6 options characterised by the vector \mathbf{C}_2 of costs and covariance matrix Σ_2 , where 3 options are partially correlated and 3 are independent. The choice probabilities resulting from numerical integration are reported in \mathbf{P}_2 .

$$\mathbf{C}_2 = \begin{bmatrix} 35.11 \\ 32.53 \\ 30.80 \\ 38.70 \\ 34.29 \\ 36.91 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 11.57 & 4.02 & 0 & 0 & 0 & 0 \\ 4.02 & 11.19 & 4.48 & 0 & 0 & 0 \\ 0 & 4.48 & 10.67 & 0 & 0 & 0 \\ 0 & 0 & 0 & 11.49 & 0 & 0 \\ 0 & 0 & 0 & 0 & 11.43 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11.54 \end{bmatrix} \quad (3.98a)$$

$$\mathbf{P}_2 = \begin{bmatrix} 0.0824 \\ 0.2058 \\ 0.5127 \\ 0.0134 \\ 0.1461 \\ 0.0396 \end{bmatrix} \quad (3.98b)$$

A third example considers 9 choice options, all of which are correlated with some of the others, with costs \mathbf{C}_3 and covariance matrix Σ_3 , that result in the choice probabilities in \mathbf{P}_3 , obtained with numerical integration.

$$\mathbf{C}_3 = \begin{bmatrix} 50.90 \\ 50.53 \\ 47.18 \\ 47.98 \\ 48.68 \\ 49.06 \\ 51.32 \\ 49.63 \\ 50.04 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 16.23 & 11.96 & 10.57 & 9.19 & 7.41 & 5.00 & 3.74 & 1.74 & 0 \\ 11.96 & 15.89 & 12.36 & 10.97 & 9.20 & 6.79 & 5.52 & 3.52 & 1.79 \\ 10.57 & 12.36 & 15.19 & 12.39 & 10.61 & 8.21 & 6.94 & 4.94 & 3.20 \\ 9.19 & 10.97 & 12.39 & 16.08 & 12.47 & 10.06 & 8.80 & 6.80 & 5.06 \\ 7.41 & 9.20 & 10.61 & 12.47 & 15.69 & 11.70 & 10.42 & 8.43 & 6.69 \\ 5.00 & 6.79 & 8.21 & 10.06 & 11.70 & 16.02 & 12.82 & 10.82 & 9.09 \\ 3.74 & 5.52 & 6.94 & 8.80 & 10.42 & 12.82 & 16.89 & 13.17 & 11.43 \\ 1.74 & 3.52 & 4.94 & 6.80 & 8.43 & 10.82 & 13.17 & 16.44 & 13.25 \\ 0 & 1.79 & 3.20 & 5.06 & 6.69 & 9.09 & 11.43 & 13.25 & 16.67 \end{bmatrix}$$

$$\mathbf{P}_3 = \begin{bmatrix} 0.0544 \\ 0.0285 \\ 0.3246 \\ 0.1582 \\ 0.0950 \\ 0.1029 \\ 0.0146 \\ 0.1032 \\ 0.1186 \end{bmatrix} \quad (3.99)$$

The examples have been taken from the second series of networks as they give a good spread of reference choice probabilities, as (3.97), (3.98) and (3.99) show, but do not include reference probabilities smaller than 10^{-3} , that have been generally excluded from the analysis of the results.

3.6.4 Results on Approximation Accuracy

3.6.4.1 Accuracy of the Simple Clark Approximations

Before reporting the results on the simple Clark approximation it should be recalled that it is a heuristic method, as explained in section 3.5.2. The results presented have been obtained without following any particular order for including the options in the calculations as no such order is suggested in the literature: the variates are simply processed in the order they have been coded in the computer program input file.

Observing the results relative to the examples reported in tables 3.3, 3.4 and 3.5 it can be seen how using this heuristic typically leads to large percentage errors that can affect probabilities of different magnitude but are particularly relevant to options with actual small choice probabilities (see *e.g.* the second option of table 3.3 or the seventh option of table 3.5). Examples of larger actual choice probabilities affected by large percentage errors are *e.g.* those of the third and of the eighth option in table 3.5.

The moderate percentage errors for the last options in the tables can be explained considering that such options are also the last ones entered in the calculations, and consistent with the considerations in section 3.5.2, the ones for which the probability is calculated using the Clark approximation in an exact way and for which a better accuracy should be expected a priori.

Option	Num.Int.	CS	Δ CS%
1	0.3117	0.2789	-10.50%
2	0.0365	0.0746	+104.58%
3	0.6519	0.6464	-0.83%

Tab 3.3 – Comparison of Simple Clark (CS) and reference results for the three alternative example. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	CS	Δ CS%
1	0.0824	0.0747	-9.40%
2	0.2058	0.2232	+8.42%
3	0.5127	0.4951	-3.43%
4	0.0134	0.0186	+38.68%
5	0.1461	0.1479	+1.28%
6	0.0396	0.0406	+2.42%

Tab 3.4 – Comparison of Simple Clark (CS) and reference results for the six alternative example. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	CS	Δ CS%
1	0.0544	0.0249	-54.21%
2	0.0285	0.0307	+7.63%
3	0.3246	0.2518	-22.43%
4	0.1582	0.1583	+0.07%
5	0.0950	0.1173	+23.47%
6	0.1029	0.1301	+26.44%
7	0.0146	0.0401	+174.66%
8	0.1032	0.1315	+27.45%
9	0.1186	0.1153	-2.74%

Tab 3.5 – Comparison of Simple Clark (CS) and reference results for the nine alternative example. Δ % obtained from the non-approximated values of the results.

The complete set of results from the numerical experiments on artificial networks confirm the trend of errors suggested by the three example cases. A selection of graphical results are reported in figures 3.8-3.16 where the approximation is marked CS. Low choice probabilities are mainly overestimated whilst high choice probabilities are more frequently underestimated. The errors for low choice probabilities are those of largest percentage importance. Although both overestimation and underestimation of low choice probabilities occur in the

experiments, especially with a large number of options, the underestimation has lower percentage importance. For very low choice probabilities there are cases as those excluded from figure 3.10 and 3.14 of very high percentage overestimation. Figure 3.12 reports several such cases. Figure 3.11 and 3.13 show the high relative frequency of overestimation errors greater than +20% that, as can be seen by comparing figs. 3.9, 3.11 and 3.13, increases with the number of options.

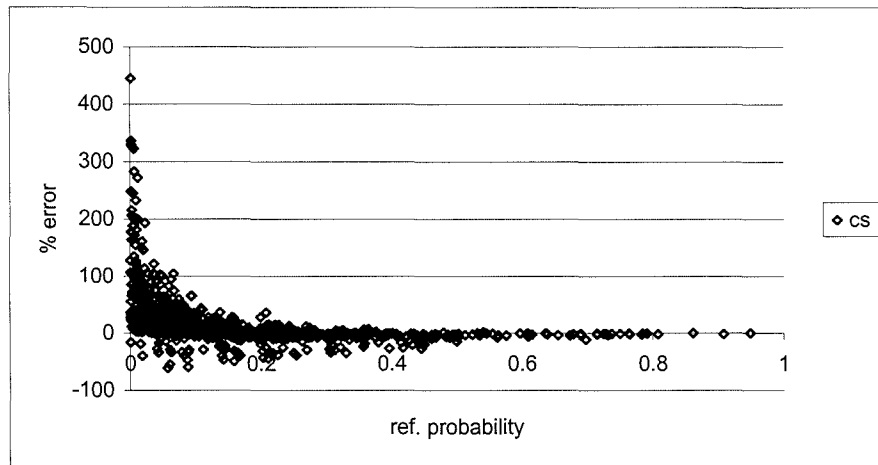


Fig. 3.8 –Percentage errors of the probabilities calculated by the simple Clark method on the 3 option cases of the first and second series of artificial networks plotted against the reference probability. Only the cases with reference probability not less than 0.001 are depicted.

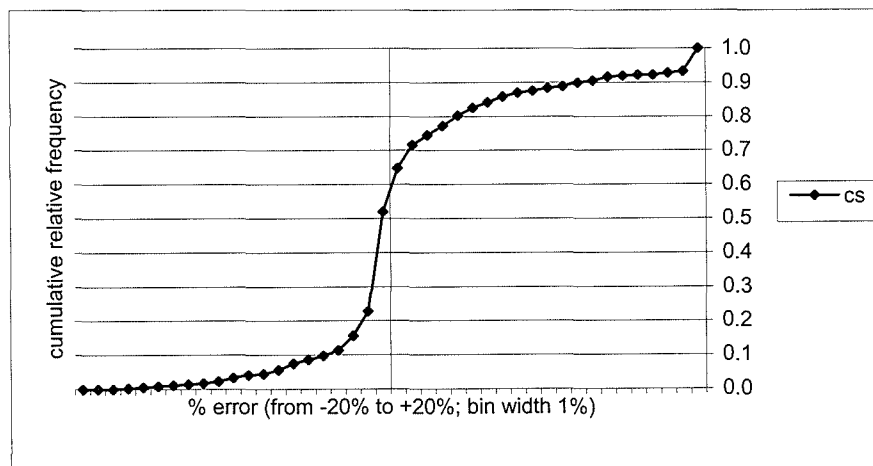


Fig. 3.9 - Cumulative relative frequencies of the percentage errors of the Simple Clark approximation on the 3 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

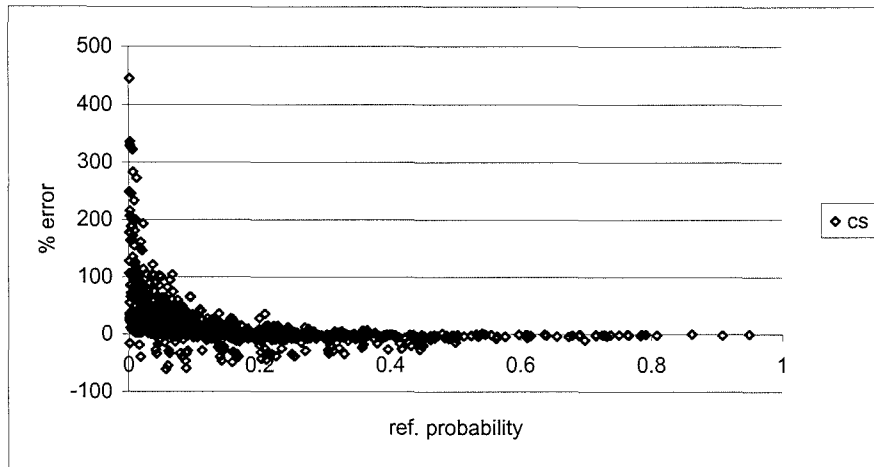


Fig. 3.10 – Percentage errors of the probabilities calculated by the simple Clark method on the 6 option cases of the first and second series of artificial networks plotted against the reference probability. Only the cases with reference probability not less than 0.001 are depicted. The graph excludes 4 data points that, for choice probabilities of the order of 10^{-3} give errors larger than 500%.

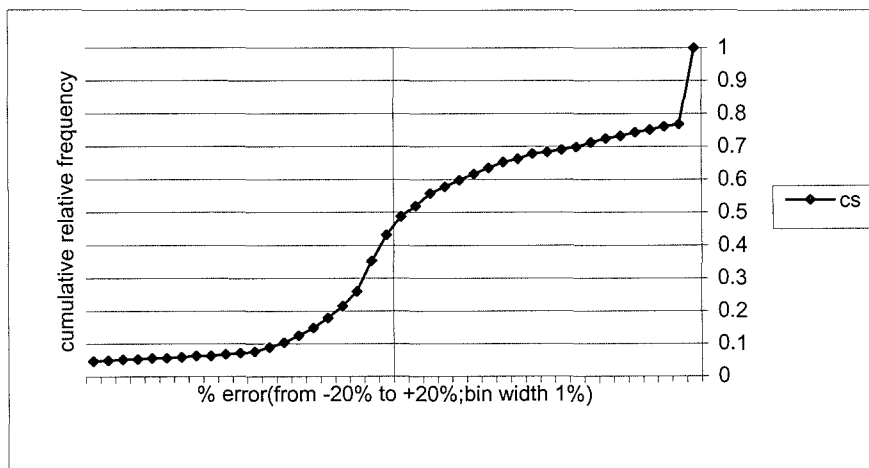


Fig. 3.11 - Cumulative relative frequencies of the percentage errors of the Simple Clark approximation on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

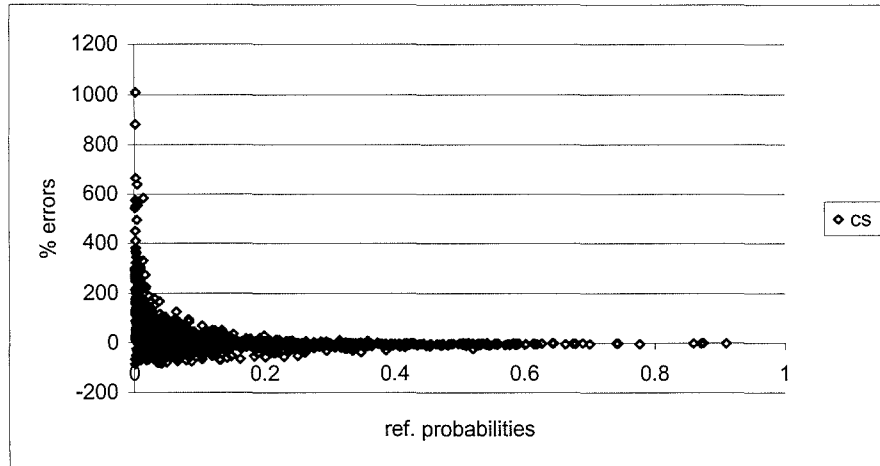


Fig. 3.12 - Percentage errors of the probability calculated by the simple Clark method plotted against the reference probability for the nine path networks of the first and second series of experiments. Only the cases with reference probability not less than 0.001 are depicted.

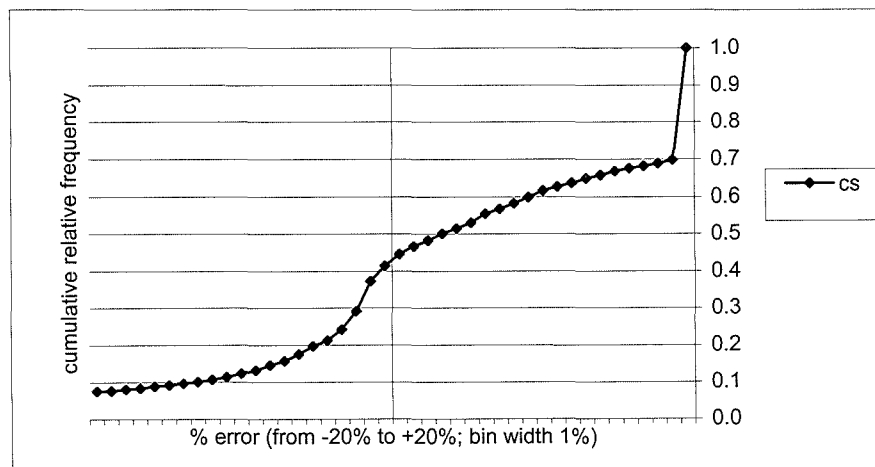


Fig. 3.13 - Cumulative relative frequencies of the percentage errors of the Simple Clark approximation on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

The sensitivity of the accuracy to the correlation of the alternatives is high, as can be seen from figs. 3.14, 3.15 and 3.16. Fig. 3.14 reports results for the extreme cases of 6 independent and 6 correlated paths in the first and second series of experiments together and show that the accuracy tends to be lower for more correlated options (i.e. more overlapping paths). In fact, the simple Clark is the approximation whose results are most sensitive to the variation of path overlapping amongst those examined. Figure 3.15 depicts the percentage inaccuracies for the cases with 12 options on the first series on networks and fig. 3.16 reports the same data for the second series of networks. They show the enlargement of the envelope of the possible errors for larger choice sets as well as the same effect when correlated options are considered, instead of independent ones. It can also be noted that the errors, particularly for large choice probabilities, typically have larger importance in the second series of networks.

In the simple Clark approximation the probabilities of choice are calculated together, thus they always sum to one.

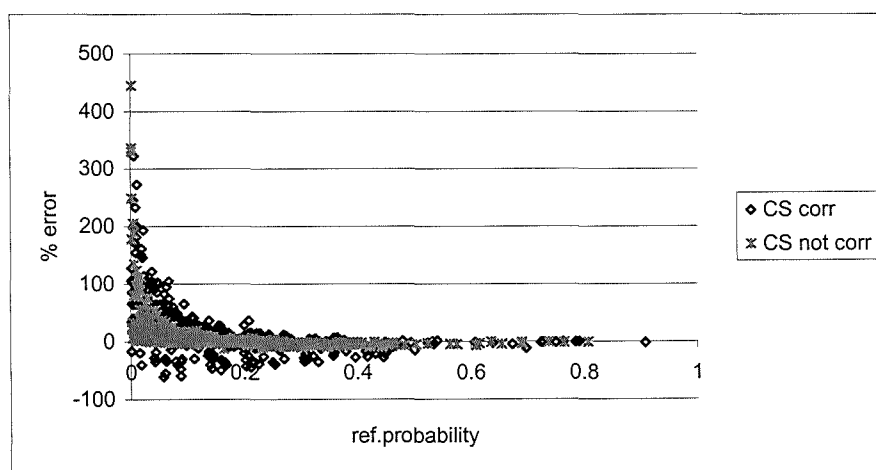


Fig. 3.14 – Detail of the graph of the percentage errors of the probability calculated by the simple Clark method on the 6 correlated and not correlated option cases of the first and second series of artificial networks plotted against the reference probability. Only the cases with reference probability not less than 0.001 are depicted. The graph excludes 4 data points that for choice probabilities of the order of 10^{-3} give percentage errors larger than 500%.

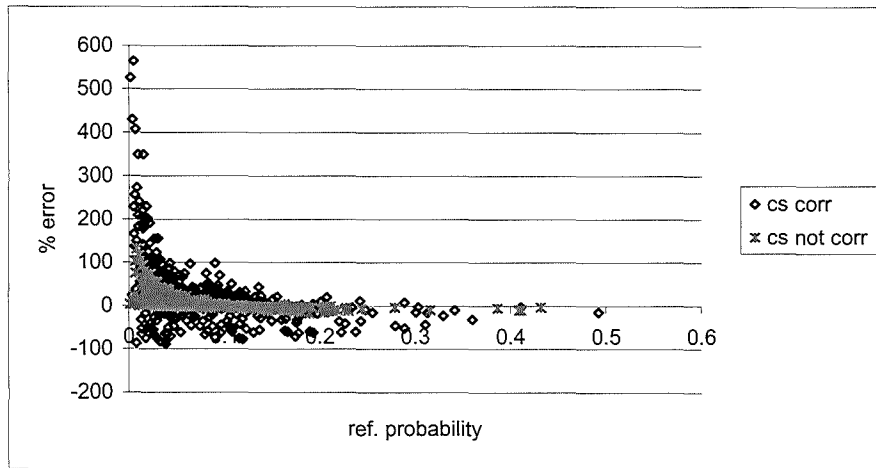


Fig. 3.15 - Percentage errors of the probability calculated by the simple Clark method plotted against the reference probability for the networks of the first series of experiments with 12 independent paths and 12 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

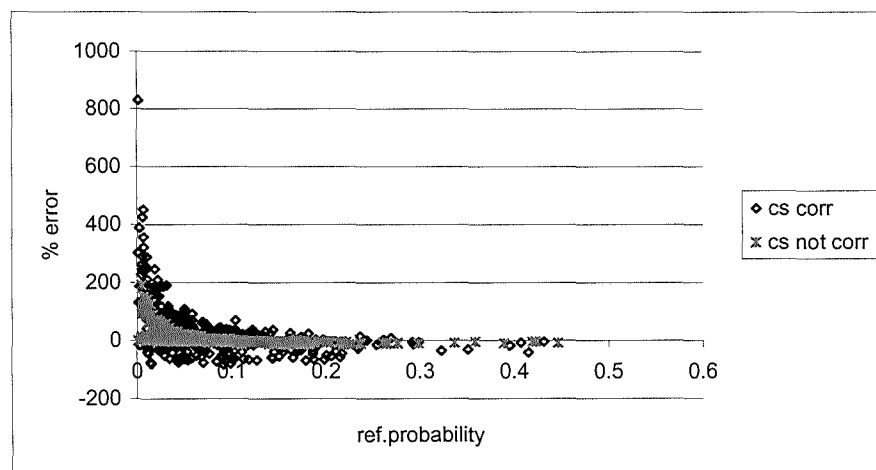


Fig. 3.16 - Percentage errors of the probability calculated by the simple Clark method plotted against the reference probability for the networks of the second series of experiments with 12 independent paths and 12 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

3.6.4.2 Accuracy of the Improved Clark Approximation

The accuracy of the Clark approximation, in the implementation referred to here as improved Clark, has been investigated extensively (Clark, 1961; Daganzo and Sheffi, 1977; Horowitz *et al.*, 1982; Kamakura, 1989; Langdon, 1984a,b). The work in the literature suggests that its accuracy is satisfactory when the variables have not very different variances and means not very close to each other and that positive correlation should improve the results. Also cases with few alternatives are more accurately approximated. The overall results seem to vary, but little, with the order of calculation (Daganzo *et al.*, 1977; Lerman and Manski, 1981; Langdon, 1984a,b). However, when the above conditions are not met the approximation has been found to give incorrect results and, in general, it tends to overestimate low choice probabilities.

Since the literature does not suggest a particular order of calculation, the results reported in the first part of this section (where the approximation is marked CI in graphs and diagrams) have been obtained carrying out the calculations with the alternatives considered simply in the order they have been coded in the input file to the program. In the second part of the section the possibility of an optimal variate processing sequence is considered.

The results for the examples with 3, 6 and 9 options, reported respectively in table 3.6, 3.7 and 3.8 show, although in a limited way, the general tendency of the inaccuracies of this approximation: small actual probabilities are mainly overestimated and larger actual probabilities tend to be underestimated. The errors are however of much smaller percentage importance than with the heuristic simple Clark method.

This trend can be seen, for instance, by observing the overestimation of option 2 in table 3.6 and the underestimation of the two other choice probabilities. The choice probabilities of option 1, 2, and 7 in table 3.8 that have actual low choice probabilities are overestimated, particularly in the cases of option 2 and 7, but option with larger choice probabilities have them mostly underestimated.

Option	Num.int.	CI	Δ CI%
1	0.3117	0.3086	-0.98%
2	0.0365	0.0399	9.27%
3	0.6519	0.6515	-0.05%

Tab 3.6 – Comparison of Improved Clark and reference results for the three alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	CI	Δ CI%
1	0.0824	0.0849	2.93%
2	0.2058	0.2036	-1.09%
3	0.5127	0.5095	-0.62%
4	0.0134	0.0145	8.35%
5	0.1461	0.1467	0.47%
6	0.0396	0.0408	3.06%

Tab 3.7 – Comparison of Improved Clark and reference results for the six alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	CI	Δ CI%
1	0.0544	0.0669	+22.8577
2	0.0285	0.0435	+52.5141
3	0.3246	0.2856	-12.0063
4	0.1582	0.1419	-10.2549
5	0.0950	0.0924	-2.7619
6	0.1029	0.1108	+7.7239
7	0.0146	0.0287	+96.6074
8	0.1032	0.1159	+12.2677
9	0.1186	0.1143	-3.58173

Tab 3.8 – Comparison of Improved Clark and reference results for the nine alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Considering all the results for the first and second series of experiments on artificial networks (see the selection of graphical results reported in figs. 3.17, 3.18, 3.19 and 3.20), the tendency towards the overestimation of low choice probabilities is clear and accompanied mainly by the underestimation of the high ones that is, however, of much smaller percentage importance. This trend of errors is in accordance with the results on the accuracy of the improved Clark method in the literature (see *e.g.* Langdon, 1984a, b). Also graphs like 3.18 and 3.20 show that most underestimation errors are of limited percentage importance whilst a large proportion of the overestimation ones have larger percentage importance. Comparing the same graphs also shows that the proportion of larger percentage overestimation errors increases with the number of choice options. Going back to the corresponding figures 3.17 and 3.19, however, it is also clear the many large percentage errors affect very small actual choice probabilities.

The approximation results decrease in accuracy as the correlation between the alternatives increases as can also be seen from fig. 3.21, which reports results for the extreme cases of independent and completely correlated paths. In fact, this trend is even clearer in figs. 3.22 and 3.23 that report the same sort of data separately for the first and second series of networks in the nine path case. The accuracy of most approximations examined is in some way sensitive to the variation of path overlapping but the Clark improved approximation shows a very important such sensitivity. Comparing figures 3.22 and 3.23 shows that the inaccuracies reported for the first series of artificial networks are smaller than those for the second series. This behaviour is common to all approximations tested.

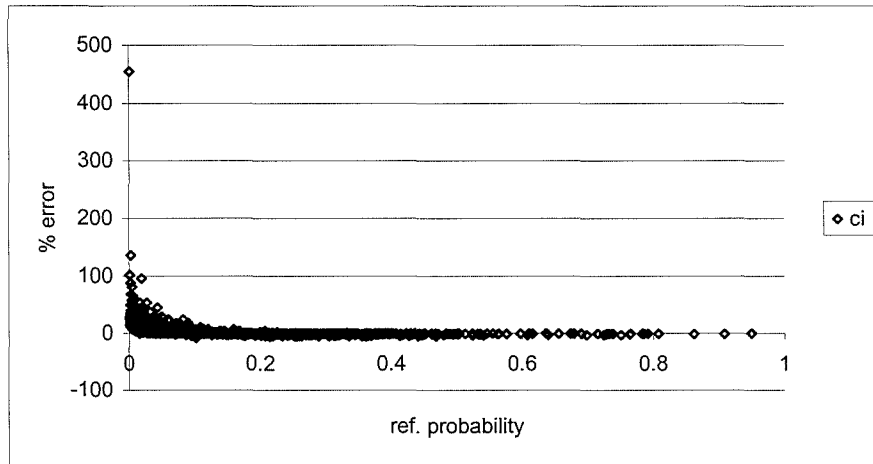


Fig. 3.17 - Percentage errors of improved Clark approximation on the six option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

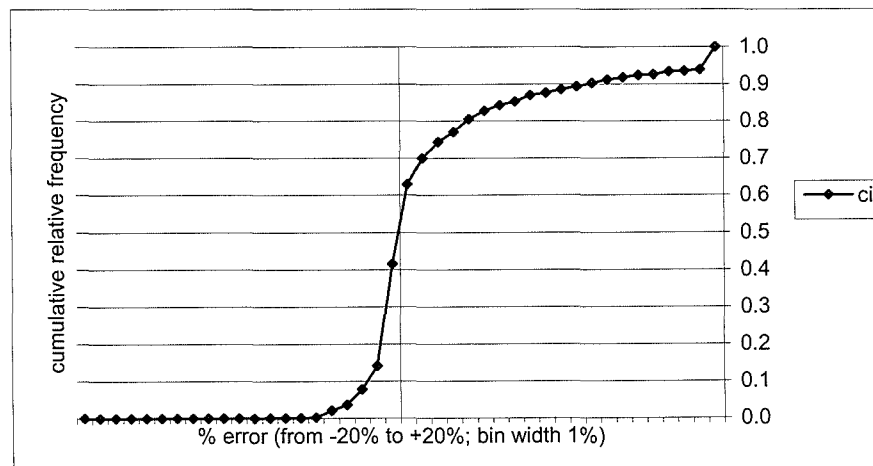


Fig. 3.18 - Cumulative relative frequencies of the percentage errors of the improved Clark approximation on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

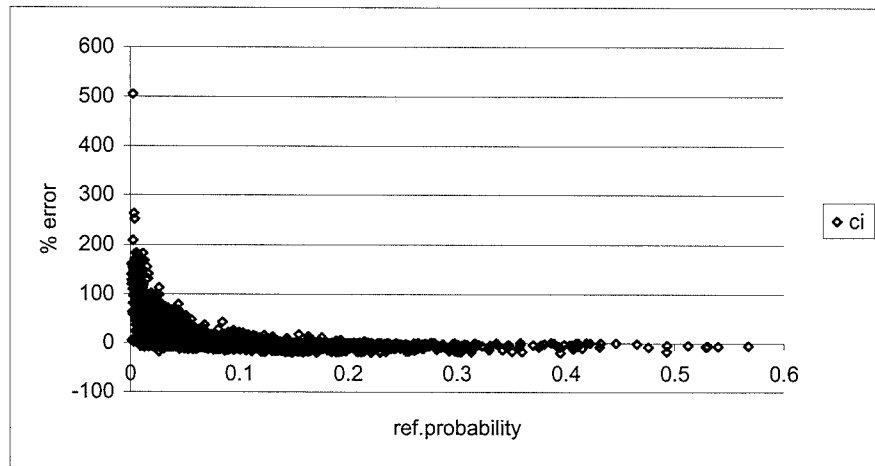


Fig. 3.19 – Percentage errors of improved Clark approximation on the 12 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

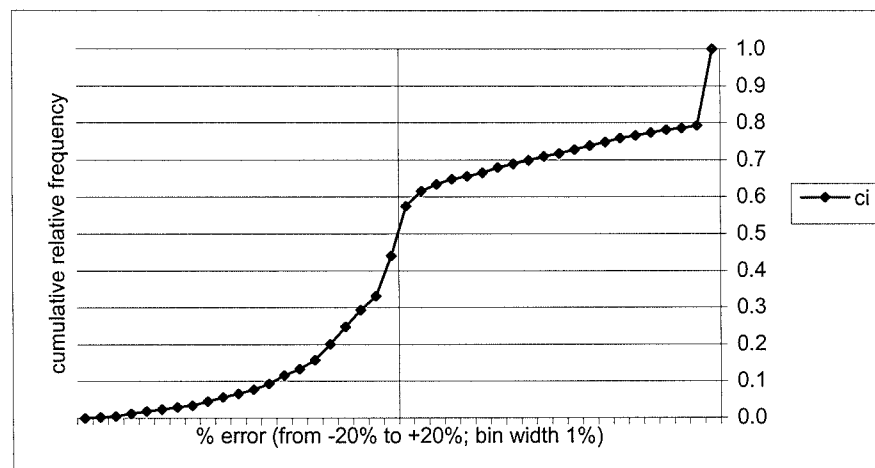


Fig. 3.20 – Cumulative relative frequencies of the percentage errors of the improved Clark approximation on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

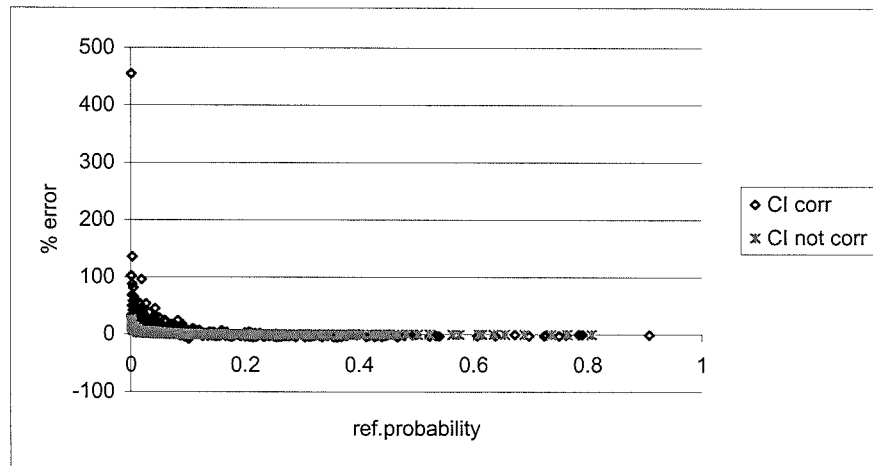


Fig. 3.21 - Percentage errors of the probability calculated by the improved Clark method plotted against the reference probability for the networks of the second series of experiments with 6 independent paths and 6 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

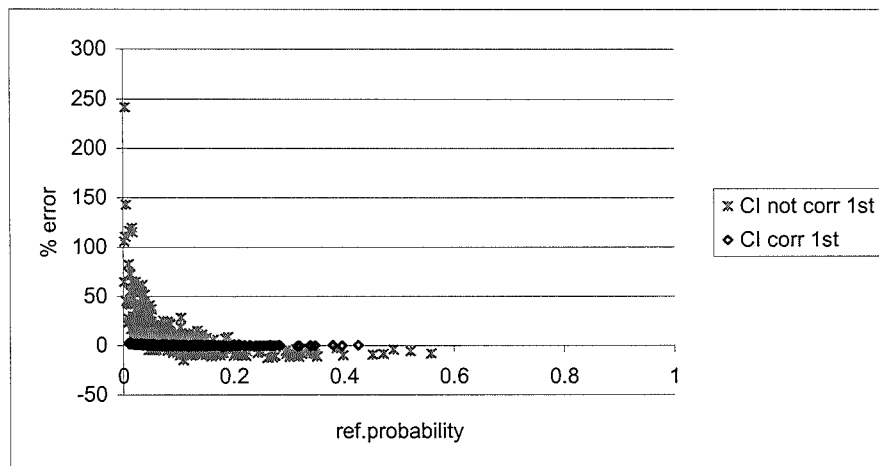


Fig. 3.22 - Percentage error of the probability calculated by the improved Clark method against the reference probability for the networks of the first series of experiments with 9 independent paths and 9 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

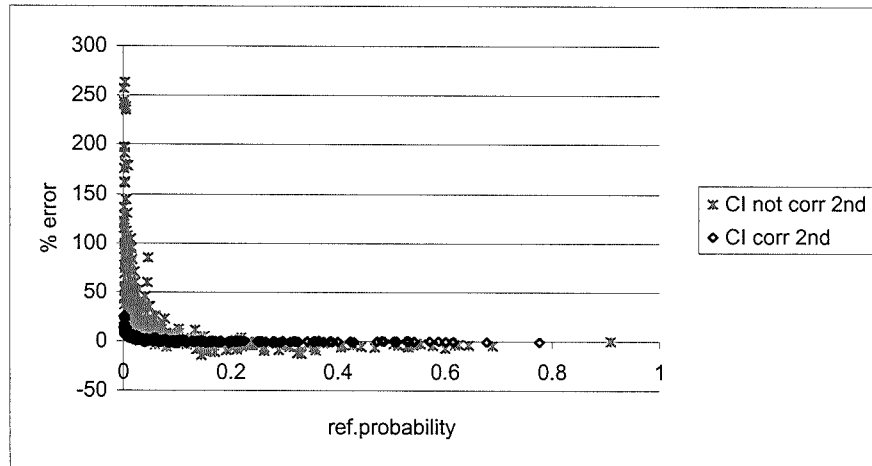


Fig. 3.23 – Percentage error of the probability calculated by the improved Clark method against the reference probability for the networks of the second series of experiments with 9 independent paths and 9 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

Options	Mean CI	St.dev. CI	Max CI	Min CI
3	0.9962	0.0037	1.0045	0.9867
6	0.9890	0.0074	1.0177	0.9709
9	0.9915	0.0107	1.0242	0.9599
12	1.0063	0.0221	1.0543	0.9797
15	1.0159	0.0346	1.1102	0.9788

Table 3.9 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the improved Clark approximation on the first and second series of artificial networks.

The final improved Clark results presented are obtained by normalising to their sum the results directly given by the approximation. The importance of such normalisation has been tested collecting the data on the difference from unity of the sum of the non-normalised results. The larger is the error in the sum of the probabilities the more important is the correction the normalisation brings about and, possibly, the errors it introduces, although the absolute correction is proportional to the value of the probability that is corrected. The percentage importance of the correction is the same for all options as it is determined by the error in the sum. A

summary of the statistics for the errors in the sum for the improved Clark approximation are shown in table 3.9. Although, on average, the sum of non-normalised probabilities is not too different from one, it varies over a rather wide range, especially for the cases with choice sets of 12 and 15.

The literature on the improved Clark method suggests that the order in which the variates are processed has a very limited effect on the results and does not offer a possible optimal processing order. Here a number of processing orders defined by simple rules has been tested to try to reduce the inaccuracies given by the approximation especially for small choice probabilities.

The overall accuracy of the results depends on how good the Normal distribution is as an approximation to the maximum of two Normal variates each time the approximation is applied. Langdon (1984a), considering the case of a pair of correlated Normal variates, compared the actual shape of the maximum distribution with the approximating Normal distribution resulting from the Clark approximation and noted that the approximation is often not good. The work of Clark (1961) provides a direct means to check the goodness of the Normal approximation to the maximum of two Normal variates by giving the formulae for their skewness and kurtosis. However these formulae are rather complex and it is difficult to use them to characterise a single parameter to inform a criterion for a processing order.

Therefore a number of candidate orders have been selected considering previous results in the literature on the viability of the Normal as an approximation of the maximum of two Normal variates. In particular, it has been considered processing first the variates with the smallest variance of the difference as these elements should improve the near-Normality of the actual maximum. Moreover, it has been considered processing first the variates with the maximum ratio of the absolute difference of the means over the absolute difference of the variances, a measure that tries to capture the variates with the closest variances and the most different means. Alternatively it has been considered processing independent variates first followed by those with largest correlation, as in the first case the approximation of Clark is exact and in the latter its accuracy should be improved according to the consideration

of Daganzo *et al.* (1977). Finally, it has been considered processing first the variates with the largest absolute difference of the mean, as the approximation should work better with means not very close to each other, or the opposite: those with the minimum absolute difference of the means are processed first to check the contrast of the orders. The latter orders were included in the investigations although in the test cases used in this work the means of the utilities of the options rather similar to each other, thus they should be expected to have a limited effect.

The analysis of the results obtained on the two series of artificial networks employed in these investigations showed little variation of the envelope of the percentage errors with the variate order and little difference from the envelope obtained by processing the variates simply in the order they have been coded in the input file to the program. This confirms the result in the literature that the effect of rearranging the variates is very limited and does not allow suggesting a possible optimal processing order. Examples of the effects of the variate reordering are given in graphical form in figures 3.24 and 3.25 that depict a selection of the results obtained with the first and the last order mentioned above, marked respectively i1 and i5.

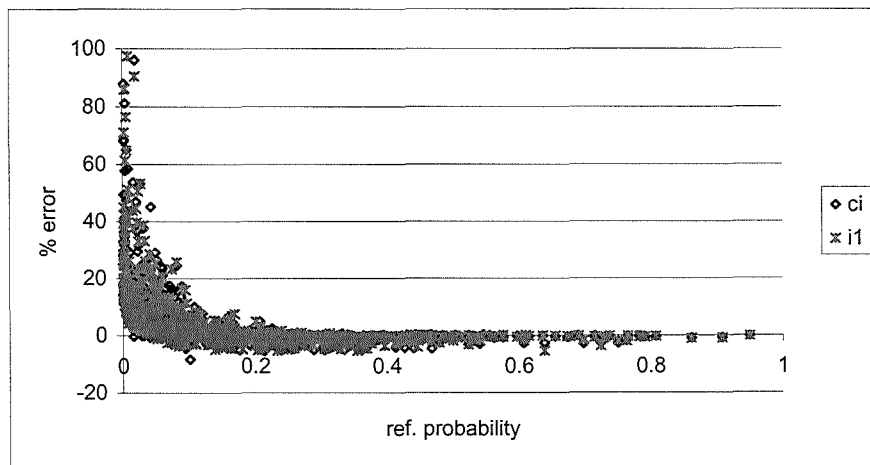


Fig. 3.24 – Percentage error of the probability calculated by the improved Clark method with an instance of random order (ci) and when the variates with the smallest variance of the difference (i1) against the reference probability on the 6 option cases of the first and second series of artificial networks. Only the cases with reference probability not less than 0.001 are depicted. The percentage errors larger than 100% (recorded for small choice probabilities) have been excluded from the graph.

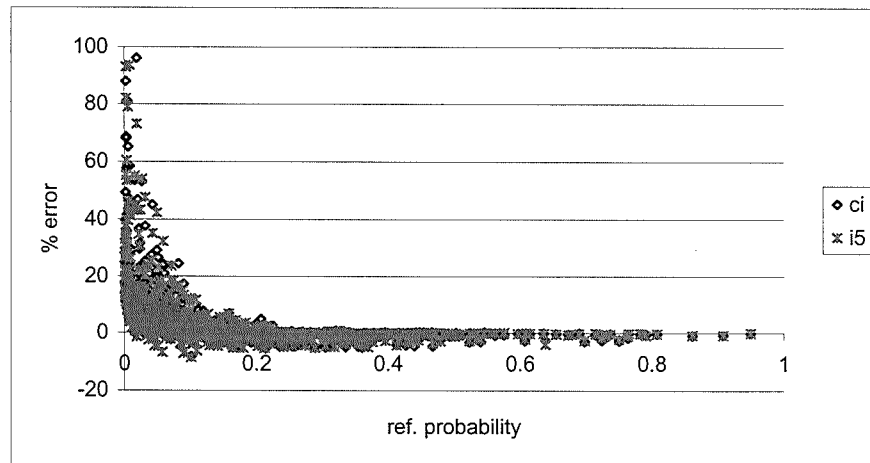


Fig. 3.25 - Percentage error of the probability calculated by the improved Clark method with an instance of random order (*ci*) and when the variates with the minimum absolute difference of the means are processed first (*i5*) against the reference probability on the 6 option cases of the first and second series of artificial networks. Only the cases with reference probability not less than 0.001 are depicted. The percentage errors larger than 100% (recorded for small choice probabilities) have been excluded from the graph.

3.6.4.3 Accuracy of the Mendell-Elston Approximation

The results on the accuracy of the Mendell-Elston approximation in the literature suggest that, when it is used for evaluating MVN integrals, its precision should decrease when the variates of the MVN distribution are correlated. This was remarked by Rice *et al.* (1979) noting that with larger correlations the assumption of Normality of the conditional distribution is less tenable. Terada and Takahashi (1988), made the same remark independently, examining the departure from Normality of the conditional distribution of one of the variates in a bivariate Normal distribution given a truncation of the other.

The accuracy of the approximation in cases with a limited number of options (3 or 5) has also been discussed by Kamakura (1989), introducing the Mendell-Elston approximation in transportation, who found it more precise than the method of Clark. The rationale for the good accuracy of the Mendell-Elston approximation, especially as compared to the method of Clark, is suggested by the discussion of Langdon (1984b) on his method, that approximates as Normal the same type of

distribution: the Normal distribution is generally a better approximation to the distribution of a variate in a MVN conditional on the truncation of one of the others than to the maximum of a two Normal variates (which is the way it is used in the Clark method).

As discussed in the description of the approximation, two alternative optimal processing orders have been suggested in the literature. Kamakura (1989), proposing the approximation for use in econometric applications, borrowed the order suggested by Langdon for his method. Rice *et al.* (1979), after tests on positively equicorrelated variates, suggested that including in the calculations the variate giving the smallest marginal integral first could be of advantage. Here this processing order is considered in general cases as a heuristic. The results obtained with the Kamakura (1989) order are presented first (and are marked ME-K in tables and figures), then they are briefly compared to those obtained with an instance of a random order (marked ME non-opt) and, finally they are compared with those from the heuristic order suggested by Rice *et al.* (marked in tables and figures ME-R).

The data for the three examples used throughout this chapter, are reported in tables 3.10, 3.11 and 3.12 and suggest that with the order put forward by Kamakura, the ME approximation gives very moderate errors.

Option	Num.Int.	ME-K	Δ ME-K%
1	0.3117	0.3130	+0.44%
2	0.0365	0.0362	-0.66%
3	0.6519	0.6507	-0.17%

Tab 3.10 – Comparison of reference results with those of the Mendell-Elston approximations for the three alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	ME-K	Δ ME-K%
1	0.0824	0.0833	+0.99%
2	0.2058	0.2055	-0.14%
3	0.5127	0.5113	-0.26%
4	0.0134	0.0136	+1.12%
5	0.1461	0.1463	+0.20%
6	0.0396	0.0400	+0.95%

Tab 3.11 – Comparison of reference results with those of the Mendell-Elston approximations for the six alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	ME-K	Δ ME-K%
1	0.0544	0.0539	-0.87%
2	0.0285	0.0284	-0.61%
3	0.3246	0.3207	-1.18%
4	0.1582	0.1562	-1.26%
5	0.0950	0.0946	-0.43%
6	0.1029	0.1044	+1.49%
7	0.0146	0.0148	+1.49%
8	0.1032	0.1053	+2.02%
9	0.1186	0.1217	+2.63%

Tab 3.12 – Comparison of reference results with those of the Mendell-Elston approximations for the nine alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

The general pattern of inaccuracies can be observed from the aggregate data from the experiments on the test networks shown in the following figures. Percentage inaccuracies are smaller in the first series of networks and larger for the second except in the cases of 12 and 15 paths, where the errors are of similar importance. However, they are generally of limited percentage importance and larger percentage errors are reported for small choice probabilities as in figs. 3.26 to 3.29.

In fact, most often the trend of percentage errors for the first series of tests has the shape of a wedge with the large end towards the low choice probabilities: these are

mainly overestimated whilst larger probabilities are mainly underestimated. Underestimation of low choice probabilities is shown for cases with 12 and 15 uncorrelated options.

The second series of tests showed, besides a similar wedge shaped trend, also the underestimation of very low choice probabilities for the cases with 6 and 9 choice options. The wedge shaped trend is repeated in the cases with 12 and 15 choice options where only in cases of uncorrelated options the overestimation of low choice probabilities changes into underestimation.

The quality of the results improves when the choice options are less or not correlated, which also causes the general wedge trend of percentage errors to be substituted by underestimation of limited importance of low choice probabilities that is hidden by errors in other cases. This change of trend is particularly evident in the cases of 12 and 15 options, where the errors for low choice probabilities are mainly of underestimation in less correlated cases and tend to be mainly overestimation errors when the options are correlated. As in figure 3.30 the dispersion of the points marking the percentage errors against the reference choice probabilities generally increases with the correlation of the paths in the test networks as does the percentage importance of the envelopes of the errors.

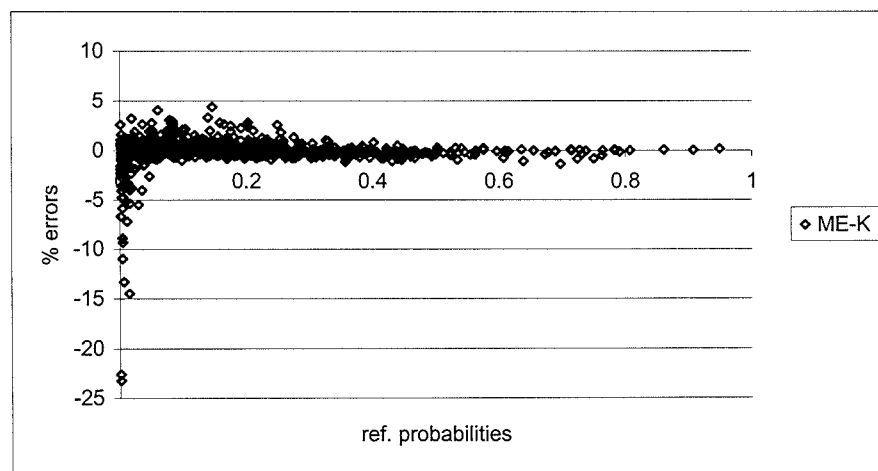


Fig. 3.26 - Percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

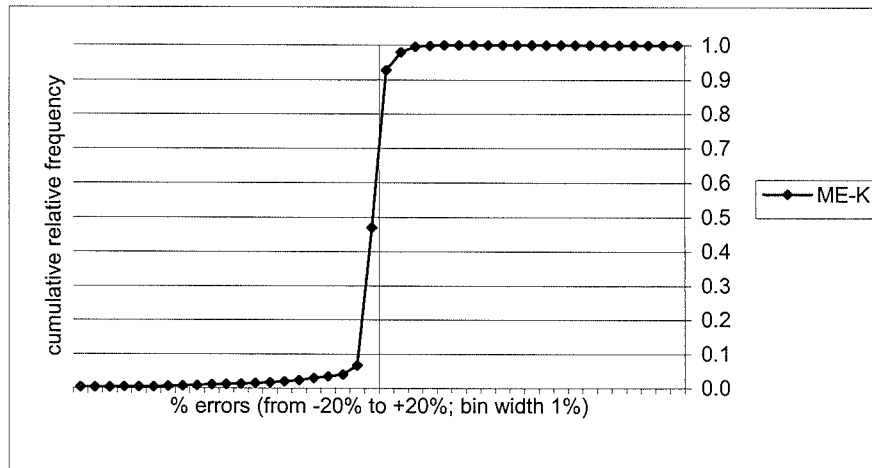


Fig. 3.27 – Cumulative relative frequencies of the percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

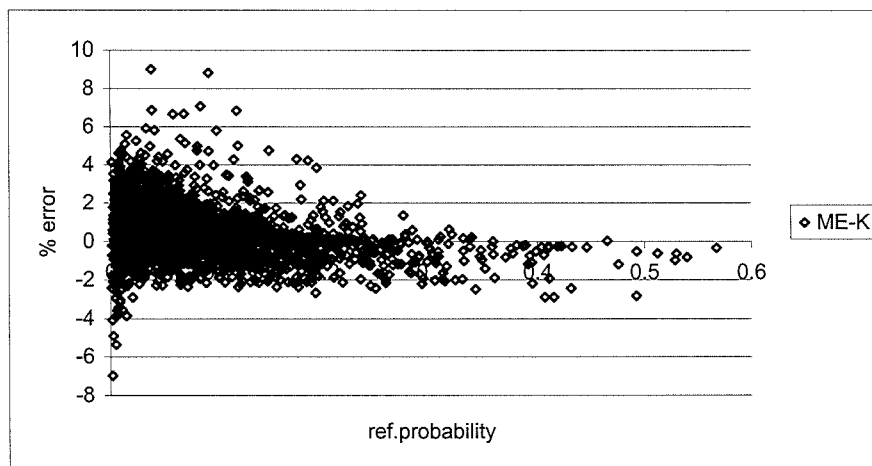


Fig. 3.28 - Percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) on the 12 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

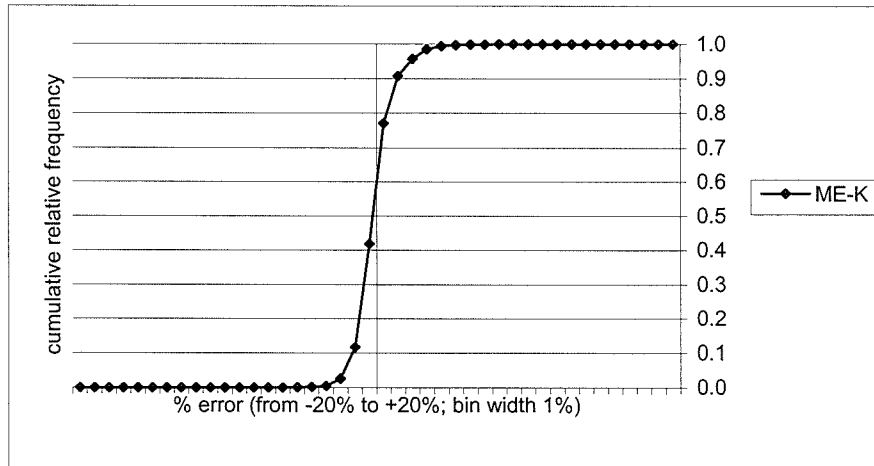


Fig. 3.29 – Cumulative relative frequencies of the percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

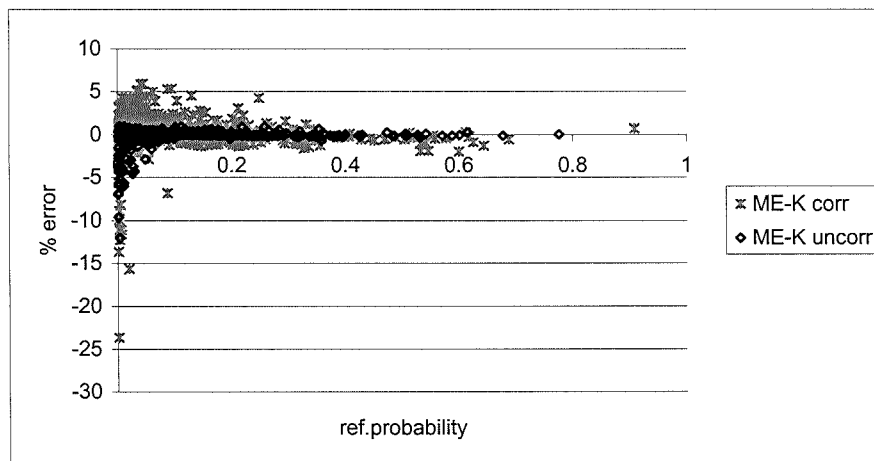


Fig. 3.30 - Percentage error of the probability calculated by the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) against the reference probability for the networks of the first series of experiments with 9 independent paths and 9 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

The results presented are obtained by normalising the results directly given by the approximation to their sum. As shown in table 3.13, that reports the summary statistics on the sum of the probabilities calculated before normalisation, the importance of the normalisation increases with the number of options in the choice set.

Options	Mean ME-K	St.dev. ME-K	Max ME-K	Min ME-K
3	1.0008	0.0020	1.0084	0.9978
6	1.0071	0.0056	1.0288	0.9956
9	1.0116	0.0097	1.0468	0.9871
12	1.0213	0.0114	1.0533	0.9860
15	1.0234	0.0138	1.0667	0.9840

Table 3.13 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the Mendell-Elston approximation with the Kamakura order (ME-K) on the first and second series of artificial networks.

The effectiveness of the optimised order suggested by Kamakura for carrying out the calculations is shown in figure 3.31 that is an example of comparison between the results obtained with the optimised sequence and those obtained when the variates are processed following the order in which they are listed in the input file to the program performing the calculations, an instance of a random order. The non-optimised processing sequence results in a larger envelope of inaccuracies and particularly in larger inaccuracies for low choice probabilities that are more underestimated as can be seen also from figure 3.32 depicting the cumulative relative frequencies of the percentage errors.

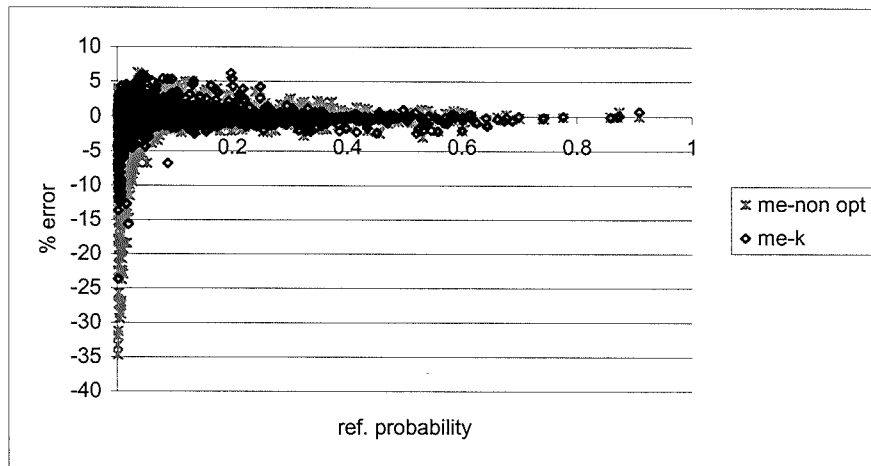


Fig. 3.31 - Percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) and with an instance of random order (ME non-opt) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

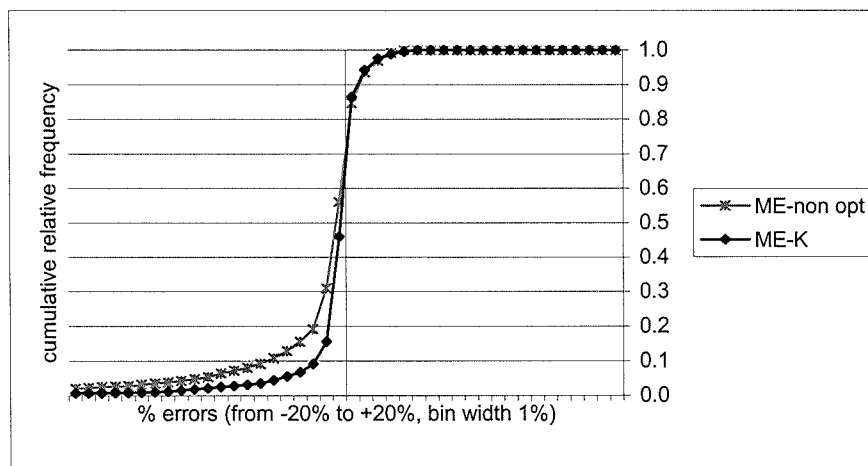


Fig. 3.32 – Cumulative relative frequencies of the percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) and with an instance of random order (ME non-opt) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

The alternative ordering suggested by Rice *et al.* is considered here as a heuristic but it is also interesting as it parallels that proposed for the Taylor series expansion approximations that have similarities with the Mendell-Elston approximation. Moreover there is an interesting difference between the method of Langdon, from which Kamakura borrowed his processing sequence, and that of Mendell-Elston. In the method of Langdon when a binomial choice is considered the conditional distributions of the utilities for both options considered as chosen are important since they will be used in later stages of calculation (see the description in section 3.5.4). The method of Mendell and Elston can be thought of as working, for each option, only along one of the two outer branches of the tree used to describe the method of Langdon. Each time a marginal normal integral is computed, which corresponds to a binomial choice computation in the method of Langdon, only the distribution of the remaining variates conditional on a variate being lower than a threshold is considered but not the opposite.

The calculation order resulting from the observation of Rice *et al.* has been tested for accuracy by applying it to the same test cases used throughout this chapter. Also the opposite order and the order opposite to that suggested by Kamakura have been tested. The results of the calculations with the orders opposite to those tested as optimal are not reported as they merely underlined the importance of using these elements as parameters to determine the calculation order by giving worse trends of percentage errors (especially for low actual choice probabilities, approximately less than 10%) than in the optimised order cases, similar to those obtained using as instance of a random order that in which the options have been coded in the input file to the program.

The alternative optimal order outperformed, although in a limited way, the order of Kamakura. The tables below reconsider part of the ME-K order results already listed in tables 3.10-3.12 and compare them with those of the heuristic Rice *et al.* order (marked ME-R) in the corresponding cases.

Option	Num.Int.	ME-R	Δ ME-R%	Δ ME-K%
1	0.3117	0.3119	+0.08%	+0.44%
2	0.0365	0.0366	+0.27%	-0.66%
3	0.6519	0.6515	-0.05%	-0.17%

Tab 3.14 – Comparison of reference results with those of the Mendell-Elston approximations for the three alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	ME-R	Δ ME-R %	Δ ME-K %
1	0.0824	0.0832	+0.95%	+0.99%
2	0.2058	0.2056	-0.13%	-0.14%
3	0.5127	0.5113	-0.27%	-0.26%
4	0.0134	0.0135	+1.00%	+1.12%
5	0.1461	0.1465	+0.27%	+0.20%
6	0.0396	0.0399	+0.84%	+0.95%

Tab 3.15 – Comparison of reference results with those of the Mendell-Elston approximations for the six alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Opt	Num Int	ME-R	Δ ME-R %	Δ ME-K %
1	0.0544	0.0547	+0.55%	-0.87%
2	0.0285	0.0285	-0.06%	-0.61%
3	0.3246	0.3229	-0.51%	-1.18%
4	0.1582	0.1569	-0.77%	-1.26%
5	0.0950	0.0946	-0.44%	-0.43%
6	0.1029	0.1035	+0.62%	+1.49%
7	0.0146	0.0148	+1.59%	+1.49%
8	0.1032	0.1042	+0.94%	+2.02%
9	0.1186	0.1198	+1.06%	+2.63%

Tab 3.16 – Comparison of reference results with those of the Mendell-Elston approximations for the nine alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

From the tables there seem to be little to choose between the two orderings, although the ME-R seems to give marginally better results in several cases. However, comparing the aggregate results for both series of test networks the trend of the errors for the ME-R is contained in that of the ME-K order (see figures 3.33 to 3.36) and the effect of the ME-R is that of reducing the underestimation of low choice probabilities and the dispersion of the percentage errors. The first effect is lost, however, as the number of options and the number of times the approximation is applied increase (compare figure 3.33 and 3.35). It should be noted that, although this cannot be noticed from the figures reported, the improvement in the quality of the approximation is particularly evident for the networks of the second series. The comparison on the cumulative relative frequency distribution of the percentage errors reported in figures 3.34 and 3.36 shows that the importance of the errors for the ME-R order is slightly but consistently lower than for the ME-K one.

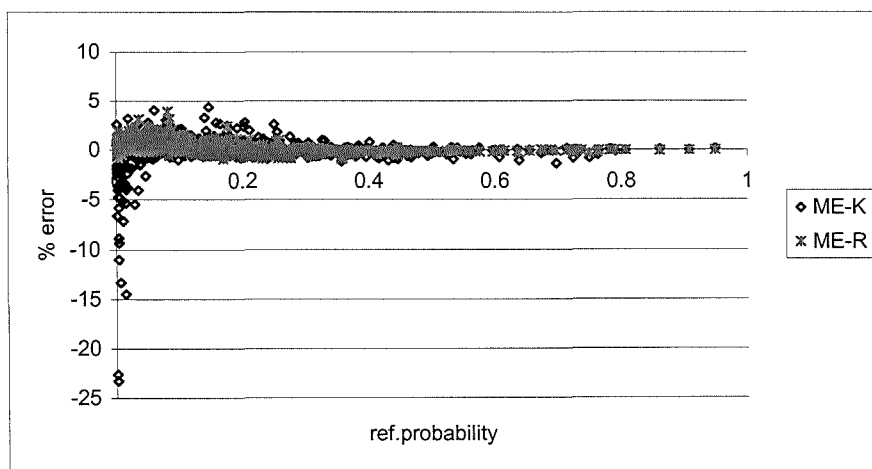


Fig. 3.33 - Percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) and with the Rice et al. order (ME-R) on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

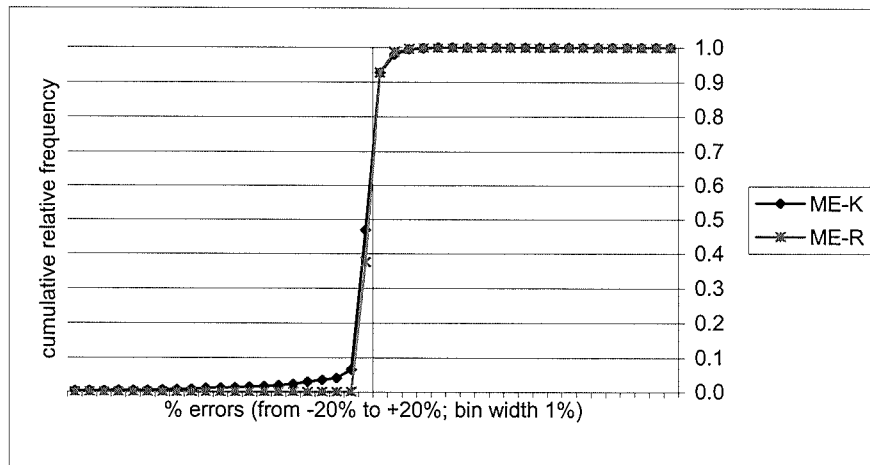


Fig. 3.34 - Cumulative relative frequencies of the percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) and with the Rice et al. order (ME-R) on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

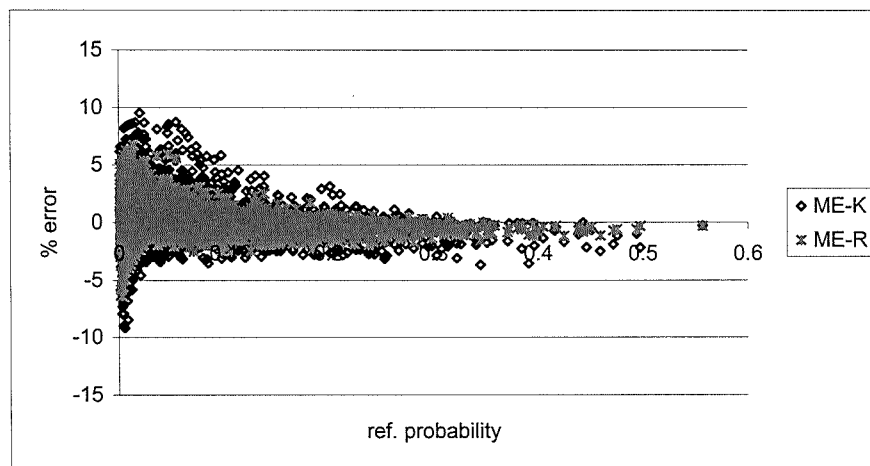


Fig. 3.35 - Percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) and with the Rice et al. order (ME-R) on the 15 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

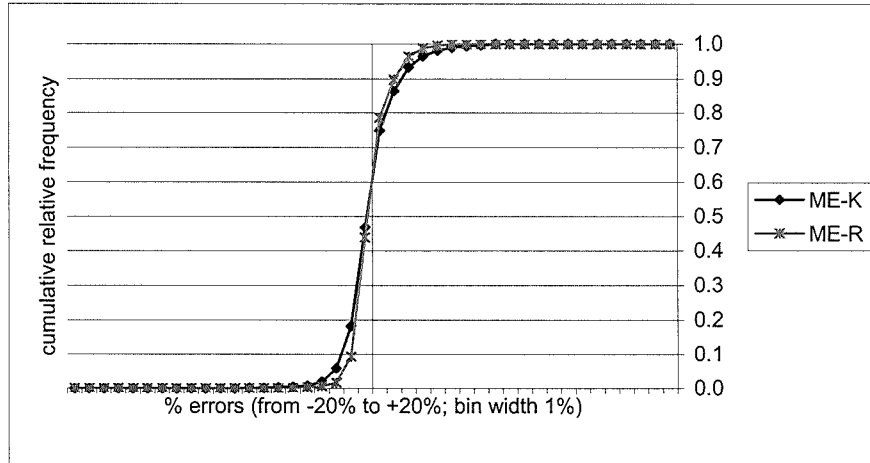


Fig. 3.36 – Cumulative relative frequencies of the percentage errors of the Mendell-Elston approximation with the optimised order of Kamakura (ME-K) and with the Rice et al order (ME-R) on the 15 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

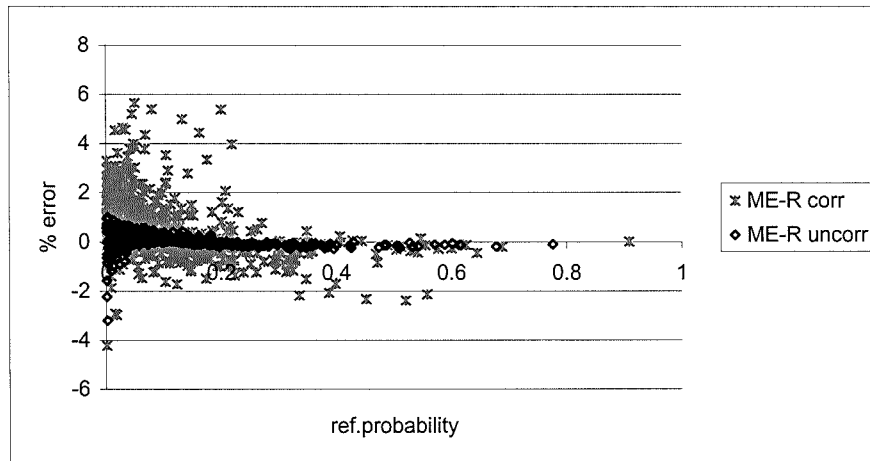


Fig. 3.37 - Percentage error of the probability calculated by the Mendell-Elston approximation with the optimised order of Rice et al. (ME-R) against the reference probability for the networks of the first series of experiments with 9 independent paths and 9 correlated paths. Only the cases with reference probability not less than 0.001 are depicted.

approximated by the Normal distribution than the distribution of the maximum of two Normal variates (that is used in the Clark method).

The separated split approximation has been implemented using the optimised processing order suggested in Langdon (1984a,b) with the next option to consider in the calculations chosen before each conditional marginal choice. The results are marked SP in the following tables and figures.

The results for the three examples used throughout this chapter, reported in tables 3.18, 3.19 and 3.20, show a good agreement with the reference values.

Option	Num.Int.	SP	Δ SP %
1	0.3117	0.3151	+1.12%
2	0.0365	0.0365	+0.02%
3	0.6519	0.6484	-0.53%

Tab 3.18 – Comparison of reference results with those of the Langdon approximations for the three alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num.Int.	SP	Δ SP %
1	0.0824	0.0833	+0.99%
2	0.2058	0.2057	-0.03%
3	0.5127	0.5123	-0.07%
4	0.0134	0.0138	+2.52%
5	0.1461	0.1448	-0.88%
6	0.0396	0.0402	+1.47%

Tab 3.19 – Comparison of reference results with those of the Langdon approximations for the six alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num. Int.	SP	Δ SP %
1	0.0544	0.0547	+0.45%
2	0.0285	0.0287	+0.47%
3	0.3246	0.3247	+0.06%
4	0.1582	0.1578	-0.25%
5	0.0950	0.0946	-0.36%
6	0.1029	0.1026	-0.32%
7	0.0146	0.0152	+3.95%
8	0.1032	0.1016	-1.57%
9	0.1186	0.1202	+1.37%

Tab 3.20 – Comparison of reference results with those of the Langdon approximations for the nine alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

The aggregate results for the first and second series of experiments confirm the good quality of the approximation and highlight the tendency to produce larger percentage errors for lower actual probabilities (due both to overestimation and underestimation with the overestimation being typically of larger percentage importance) as shown in figs. 3.38, 3.39, 3.40 and 3.41. High probabilities are often slightly underestimated. However, except with very small actual probabilities the percentage errors are of rather limited importance.

The separated split approximation is sensitive to the correlation of the options as shown by the envelopes of the errors reported in figures 3.42 and 3.43 that depict results for cases with 9 options. The same figures show how the importance of the errors tends to be larger for the second series of networks.

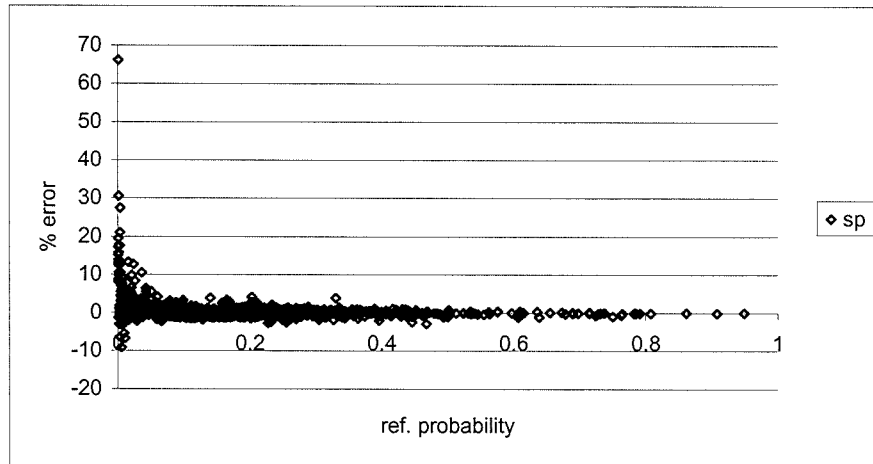


Fig. 3.38 – Percentage errors of the separated split approximation (with optimised and non-optimised calculation order) on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

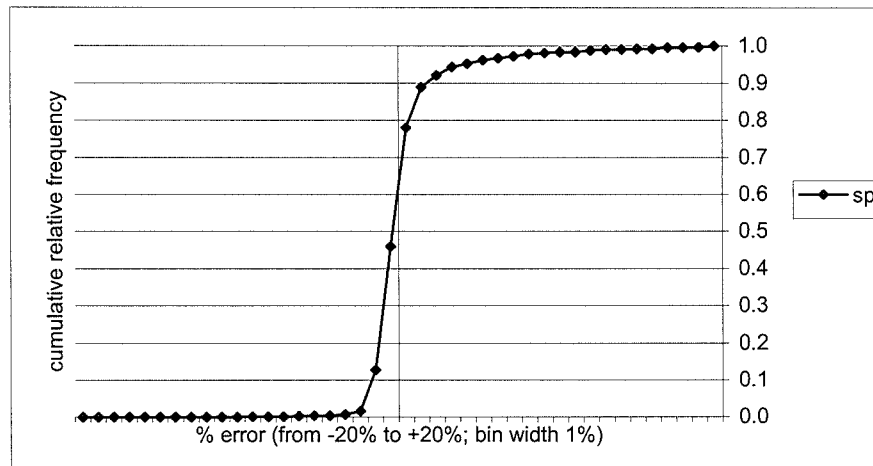


Fig. 3.39 – Cumulative relative frequencies of the percentage errors of the separated split approximation on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

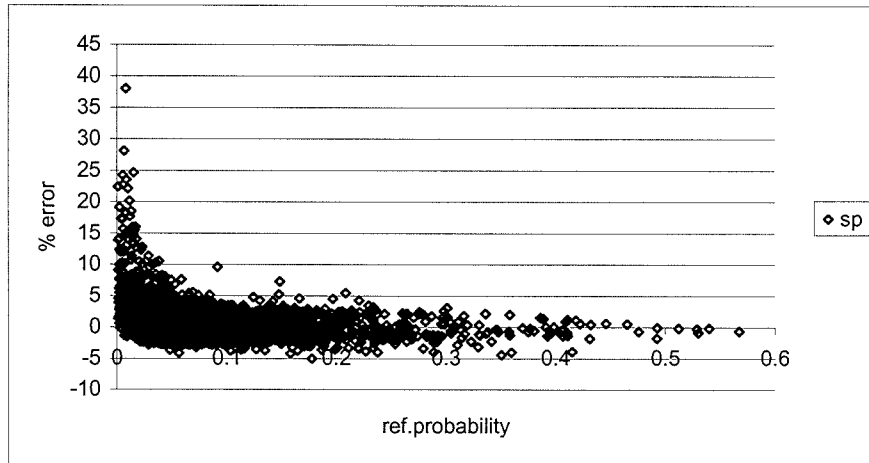


Fig. 3.40 – Percentage errors of the separated split approximation on the 12 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

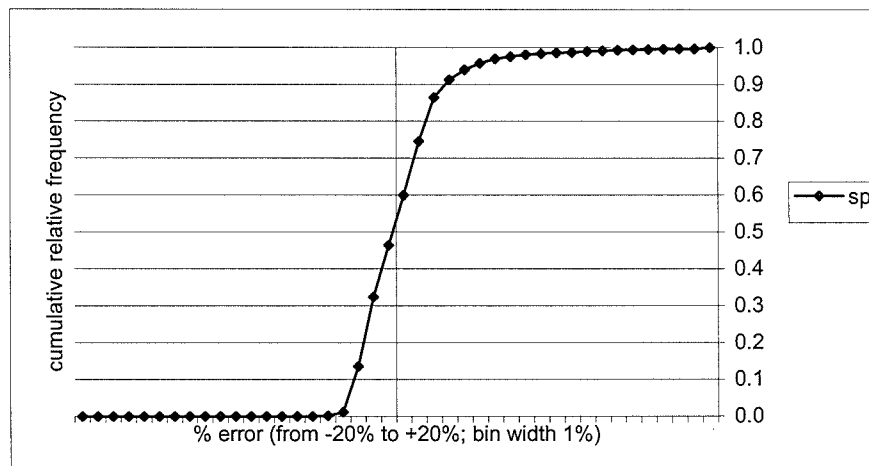


Fig. 3.41 – Cumulative relative frequencies of the percentage errors of the separated split approximation on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

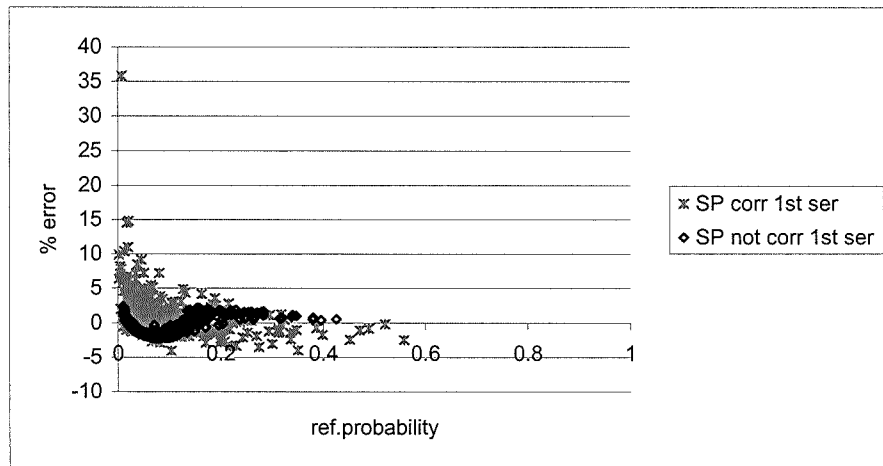


Fig. 3.42 –Percentage errors of the separated split approximation on the 9 option cases from the first series of artificial networks plotted against the reference probabilities. Only the data points for the networks with uncorrelated path (SP not corr) and correlated paths (SP corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

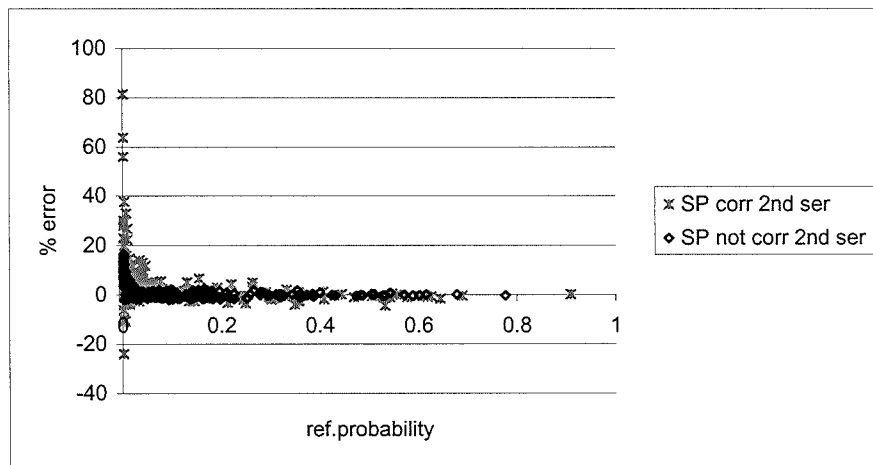


Fig. 3.43 –Percentage errors of the separated split approximation on the 9 option cases from the second series of artificial networks plotted against the reference probabilities. Only the data points for the networks with uncorrelated path (SP not corr) and correlated paths (SP corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

In the Langdon approximation case the choice probabilities are obtained together so they sum to one and do not need normalisation.

The results presented so far have been obtained by processing the variates in the optimal order suggested by Langdon. Figures 3.44 and 3.45 refer to the networks with 9 choice options and compare the results obtained with the optimal processing order with those obtained by processing the variates in a non-optimised order (marked SP (n opt) in the figures), an instance of the many possible random processing orders. The envelopes of the errors in this case are not too different but the optimal processing order gives smaller percentage errors. This has been confirmed also looking at data resulting from other non-optimal processing orders.

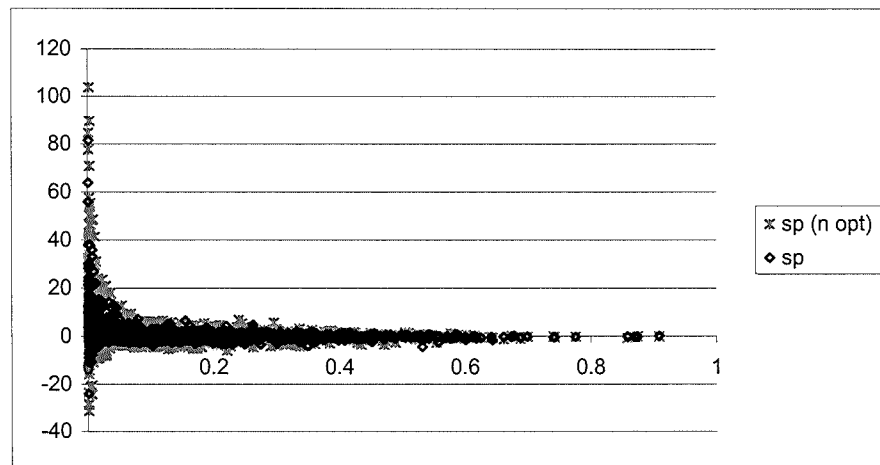


Fig. 3.44 – Percentage errors of the separated split approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

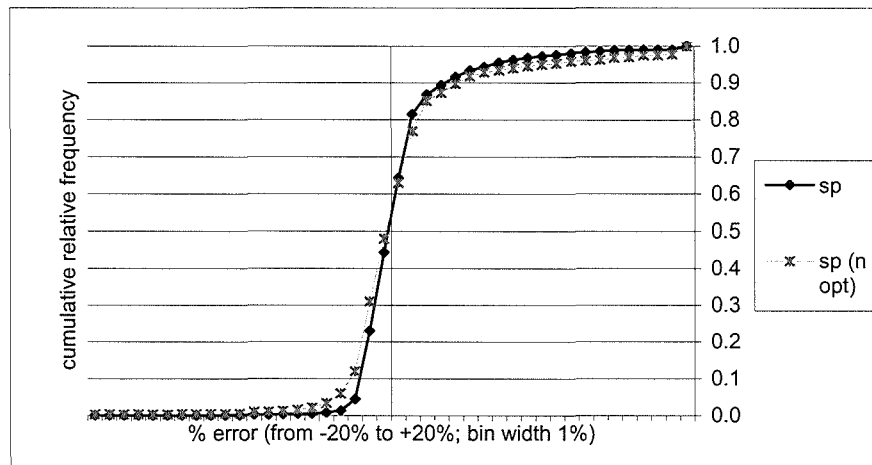


Fig. 3.45 - Cumulative relative frequencies of the percentage errors of the separated split approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

3.6.4.5 Accuracy of the FOMN Family of Approximations

The results for the crude and generalised FOMN methods are presented together in this section (the tests for the improved FOMN approximation have not been carried out). They confirm the ranking of the approximations found in the literature (Tang and Melchers, 1987): the crude FOMN method is less precise than the generalised FOMN, which gives very accurate results.

The results presented in the first part of this section have been obtained by writing the program for these approximations as described in Tang and Melchers (1987) except for the use of a bivariate Normal routine to solve the eventual bivariate conditional marginal integral in the crude FOMN method as in the generalised FOMN method. In particular, no optimal processing order for the variates entering the MVN integral calculation is followed: the program simply processes them in the order they are coded in the input file. This non-optimised processing order has been considered because the literature presenting the FOMN approximations does not seem to discuss the processing sequence. However, the use of a possible optimised order is discussed later in the section.

In tables and figures the crude FOMN method is marked CRD and the generalised FOMN method is marked GEN.

The following tables 3.21 and 3.22 present the results obtained on the examples from the second series of artificial networks. The data for the three option example are omitted as in that case all the approximations of this group, in the implementation used here, employ a routine to calculate directly the bivariate Normal integral.

Opt	Num Int	CRD	Δ CRD%	GEN	Δ GEN%
1	0.0824	0.0788	-4.44%	0.0819	-0.60%
2	0.2058	0.2112	+2.60%	0.2054	-0.19%
3	0.5127	0.5292	+3.23%	0.5132	+0.11%
4	0.0134	0.0108	-19.48%	0.0133	-0.91%
5	0.1461	0.1362	-6.77%	0.1465	+0.29%
6	0.0396	0.0339	-14.46%	0.0396	+0.08%

Tab 3.21 – Comparison of reference results with those of the crude FOMN (CRD) and generalised FOMN (GEN) approximations for the six alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Opt	Num Int	CRD	Δ CRD%	GEN	Δ GEN%
1	0.0544	0.0504	-7.36%	0.0537	-1.30%
2	0.0285	0.0277	-3.03%	0.0282	-1.25%
3	0.3246	0.3508	+8.08%	0.3213	-1.01%
4	0.1582	0.1623	+2.59%	0.1575	-0.40%
5	0.0950	0.0932	-1.84%	0.0954	+0.45%
6	0.1029	0.0987	-4.04%	0.1040	+1.08%
7	0.0146	0.0120	-17.76%	0.0148	+1.40%
8	0.1032	0.0967	-6.27%	0.1045	+1.28%
9	0.1186	0.1082	-8.76%	0.1205	+1.68%

Tab 3.22 – Comparison of reference results with those of the crude FOMN (CRD) and generalised FOMN (GEN) approximations for the nine alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

The crude FOMN method shows moderate inaccuracies both in the six and in the nine option case. In the six option case the largest percentage errors refer to two

options with the smallest actual probabilities whilst in the nine option case there are noticeable percentage errors also for options, like 3 and 9, with larger actual probabilities. The generalised FOMN method gives very precise results on both examples with the worst errors just over 1% in the nine option case.

The difference in precision and especially the very good results of the generalised FOMN approximation are evident also from the following figures which report a selection of the aggregate results obtained on the two series of artificial networks.

Examining envelopes of the errors as those reported in figures 3.46, 3.49 and 3.51 it is clear that the crude FOMN approximation applied to the MNP problem tends to underestimate the small choice probabilities and overestimate larger ones. This tendency is amplified in magnitude as the number of options, and therefore the number of times the approximation is applied, increases as can be seen by comparing those figures.

The graphs depicting the generalised FOMN results show a much smaller magnitude of the percentage errors, that confirms the very limited extent of possible errors suggested by the tables on the two example cases. This is evident from the graphs reporting the percentage errors against the reference probability (figs. 3.47, 3.50, 3.51) and also from those depicting the cumulative relative frequency of the percentage errors (figs. 3.48, 3.52).

The envelope of the errors is, generally, wedge shaped, presenting both underestimation and overestimation of low choice probabilities. The width of such wedge shaped envelope, and thus the magnitude of the possible errors, becomes larger as the number of choice options increases (compare figs. 3.47, 3.50, 3.51). However, the errors remain of much smaller magnitude than in the crude FOMN case, as shown in figs. 3.51 and 3.52.

A difference of the FOMN methods from the other approximations is that the width of the envelope of the errors for a given number of options does not vary significantly with the overlapping of the options. However, with increasing overlapping the errors points in the graphs are more scattered. Examples of

comparison of the inaccuracies for different overlapping of the paths are given in figure 3.49 for the crude FOMN case and in figure 3.50 for the generalised FOMN case. In the generalised FOMN case the wedge shaped envelope shown in cases of correlated paths changes to a trend of underestimation of low probabilities and overestimation of larger ones when there is no correlation between the paths.

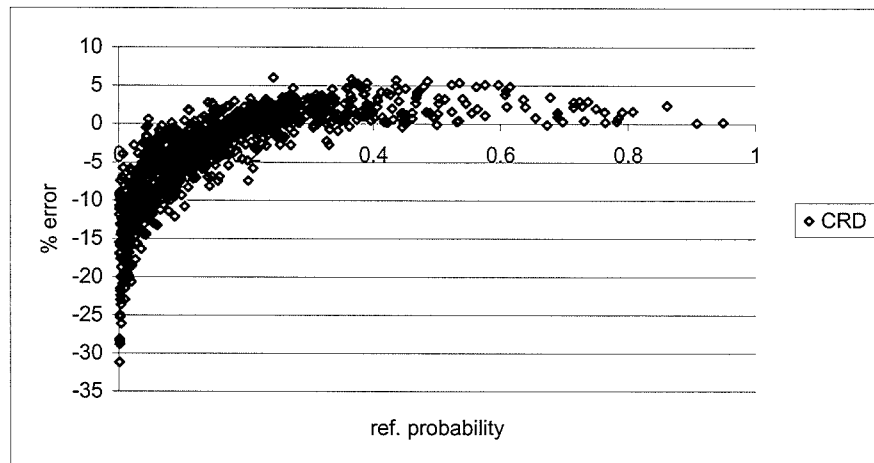


Fig. 3.46 – Percentage errors of the crude FOMN approximation on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

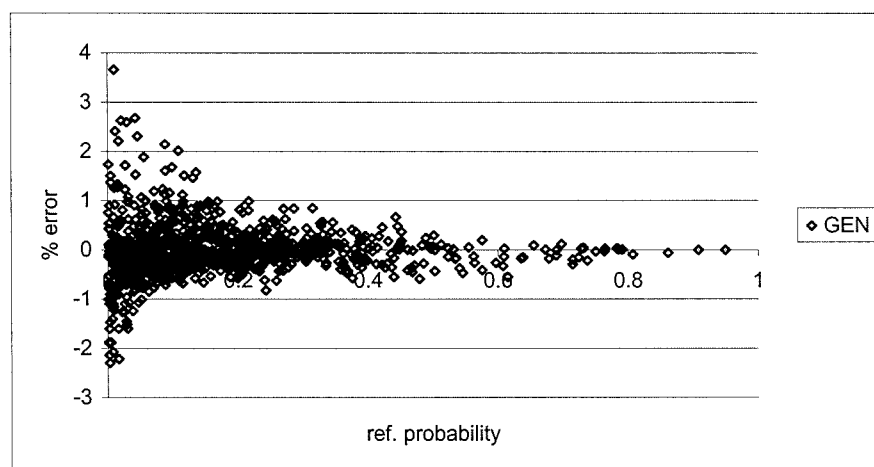


Fig. 3.47 – Percentage errors of the generalised FOMN approximation on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

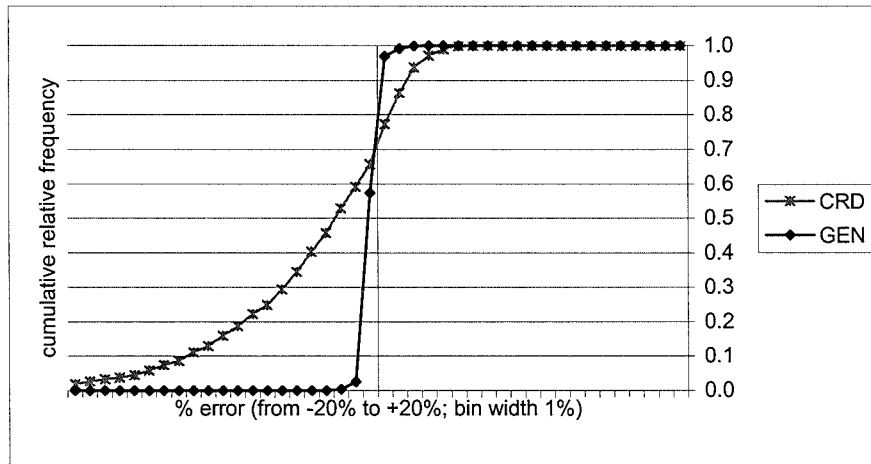


Fig. 3.48 – Cumulative relative frequencies of the percentage errors of the crude and generalised FOMN approximation on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

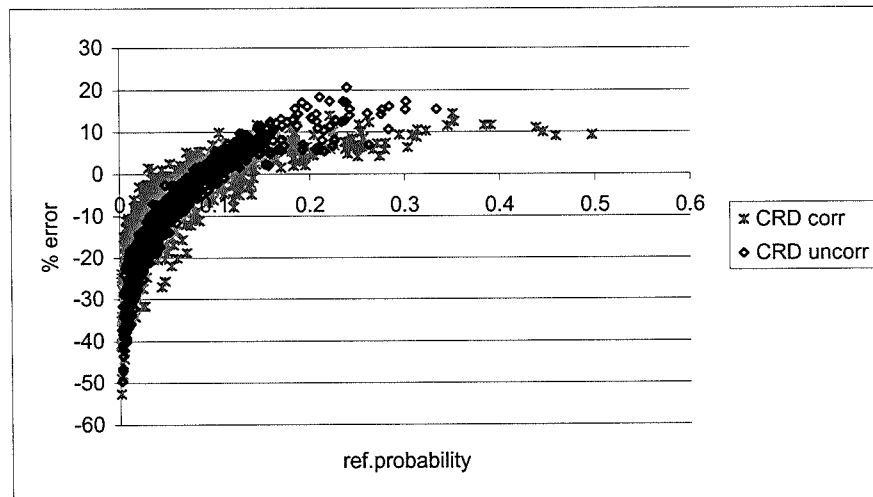


Fig. 3.49 – Percentage errors of the crude FOMN on the 15 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data point for the networks with uncorrelated path (CRD uncorr) and correlated paths (CRD corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

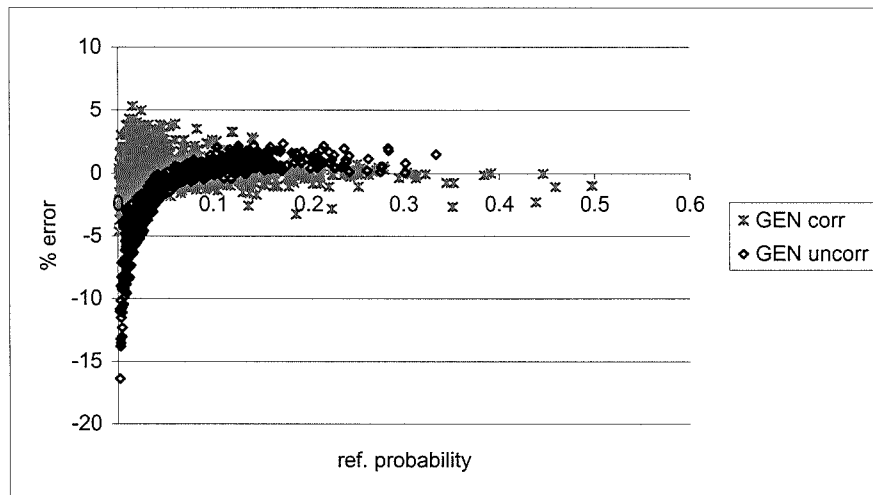


Fig. 3.50 – Percentage errors of the generalised FOMN on the 15 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data point for the networks with uncorrelated path (GEN uncorr) and correlated paths (GEN corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

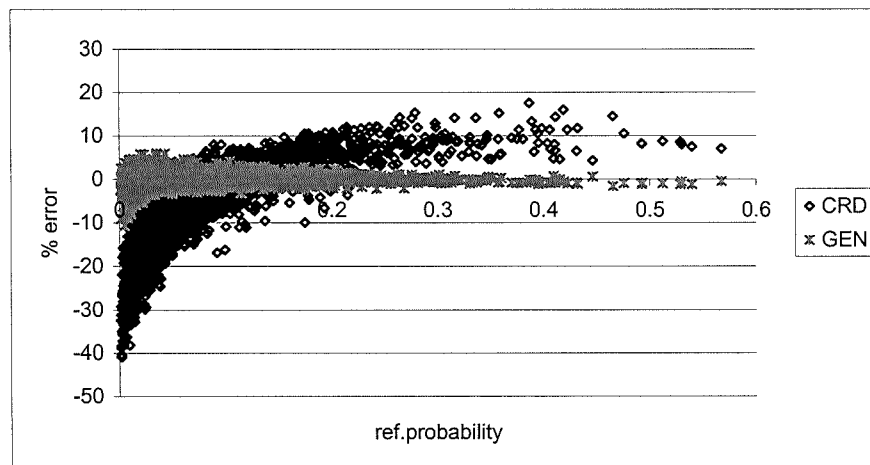


Fig. 3.51 - Percentage errors of the crude and generalised FOMN approximation on the 12 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

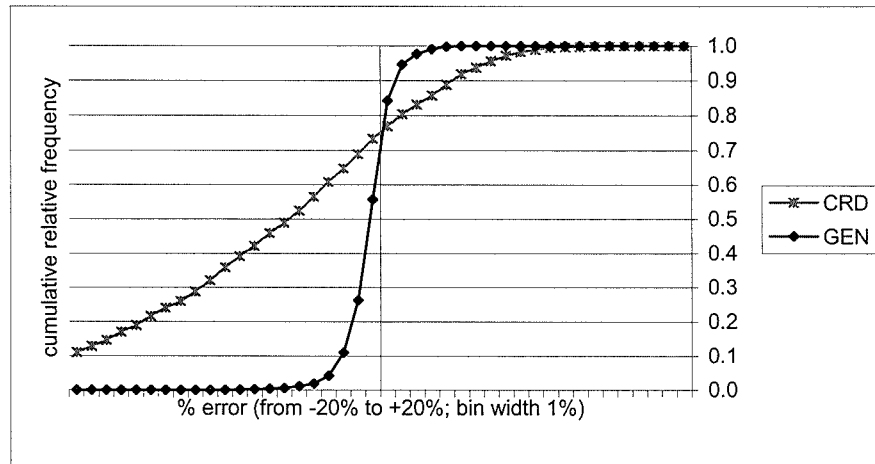


Fig. 3.52 – Cumulative relative frequencies of the percentage errors of the crude and generalised FOMN approximation on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

The probabilities of each of the options are calculated separately in the methods of the FOMN group and are thus normalised to their sum. Tables 3.23 and 3.24 detail mean, standard deviation, maximum and minimum difference from one of the sum of the probabilities obtained in the experiments organised by number of choices respectively for the crude FOMN and the generalised FOMN approximations.

The numbers reported in the 3 option cases are due to the fact that probabilities are obtained directly from a bivariate Normal routine and sum precisely to one.

It is evident that the normalisation has a large importance on the crude FOMN results and that it increases with the number of options in the choice set. The normalisation effect, as shown by the maximum and minimum values, is always that of amplifying the magnitude of the calculated probabilities, as the sum of the original probabilities is always smaller than one. This general tendency of the approximation to underestimate the correct results (before normalisation) has been observed also in Tang and Melchers (1987) and Pandey (1998a,b). However, in structural reliability the tendency towards underestimation is a safe behaviour as it results in estimating conservatively the reliability of a structure.

The normalisation effect in the generalised FOMN method is, instead, of very small importance, although of the same order of magnitude of the inaccuracies shown by the approximation and it increases with the number of options concurrently with the decrease in accuracy of the approximation. In this case, rather than underestimation of the sum of the probabilities, there is most often overestimation and the final probabilities are slightly smaller than the original ones.

Options	Mean	st.dev.	Max	Min
	CRD		CRD	CRD
3	1.0000	0.0000	1.0000	1.0000
6	0.9113	0.0328	0.9973	0.8552
9	0.8448	0.0544	0.9961	0.7474
12	0.7463	0.0420	0.8594	0.6655
15	0.6942	0.0468	0.8282	0.5994

Table 3.23 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the crude FOMN (CRD) approximation on the first and second series of artificial networks.

Options	Mean	st.dev.	Max	Min
	GEN		GEN	GEN
3	1.0000	0.0000	1.0000	1.0000
6	1.0070	0.0048	1.0211	0.9999
9	1.0184	0.0095	1.0458	1.0001
12	1.0371	0.0138	1.0814	1.0104
15	1.0510	0.0202	1.1015	1.0046

Table 3.24 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the generalised FOMN (GEN) approximation on the first and second series of artificial networks.

Although the results reported above, as well as some of those reported on the next pages, have been obtained with no particular processing order of the variates, experiments to establish a possible preferred order were carried out. The literature on

this family of approximations seems to have no discussion of a possible preferential processing order. A possible optimised order has been obtained by analogy with the Mendell-Elston and the Taylor series approximations where more precise results are obtained if the calculations are started with the variate with the smallest marginal integral. Such order gives improved results that are particularly interesting for the generalised FOMN approximation and that suggest to propose it as an optimal processing order for the approximations of this family. The opposite order, tested to check the importance of the ordering criterion gave much larger errors, not better than those from the random order examined above.

The following tables and pictures are a selection of the comparisons of the results obtained with the heuristic optimal order suggested and with the original non-optimal order. The results obtained with the variate reordering are marked *crd (opt)* and *gen (opt)*, respectively for the crude and generalised approximations. The results obtained with the non-optimal ordering are marked as before.

Option	Num.Int.	Crd $\Delta\%$	Crd (opt) $\Delta\%$	Gen $\Delta\%$	Gen (opt) $\Delta\%$
1	0.0824	-4.44%	-4.37%	-0.60%	-0.18%
2	0.2058	+2.60%	+1.12%	-0.19%	-0.02%
3	0.5127	+3.23%	+1.69%	+0.11%	+0.09%
4	0.0134	-19.48%	-10.26%	-0.91%	-0.50%
5	0.1461	-6.77%	-2.16%	+0.29%	-0.06%
6	0.0396	-14.46%	-7.10%	+0.08%	-0.25%

Tab 3.25 – Comparison of reference results with those of the crude and generalised FOMN approximations obtained considering the alternative calculation orders described in the text for the six alternative example defined in section 3.6.3. $\Delta\%$ obtained from the non-approximated values of the results.

Option	Num.Int.	Crd $\Delta\%$	Crd (opt) $\Delta\%$	Gen $\Delta\%$	Gen (opt) $\Delta\%$
1	0.0544	-7.36%	-4.40%	-1.30%	-0.06%
2	0.0285	-3.03%	-4.51%	-1.25%	-0.15%
3	0.3246	+8.08%	+2.84%	-1.01%	+0.34%
4	0.1582	+2.59%	+1.63%	-0.40%	-0.03%
5	0.0950	-1.84%	-0.64%	+0.45%	-0.25%
6	0.1029	-4.04%	-1.49%	+1.08%	-0.14%
7	0.0146	-17.76%	-10.14%	+1.40%	-0.76%
8	0.1032	-6.27%	-1.09%	+1.28%	-0.22%
9	0.1186	-8.76%	-2.78%	+1.68%	-0.15%

Tab 3.26 – Comparison of reference results with those of the crude and generalised FOMN approximations obtained considering the alternative calculation orders described in the text for the nine alternative example defined in section 3.6.3. $\Delta\%$ obtained from the non-approximated values of the results.

Tables 3.25 and 3.26 re-examine the 6 and 9 alternative examples presenting the percentage errors with the optimal reordering and without. In most cases there is a reduction of the absolute importance percentage error. However, in some cases there is a change in sign of such inaccuracies that goes together with a reduction in absolute importance (see *e.g.* the generalised results for options 5 to 9 in table 3.26) or in one of the cases (option 6 in table 3.25) with an increase. Looking at more than the data reported in the tables, these changes are partly due to different accuracy of the non-normalised results and partly to the normalisation of the probabilities to their sum.

Looking at the overall trend of percentage errors and at their relative frequency (figs. 3.53 to 3.56) it is evident that the reordering brings about a reduction in the likely spread of percentage errors. Whilst each approximation with reordering of the variates performs better than its corresponding version without reordering, the generalised FOMN method continues to give much smaller errors than the crude FOMN. Also, the spread of percentage errors increases with the number of options as is the case without reordering.

The change of processing order did not change the shape of the envelope of the crude FOMN errors. In the generalised FOMN errors the trend is no longer wedge shaped but shows underestimation of low choice probabilities and some overestimation of larger ones, although of limited magnitude.

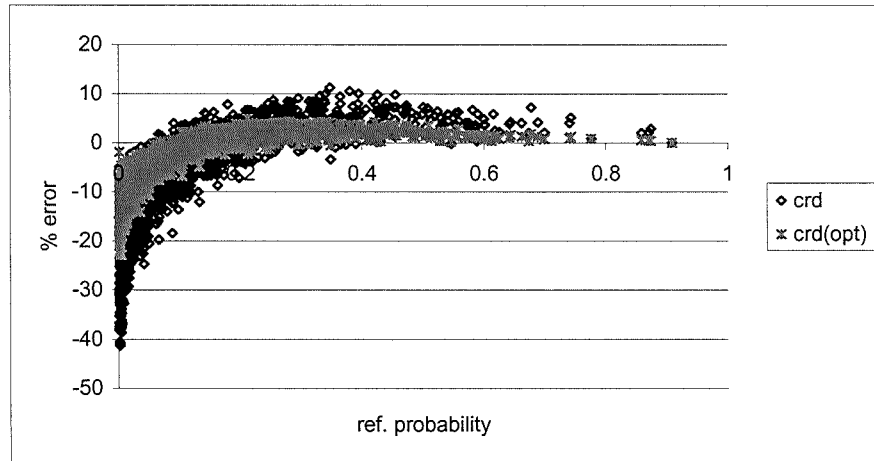


Fig. 3.53 – Percentage errors of the crude FOMN approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

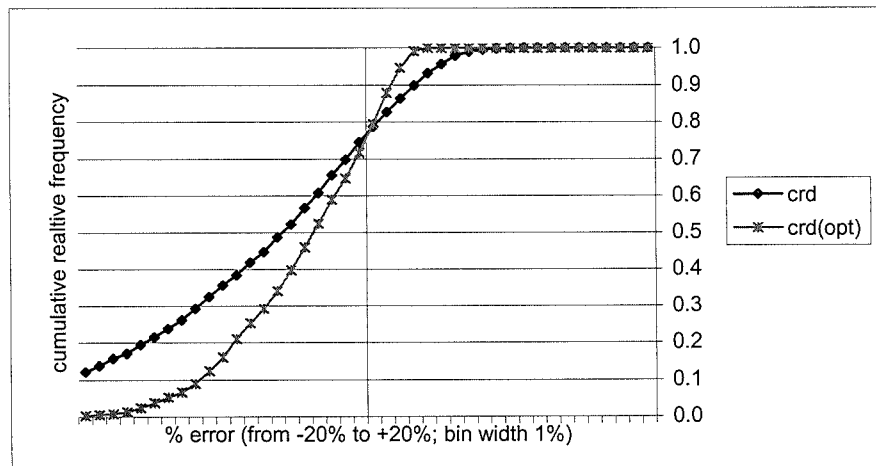


Fig. 3.54 – Cumulative relative frequencies of the percentage errors of the crude FOMN approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

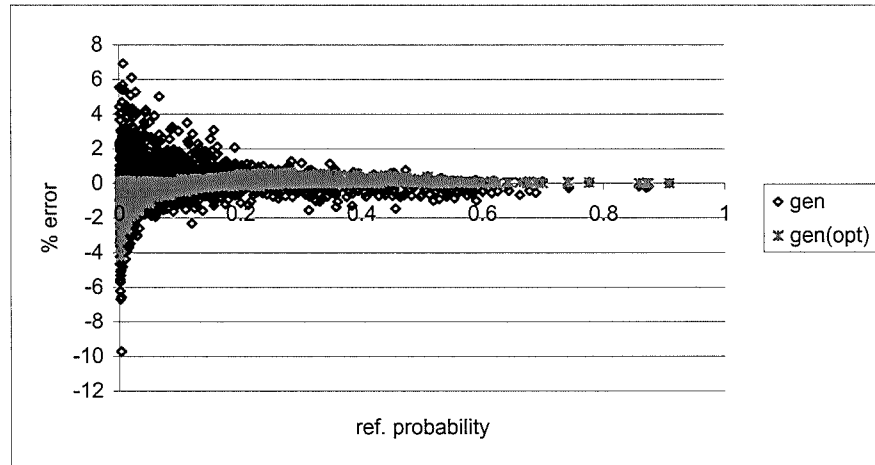


Fig. 3.55 – Percentage errors of the generalised FOMN approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

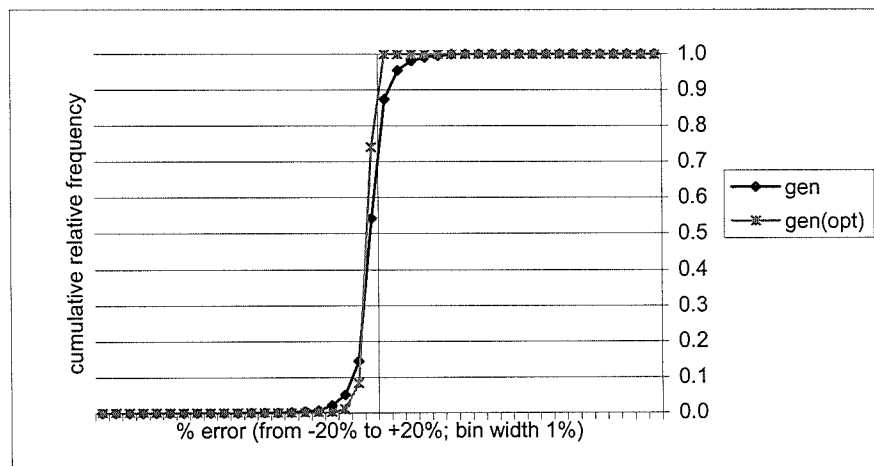


Fig. 3.56 – Cumulative relative frequencies of the percentage errors of the generalised FOMN approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

When different correlations of the paths are considered, the crude FOMN shows the same trend of underestimation and overestimation present when data from all test networks are considered. An increase in the correlation increases the dispersion of the percentage errors but not substantially the upper and lower limit of the arch trend they make, as shown in fig. 3.57. In the generalised FOMN case, the trend of the percentage errors is again almost unchanged by the variation of overlapping amongst

the paths, but when paths are more correlated the data points are more scattered as in figure 3.58.

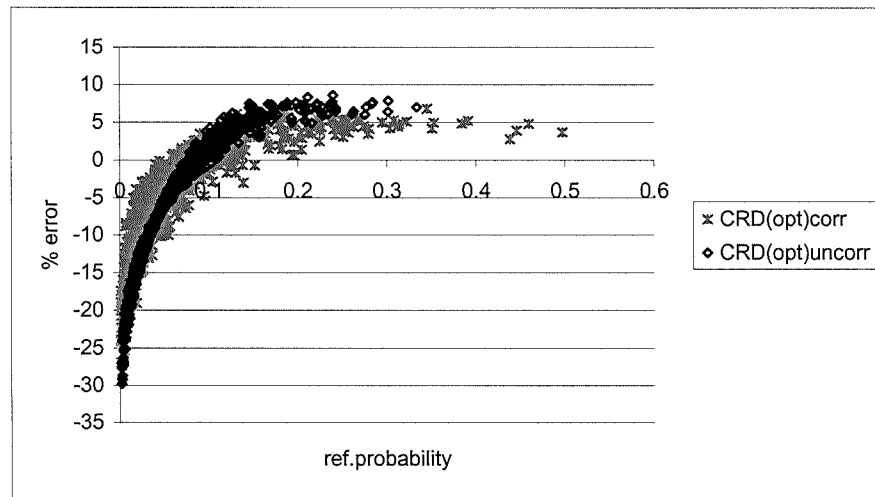


Fig. 3.57 – Percentage errors of the crude FOMN with variate reordering on the 15 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data point for the networks with uncorrelated path (CRD(opt)uncorr) and correlated paths (CRD(opt)corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

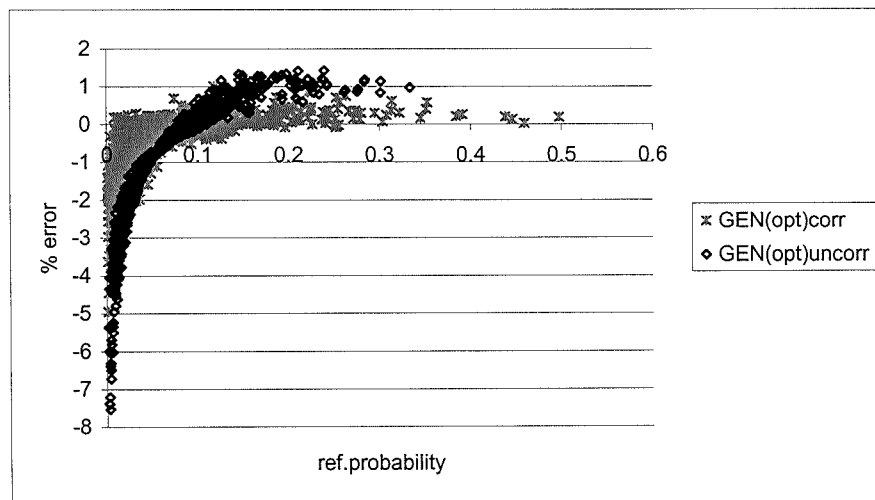


Fig. 3.58 – Percentage errors of the generalised FOMN with variate reordering on the 15 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data point for the networks with uncorrelated path (GEN(opt)uncorr) and correlated paths (GEN(opt)corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

Tables 3.27 and 3.28 detail mean, standard deviation, maximum and minimum difference from one of the sum of the probabilities obtained in the experiments with the approximations using the optimal order organised by number of choices respectively for the crude FOMN and the generalised FOMN approximations.

As without variate reordering, in the crude FOMN case, the normalisation has always the effect of increasing the original choice probabilities. In other words the original approximation always underestimates the sum of the probabilities. However, comparing table 3.27 with table 3.23, the importance of such effect is reduced by the variate reordering. The lesser accuracy when the choice set is larger is shown also by the reduction of the accuracy in the estimation of the sum of the probabilities. In the generalised FOMN case the effect of the normalisation is, instead, to reduce the original choice probabilities, but its importance is even smaller than in the case without optimal reordering.

Options	Mean crd(opt)	St.dev. crd(opt)	Max crd(opt)	Min crd(opt)
3	1.0000	0.0000	1.0000	1.0000
6	0.9301	0.0299	0.9991	0.8720
9	0.8877	0.0498	0.9988	0.7800
12	0.8084	0.0388	0.9110	0.7113
15	0.7729	0.0467	0.8994	0.6477

Table 3.27 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the optimised order crude FOMN (CRDopt) approximation on the first and second series of artificial networks.

Options	Mean GENopt	st.dev. GENopt	Max GENopt	Min GENopt
3	1.0000	0.0000	1.0000	1.0000
6	1.0017	0.0013	1.0066	1.0000
9	1.0036	0.0024	1.0108	1.0000
12	1.0074	0.0028	1.0168	1.0015
15	1.0092	0.0043	1.0270	1.0012

Table 3.28 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the optimised order generalised FOMN (GENopt) approximation on the first and second series of artificial networks.

3.6.4.6 Accuracy of the Taylor Series Approximations

The Taylor series approximation has been tested in the literature in its original application to solve MVN integrals rather than MNP problems by Olson and Weissfeld (1991) and Cox and Wermuth (1991), who proposed it. Most of the tests carried out by Olson and Weissfeld are on the solution of trivariate Normal integrals and compare the first order Taylor approximation with the Mendell and Elston one (for which they did not mention using a preferred calculation order). Their results show little difference from the Mendell-Elston ones and, in general, they found that the Taylor approximation gives accurate results but its accuracy diminishes when the correlation amongst the variates increases and when the integrals have a large actual value. Olson and Weissfeld also tested the first order Taylor approximation in cases with up to 20 variates (considered equicorrelated and with the same integration limit). The results they reported show that the approximation becomes noticeably less accurate with dimensions of the integral larger than 7. Cox and Wermuth tested the first and second order approximation for the solution of bivariate Normal integrals and the first order only for the solution of trivariate Normal integrals obtaining in all cases results with good precision as long as their optimal calculation order was followed.

The Taylor approximation has been tested here considering its first and second order, marked respectively T1 and T2 in the following tables and graphs. The calculations have been carried out by considering the variates in the order suggested by Olson and Weissfeld (1991) and Cox and Wermuth (1991), that, as mentioned in 3.5.6 consists of including in the calculations at each stage the variate giving the smallest marginal integral first.

The results obtained are in accordance with those on the limited set of cases with large dimension explored by Olson and Weissfeld and show that, although the precision is good for a small number of variates, it deteriorates quickly for larger dimensions of the problem.

A first impression on the precision of this method can be obtained observing the 3 cases taken from the second series of artificial test networks reported respectively in tables 3.29, 3.30 and 3.31.

The results are rather good and, in most cases, the second order approximation gives errors of lower percentage importance, although in the cases of six and nine alternatives they are not too dissimilar. The tendency of the percentage errors to increase with the number of options is not evident from these examples although there are higher percentage errors in the nine alternative example for option 7, whose actual choice probability is of the order of one percent. In two cases of the same example, the percentage importance of the second order approximation error is higher than that of the first order one: this is an effect of the normalisation to the sum of the raw results that also reduces the percentage importance of the errors for each approximation.

Opt	Num Int	T1	$\Delta T1\%$	T2	$\Delta T2\%$
1	0.3117	0.3110	-0.19%	0.3117	+0.01%
2	0.0365	0.0360	-1.30%	0.0366	+0.20%
3	0.6519	0.6529	+0.17%	0.6517	-0.07%

Tab 3.29 – Comparison of reference results with those of the Taylor series approximations for the three alternatives example defined in section 3.6.3. $\Delta\%$ obtained from the non-approximated values of the results.

Opt	Num Int	T1	$\Delta T1\%$	T2	$\Delta T2\%$
1	0.0824	0.0806	-2.18%	0.0808	-1.96%
2	0.2058	0.2035	-1.11%	0.2049	-0.45%
3	0.5127	0.5149	+0.45%	0.5188	+1.20%
4	0.0134	0.0129	-3.49%	0.0130	-3.42%
5	0.1461	0.1488	+1.86%	0.1435	-1.75%
6	0.0396	0.0392	-1.10%	0.0390	-1.43%

Tab 3.30 – Comparison of reference results with those of the Taylor series approximations for the six alternatives example defined in section 3.6.3. $\Delta\%$ obtained from the non-approximated values of the results.

Opt	Num Int	T1	$\Delta T1\%$	T2	$\Delta T2\%$
1	0.0544	0.0552	+1.58%	0.0542	-0.33%
2	0.0285	0.0283	-0.79%	0.0282	-1.35%
3	0.3246	0.3298	+1.62%	0.3329	+2.56%
4	0.1582	0.1583	0.09%	0.1579	-0.16%
5	0.0950	0.0941	-0.91%	0.0905	-4.75%
6	0.1029	0.1029	-0.01%	0.1044	+1.46%
7	0.0146	0.0131	-9.95%	0.0133	-8.98%
8	0.1032	0.1011	-2.08%	0.1018	-1.37%
9	0.1186	0.1171	-1.25%	0.1169	-1.43%

Tab 3.31 – Comparison of reference results with those of the Taylor series approximations for the nine alternatives example defined in section 3.6.3. $\Delta\%$ obtained from the non-approximated values of the results.

Figures 3.59, 3.60, 3.61 and 3.62 report two examples of aggregate comparison of the accuracy between the two orders of approximation. The inspection of the dispersion and of the envelope of the percentage errors plotted against the reference probabilities shows that the inaccuracies given by the two orders of approximation are very similar. In fact, although the second order approximation generally gives less disperse percentage errors, there are only minor differences between the inaccuracies especially as the number of alternatives increases. Both approximations show a decreasing accuracy for larger number of alternatives and this is mirrored by

the data about the normalisation of the results (see table 3.32), whose effect becomes of larger importance with more alternatives.

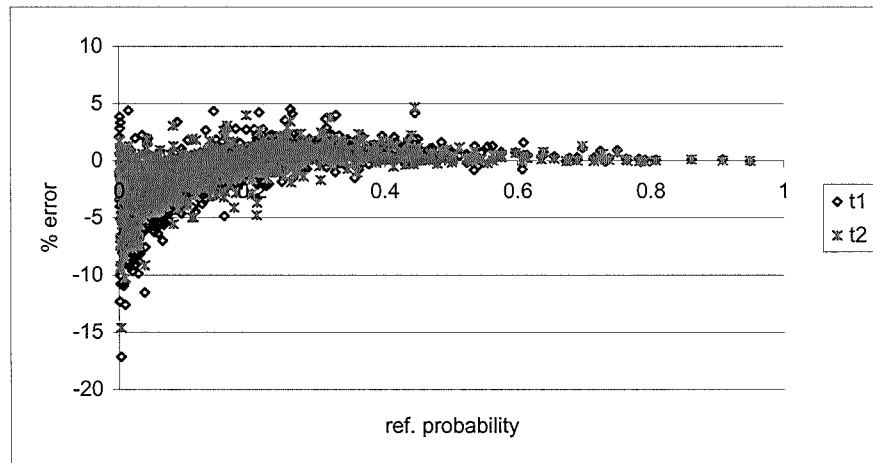


Fig. 3.59 – Percentage errors of the first and second order Taylor approximations on the six option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

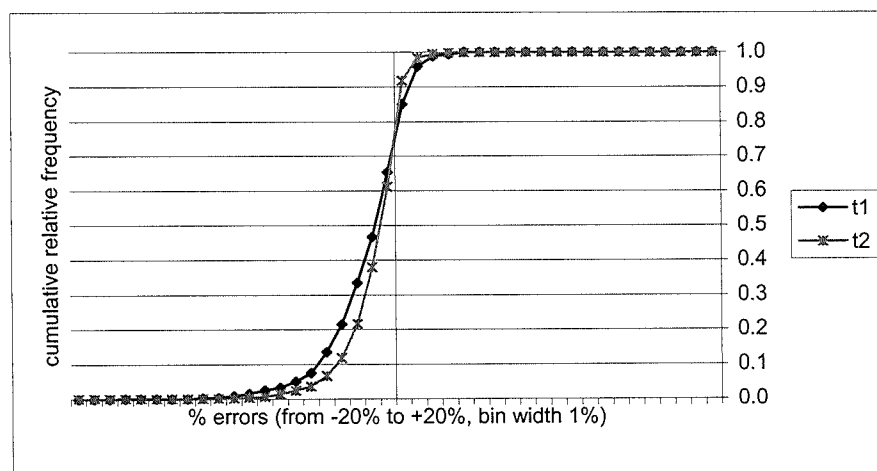


Fig. 3.60 – Cumulative relative frequencies of the percentage errors of the first and second order Taylor approximations on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

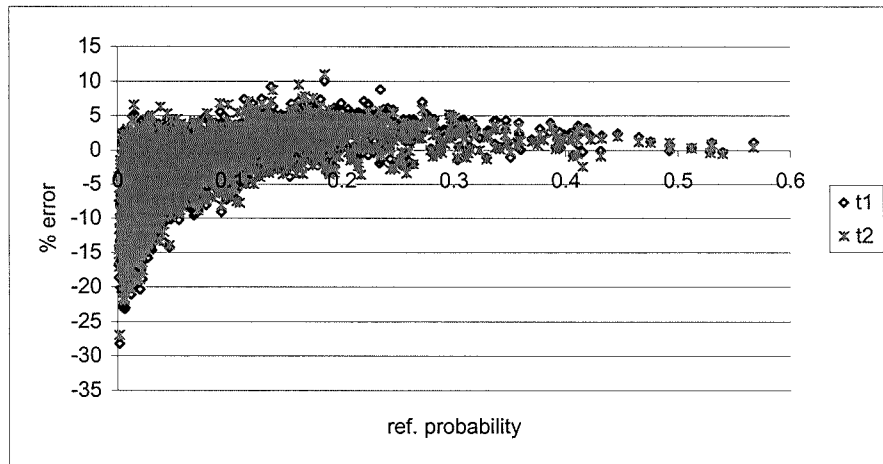


Fig. 3.61 – Percentage errors of the first and second order Taylor approximations on the 12 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

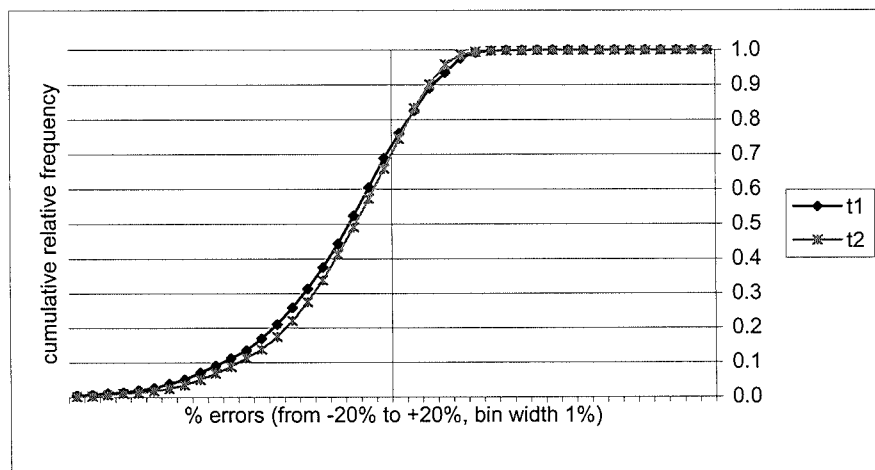


Fig. 3.62 - Cumulative relative frequencies of the percentage errors of the first and second order Taylor approximations on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

The figures above show also that the trends of overestimation and underestimation are similar for both approximation orders. Except for the case with only 3 options, there is mainly a tendency to underestimate small probabilities and overestimate larger ones. In the three option case this happens for the first order approximation but the second order one tends to overestimate lower choice probabilities.

Examining the aggregate data for different amount of path overlapping, it is clear that the dispersion of the percentage errors increases with the correlation between the paths (see figure 3.63 and 3.64): the plots have an arch shaped trend for non-correlated paths which changes to a wedge shaped one (with prevailing underestimation of low choice probabilities) for more correlated options. Thus the underestimation-overestimation trends are more defined for less correlated options, whose errors are less disperse. The percentage importance of the overestimation of the large probabilities remains similar with increasing correlation of the paths whilst lower ones have larger percentage errors.

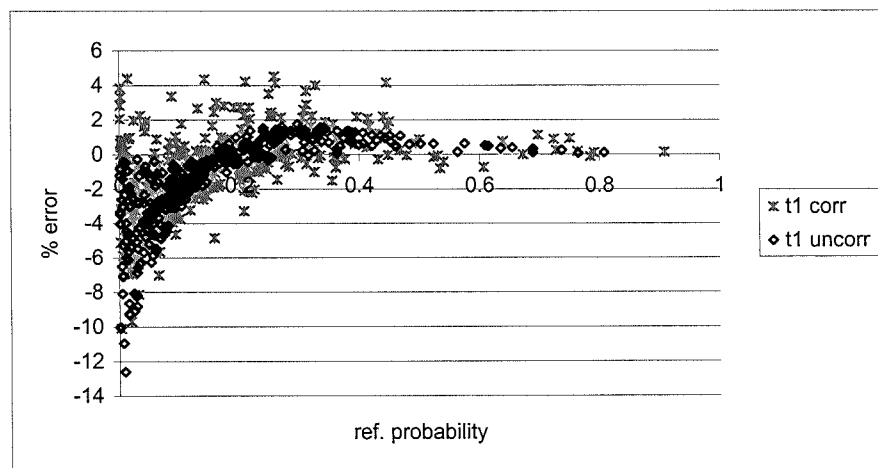


Fig. 3.63 – Percentage errors of the first order Taylor approximation on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data points for the networks with uncorrelated path (t1 uncorr) and correlated paths (t1 corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

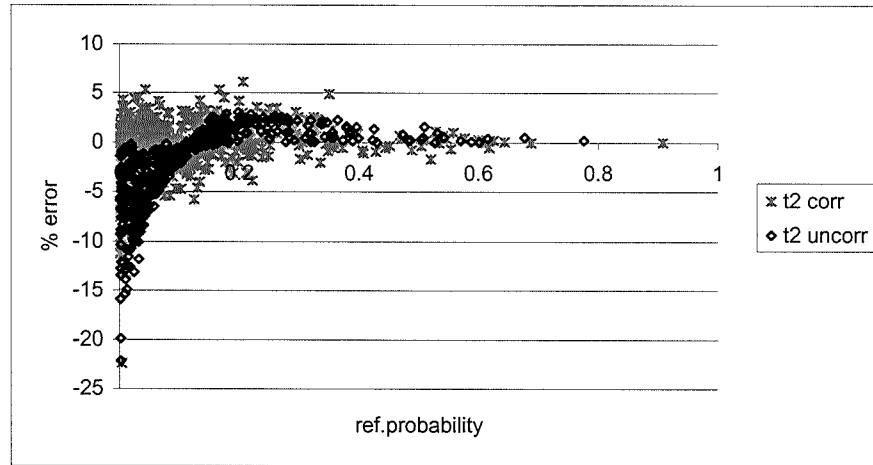


Fig. 3.64 - Percentage errors of the second order Taylor approximation on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data points for the networks with uncorrelated path (t2 uncorr) and correlated paths (t2 corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

The results on the normalisation of the choice probabilities mirror those on the accuracy of the percentage errors of the probabilities: as the number of options, and therefore of MVN integrals to solve increases, so does the difference of their sum from one. Table 3.32 reports mean, standard deviation, maximum and minimum difference from one of the sum of the probabilities obtained in the experiments organised by number of choices.

Options	Mean T1	Mean T2	st.dev. T1	st.dev. T2	Max T1	Max T2	Min T1	Min T2
3	1.0129	0.9992	0.0068	0.0009	1.0289	1.0008	1.0002	0.9964
6	1.0405	1.0132	0.0195	0.0168	1.0910	1.0637	1.0023	0.9798
9	1.0543	1.0250	0.0371	0.0359	1.1381	1.1133	0.9506	0.9228
12	1.0785	1.0491	0.0701	0.0703	1.1756	1.1489	0.8999	0.8718
15	1.0818	1.0537	0.0919	0.0930	1.2539	1.2052	0.8339	0.8086

Table 3.32 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the first (T1) and second (T2) order Taylor approximation on the first and second series of artificial networks.

To verify the importance of using the optimal calculation sequence suggested by Olson and Weissfeld (1991) and Cox and Wermuth (1991), the tests on the artificial networks were also carried out using as calculation sequence the order in which the options are coded as input data to the computer program, an instance of the many possible random orders. Figures 3.65, 3.66, 3.67 and 3.68 are examples of comparisons of percentage errors between cases with such order and with optimised order. The reduction of the percentage errors with the use of the optimal order suggested in the literature is clearly significant, and is particularly important in the second series of networks. Moreover, comparing plots similar to those reported for both approximation orders, it is clear that using a correct calculation sequence (designed to reduce the importance of the disregarded Taylor series terms) is more important than refining the Taylor series including higher order terms for obtaining more accurate results.

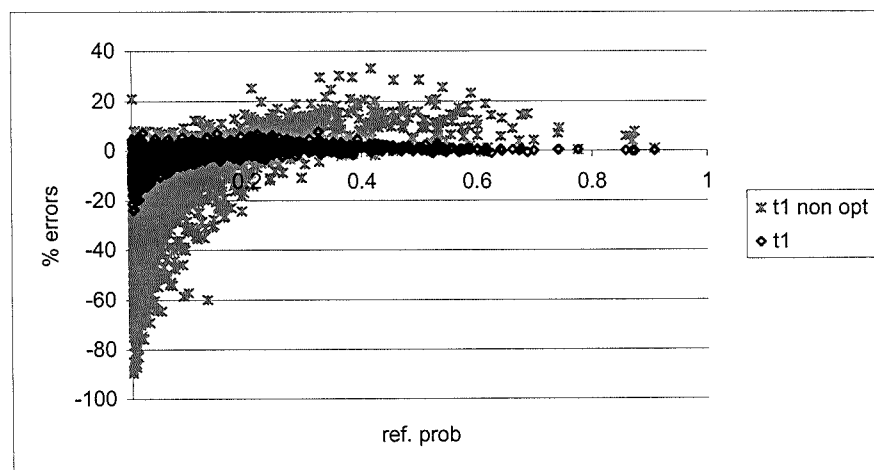


Fig. 3.65 – Percentage errors of the first order Taylor approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

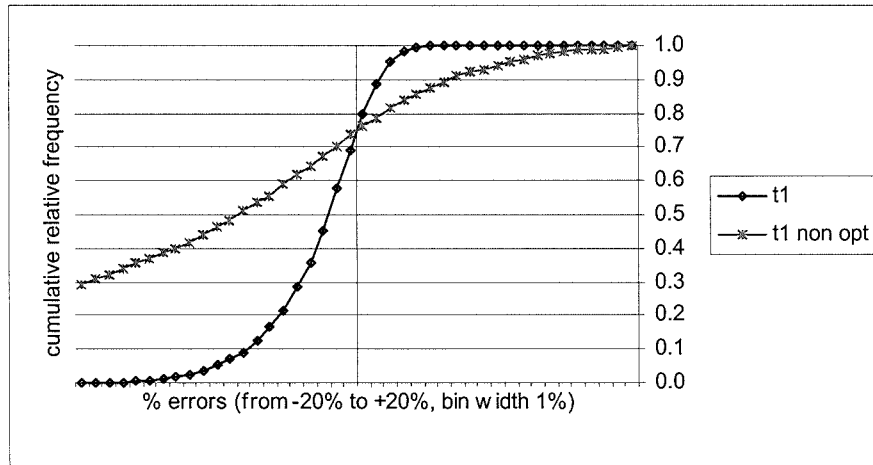


Fig. 3.66 - Cumulative relative frequencies of the percentage errors of the first order Taylor approximation (with optimised and non-optimised calculation order) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

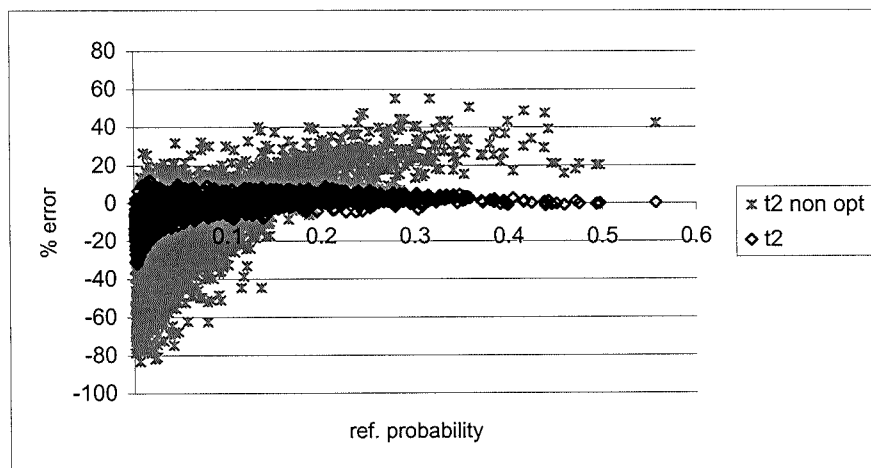


Fig. 3.67 – Percentage errors of the second order Taylor approximation (with optimised and non-optimised calculation order) on the 15 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

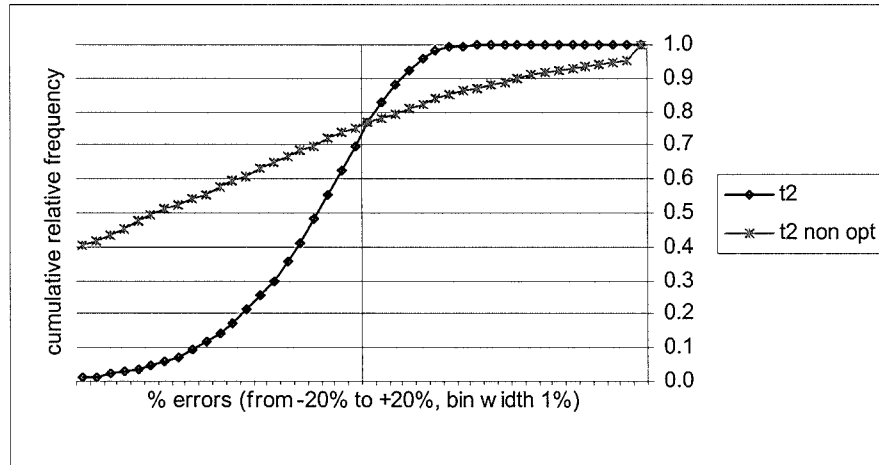


Fig. 3.68 - Cumulative relative frequencies of the percentage errors of the second order Taylor approximation (with optimised and non-optimised calculation order) on the 15 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

3.6.4.7 Accuracy of the Solow-Joe Approximation

Tests on the accuracy of this approximation are reported by Joe (1995), who extended to the case relevant in choice modelling the approximation originally devised by Solow (1990). In its original application for solving MVN integrals, this approximation resulted in more precise estimates for lower correlation of the variates. Joe (1995) gave no percentage error indication but mentioned that the errors recorded were typically in the third decimal place and the maximum ones were in the second decimal place.

The Solow-Joe approximation has been tested here by averaging the results obtained from several calculation sequences as proposed by Joe (1995). Such results are marked SJ in the following tables and pictures. Although Joe suggested to use only a sample of calculation sequences for MVN integrals larger than 6, here all the permutations of the variates have been included for up to 9 options and a sample of 1000 has been used for the 12 and 15 option cases.

The results on the excerpt of cases from the second series of artificial networks are reported in tables 3.33 and 3.34 with the exception of the case with 3 options as this

approximation solves them by employing directly a routine for the bivariate Normal integral. The inaccuracies appear in all cases to be of moderate importance. Small choice probabilities in table 3.34 show larger percentage errors, suggesting the general trend of the errors of this method that can be noticed from the aggregate data reported later.

Option	Num. Int.	SJ	Δ SJ%
1	0.0824	0.0813	-1.38%
2	0.2058	0.2065	+0.34%
3	0.5127	0.5159	+0.63%
4	0.0134	0.0126	-6.41%
5	0.1461	0.1458	-0.16%
6	0.0396	0.0379	-4.31%

Tab 3.33 – Comparison of reference results with those of the Solow-Joe approximation with averaging (SJ) for the six alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num. Int.	SJ	Δ SJ%
1	0.0544	0.0506	-7.03%
2	0.0285	0.0265	-7.11%
3	0.3246	0.3328	2.54%
4	0.1582	0.1610	1.81%
5	0.0950	0.0955	0.51%
6	0.1029	0.1033	0.43%
7	0.0146	0.0133	-8.92%
8	0.1032	0.1022	-0.97%
9	0.1186	0.1148	-3.21%

Tab 3.34 – Comparison of reference results with those of the Solow-Joe approximation with averaging (SJ) for the nine alternatives example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

The following pictures show a sample of the aggregate graphical results obtained. To assess the importance of the averaging effect, figure 3.69 and 3.70 report also the data obtained following a single random order (the results are marked SR in the

pictures). As with the other approximations, the order resulting from the sequence in which the options have been coded in the input files has been used as an instance of a random order.

It is clear from figs. 3.69 and 3.70, that the random order gives results that are much less accurate than those obtained averaging over all the permutations. The same consideration is valid for cases of larger choice sets also when the averaging is carried out only over a large sample of the possible permutations.

As figures 3.69 and 3.71 show, this approximation tends to underestimate the low choice probabilities and in particular the very small ones. Higher probabilities are typically slightly overestimated.

Examining the envelopes of the percentage errors for different amount of overlapping and number of paths suggests that the trend of underestimation of low choice probabilities and overestimation of larger ones is always present, although more correlated options result in data points that are more scattered, as shown in figure 3.73.

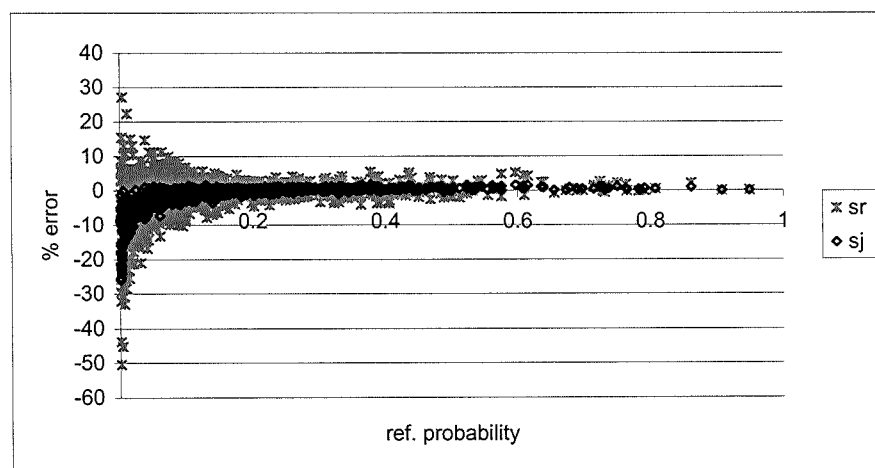


Fig. 3.69 – Percentage errors of the Solow-Joe approximation with averaged results (SJ) and with random order (SR) on the six option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

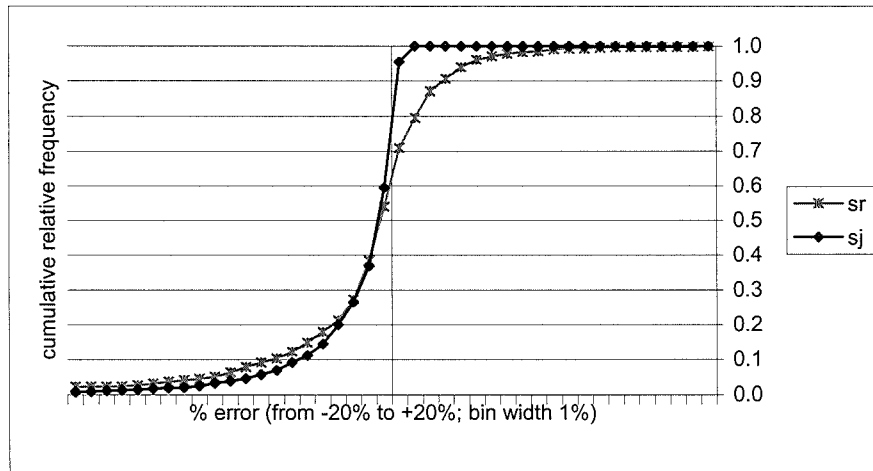


Fig. 3.70 – Cumulative relative frequencies of the percentage errors of the Solow-Joe approximation with averaged results (SJ) and with random order (SR) on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

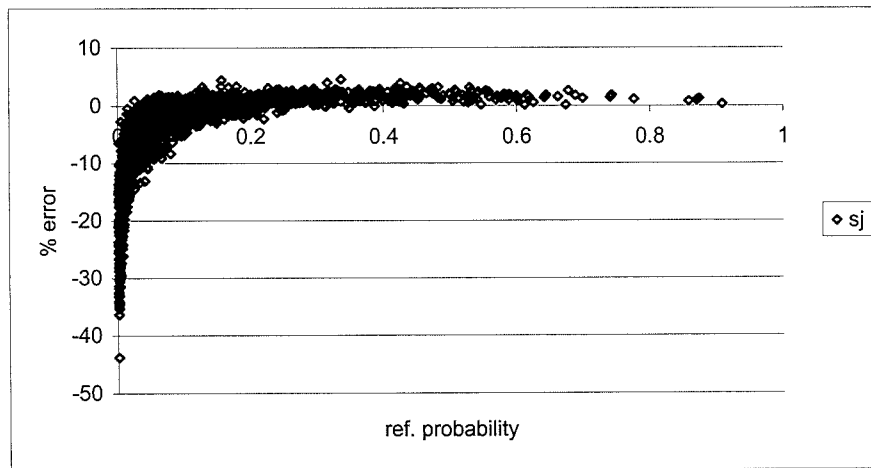


Fig. 3.71 – Percentage errors of the Solow-Joe approximation with averaged results (SJ) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

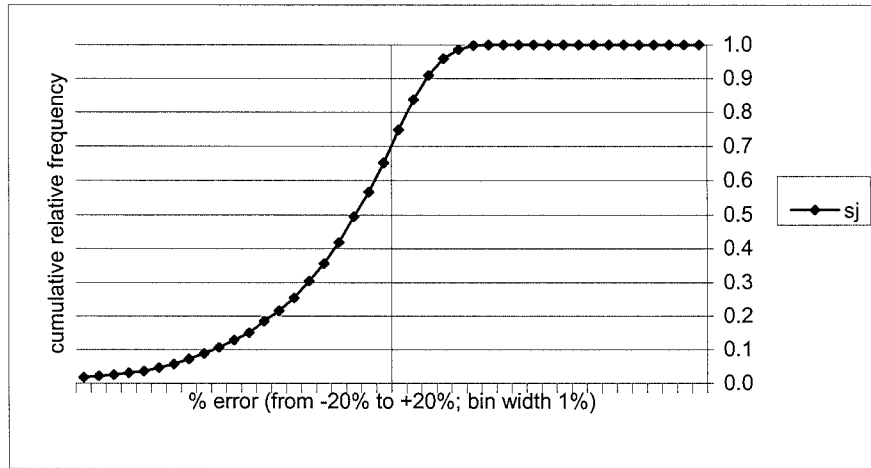


Fig. 3.72 – Cumulative relative frequencies of the percentage errors of the Solow-Joe approximation with averaged results (SJ) on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

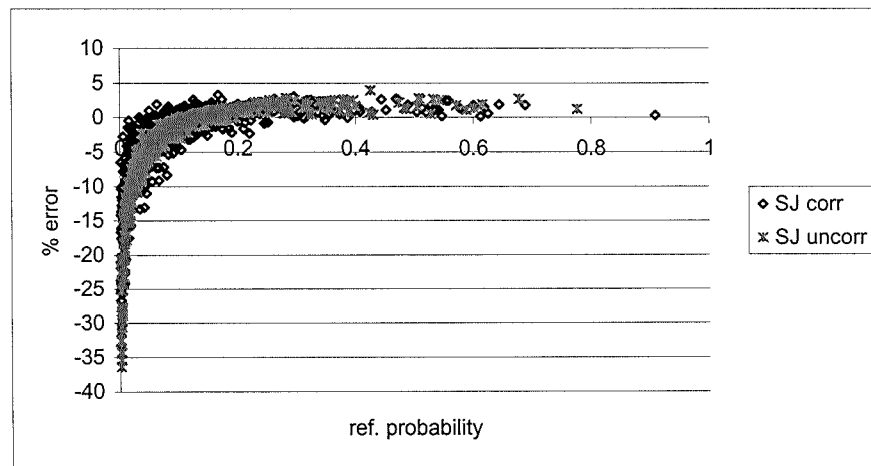


Fig. 3.73 – Percentage errors of the Solow-Joe approximation on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data points for the networks with uncorrelated paths (SJ uncorr) and correlated paths (SJ corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

Aggregate data on the importance of the normalisation on the results can be seen from table 3.35 that reports mean, standard deviation maximum and minimum sum of the probabilities before normalisation. The accuracy of the sum of the raw probabilities decreases for larger choice sets thus making more important the effect of the normalisation for the final results. It should be noted that for the 12 and 15

option cases larger errors could be explained by the averaging over only a sample of the permutations.

Options	Mean SJ	st.dev. SJ	Max SJ	Min SJ
3	1.0000	0.0000	1.0000	1.0000
6	0.9977	0.0051	1.0126	0.9869
9	0.9860	0.0083	1.0104	0.9626
12	0.9702	0.0130	1.0003	0.9384
15	0.9488	0.0154	0.9985	0.9160

Table 3.35 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the averaged Solow-Joe approximation (SJ) on the first and second series of artificial networks.

Although averaging the results over all or a subset of the permutations of the variates is useful to reduce the inaccuracies, it would be convenient to be able to use an optimal calculation order, at least as a heuristic. This is not only to avoid the averaging process but also because, investigating possible optimal calculation sequences with other approximations, processing orders different or opposite to the possible optimal ones were tested and those results, although not reported here, confirmed that using, for instance, the order opposite to the optimal one gives errors that may be large. Averaging the results from all the possible calculation orders, or a large number of random permutations, will therefore mean including also very inaccurate results while spending computing time to obtain them.

Possible optimised single sequences have been tested as heuristics. The possibility of obtaining good accuracy results by selecting the calculation order on the basis of the marginal Normal integral has been considered because of the analogy between the Solow-Joe and the extended Joe approximation with the Taylor approximation.

Reordering the variates to include either the variate with the largest limit of integration first (marked in the figures as SA) or with the smallest one first (marked in the figures as SB) gave unsatisfactory results. Figures 3.74 and 3.75 show an example of the envelopes of percentage inaccuracies obtained that are evidently much larger than those due to the original approximation.

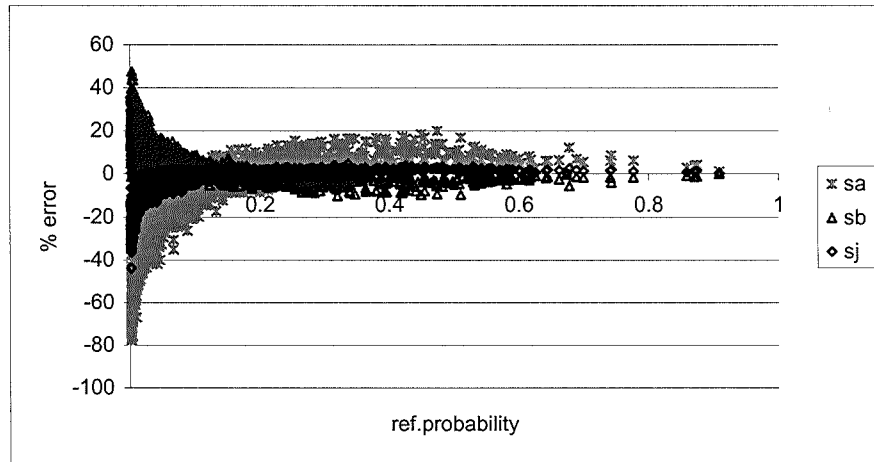


Fig. 3.74 – Percentage errors of the Solow-Joe approximation with full enumeration (sj) and with sa and sb calculation orders (see text) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

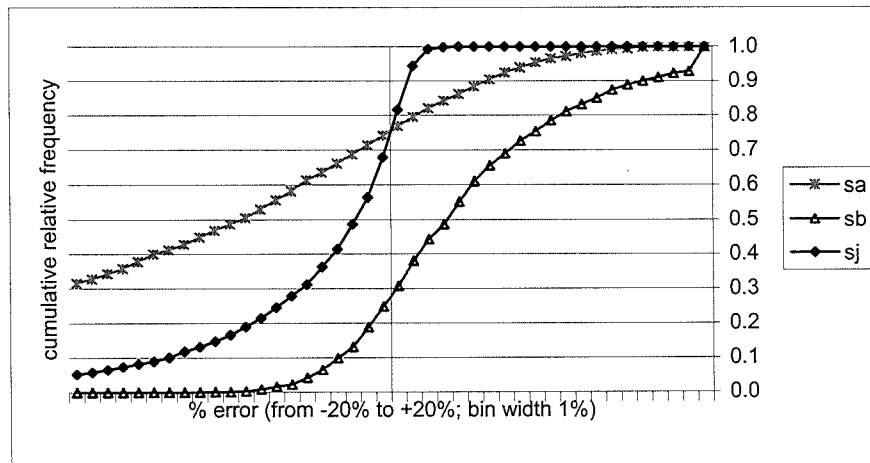


Fig. 3.75 – Cumulative relative frequencies of the percentage errors of the Solow-Joe approximation with full enumeration (sj) and with sa and sb calculation orders (see text) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

Although none of these two calculation sequences can be taken as effective, their opposite trends, that are evident in the data presented as well as in the others obtained, suggested to try to use the average of their results as a possible heuristic method to approximate the results from averaging a larger number of orders.

A selection of the results obtained for this method (marked SC in table and figures) is reported in figure 3.76 and 3.77, and the probabilities for the two example networks are compared with those of the original approximation in tables 3.36 and 3.37.

Option	Num. Int.	SC	Δ SC%	Δ SJ%
1	0.0824	0.0811	-1.65%	-1.38%
2	0.2058	0.2063	+0.24%	+0.34%
3	0.5127	0.5172	+0.90%	+0.63%
4	0.0134	0.0128	-4.34%	-6.41%
5	0.1461	0.1446	-0.97%	-0.16%
6	0.0396	0.0379	-4.35%	-4.31%

Tab 3.36 – Comparison of reference results with those of the Solow-Joe approximation in the SC version (see text) and with averaging (SJ) for the six alternative example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

Option	Num Int	SC	Δ SC%	Δ SJ%
1	0.0544	0.0520	-4.40%	-7.03%
2	0.0285	0.0274	-3.91%	-7.11%
3	0.3246	0.3340	+2.92%	+2.54%
4	0.1582	0.1623	+2.64%	+1.81%
5	0.0950	0.0960	+1.10%	+0.51%
6	0.1029	0.1033	+0.45%	+0.43%
7	0.0146	0.0134	-8.18%	-8.92%
8	0.1032	0.1003	-2.84%	-0.97%
9	0.1186	0.1111	-6.27%	-3.21%

Tab 3.37 – Comparison of reference results with those of the Solow-Joe approximation in the SC version (see text) and with averaging (SJ) for the nine alternative example defined in section 3.6.3. Δ % obtained from the non-approximated values of the results.

The results in the tables are in good accordance with the numerical integration ones and with those from the averaging of the possible variate permutations. Thus they appear to have the same range of likely errors given by the original approximation. This impression is confirmed by graphs such as those in figures 3.76 and 3.77 that show the good accordance of the envelopes of the errors of the original approximation and of the heuristic modification considered here. Graphs depicting the percentage error envelopes for different amount of overlapping and number of paths are not reported here but, again, their trend is similar to that of the original approximation: the trend of underestimation of low choice probabilities and overestimation of larger ones is constant for different overlapping of paths and more correlated options result in data point that are more scattered.

Table 3.38 reports the data on the importance of the normalisation for the SC modification of the Solow-Joe approximation. Also these data are very similar to those for the original Solow-Joe approximation.

Although no theoretical support is put forward here for this modification of the approximation it seems a viable alternative to the original method.

Options	Mean SC	st.dev. SC	Max SC	Min SC
3	1.000	0.0000	1.0000	1.0000
6	0.9963	0.0049	1.0062	0.9790
9	0.9859	0.0111	1.0153	0.9596
12	0.9769	0.0215	1.0455	0.9317
15	0.9677	0.0280	1.0643	0.9106

Table 3.38 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the Solow-Joe approximation in the SC implementation (see text) on the first and second series of artificial networks.

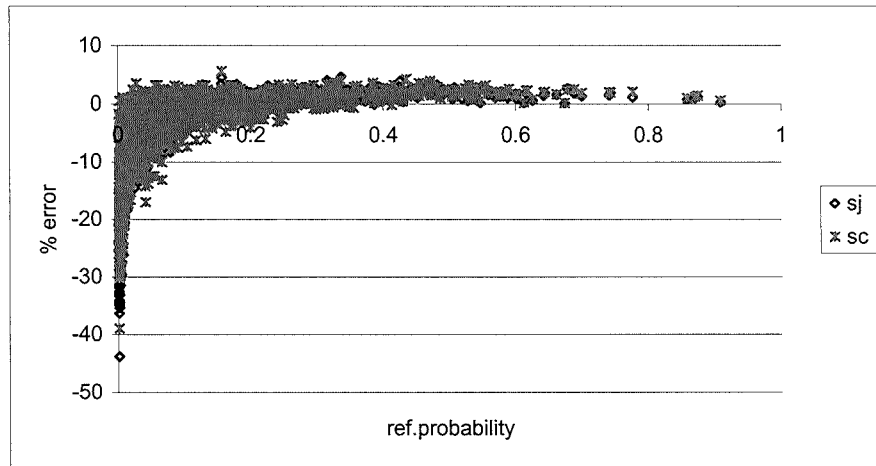


Fig. 3.76 - Percentage errors of the Solow-Joe approximation with full enumeration (sj) and with sc calculation order (see text) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

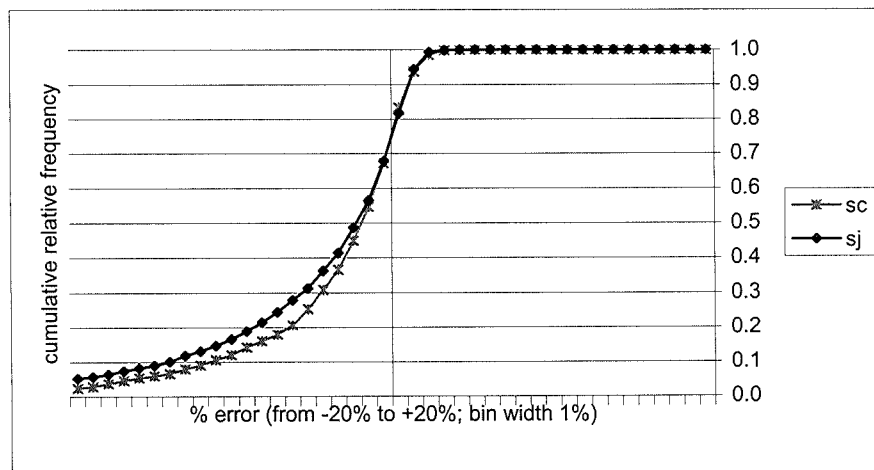


Fig. 3.77 - Cumulative relative frequencies of the percentage errors of the Solow-Joe approximation with full enumeration (sj) and with sc calculation order (see text) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

3.6.4.8 Accuracy of the Extended Joe Approximation

This approximation has been tested for the evaluation of MVN integrals by its author (Joe, 1995) who remarked that it is typically very accurate and that its accuracy decreases with the correlation of the options. He suggested that, for its precision, it could be used for MVN integral of dimension 12 and over as the typical expected errors should be in the fourth decimal place whilst the maximum ones in the third. He also compared it with the Solow-Joe approximation and with the Mendell-Elston approximation (for which he did not mention using a preferred order) finding it more precise.

The experiments reported here were carried out using the original program developed by Joe and therefore obtaining the results before normalisation by averaging those resulting from all the possible orderings of the variates, for 6 and 9 options, or from a sample of 1000 of them, for the cases of 12 and 15 options.

The data obtained confirmed the good accuracy of this method that, however, should be seen to some extent separately from the other ones as it uses up to quadrivariate MVN integrals calculated exactly by numerical integration, whilst the other methods employ either only univariate or only univariate and bivariate Normal integrals calculated directly.

The results for the two examples with 6 and 9 options used throughout this chapter, reported in tables 3.39 and 3.40, where the extended Joe approximation data are marked EJ, confirmed the good accuracy of this method. It should be noted that in the six option example the approximation is applied only once for each option (and each variate ordering).

Option	Num.Int.	EJ	$\Delta EJ\%$
1	0.0824	0.0823	-0.23%
2	0.2058	0.2059	+0.03%
3	0.5127	0.5131	+0.09%
4	0.0134	0.0134	-0.39%
5	0.1461	0.1459	-0.08%
6	0.0396	0.0396	-0.38%

Tab 3.39 – Comparison of Extended Joe and reference results for the six alternative example. $\Delta\%$ obtained from the non-approximated values of the results.

Option	Num.Int.	EJ	$\Delta EJ\%$
1	0.0544	0.0539	-0.99%
2	0.0285	0.0283	-0.73%
3	0.3246	0.3259	+0.42%
4	0.1582	0.1586	+0.29%
5	0.0950	0.0952	+0.18%
6	0.1029	0.1029	+0.04%
7	0.0146	0.0145	-0.53%
8	0.1032	0.1028	-0.36%
9	0.1186	0.1178	-0.63%

Tab 3.40 – Comparison of Extended Joe and reference results for the nine alternative example. $\Delta\%$ obtained from the non-approximated values of the results.

The figures 3.78, 3.79, 3.80 and 3.81 present a selection of the data obtained and give a further confirmation of the precision of this method which shows increasing inaccuracies for small actual probabilities and as the number of choice options increases. However, such inaccuracies remain of limited percentage importance.

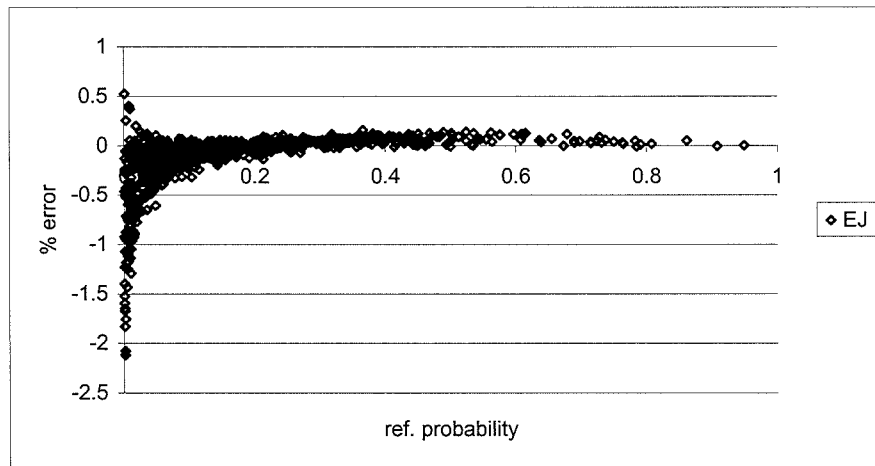


Fig. 3.78 - Percentage errors of the first and extended Joe approximations on the 6 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

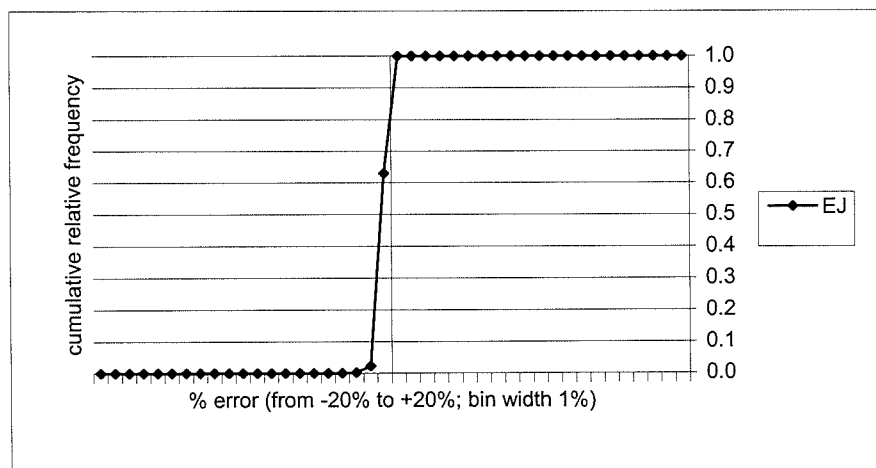


Fig. 3.79 - Cumulative relative frequencies of the percentage errors of the Extended Joe approximation on the 6 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

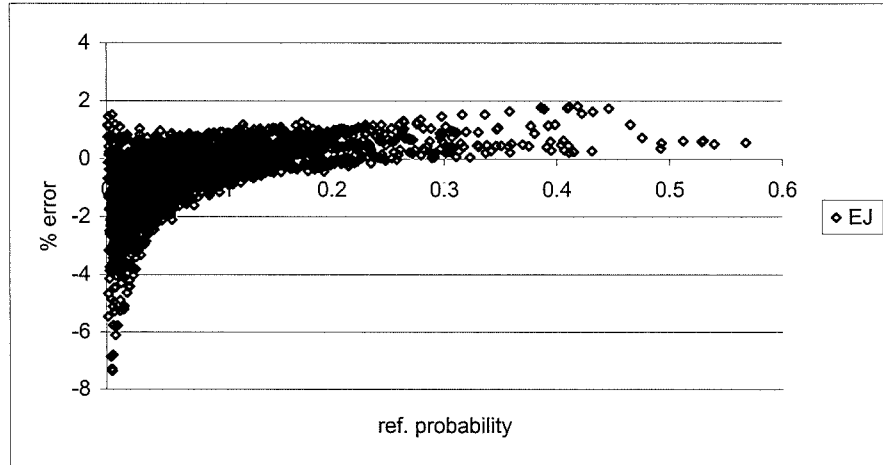


Fig. 3.80 – Percentage errors of the first and extended Joe approximations on the 12 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

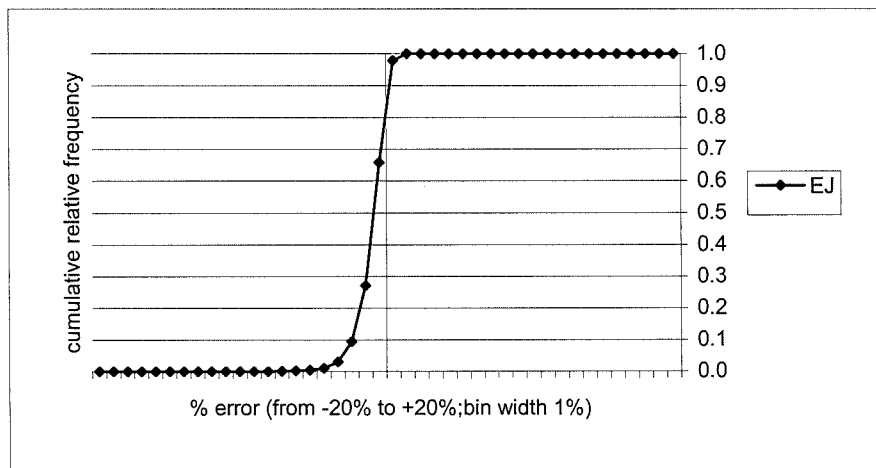


Fig. 3.81 – Cumulative relative frequencies of the percentage errors of the Extended Joe approximation on the 12 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

The correlation of the options seems to have limited effect on how important are the percentage errors reported by this approximation as shown in fig. 3.82. However, as shown in the picture, the envelope of the percentage errors for networks with uncorrelated paths has a more defined arch trend that spans a slightly larger range than the envelope of the percentage errors for the cases with correlated options.

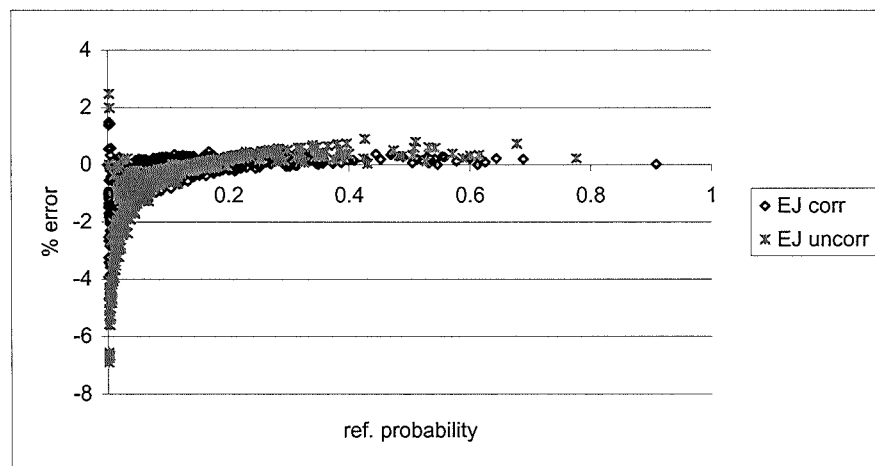


Fig. 3.82 – Percentage errors of the extended Joe approximation on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Only the data points for the networks with uncorrelated path (EJ uncorr) and correlated paths (EJ corr) are plotted. Data points for reference probabilities smaller than 0.001 are not depicted.

Table 3.41 shows the very limited influence of the normalisation on this approximation's results in the cases examined that is in accordance with the good precision already remarked. The data for the three option cases are due to the fact that in those cases the approximation uses directly a routine for bivariate normal integrals that gives precise results.

Options	Mean EJ	St.dev. EJ	Max EJ	Min EJ
3	1.0000	0.0000	1.0000	1.0000
6	0.9993	0.0003	1.0001	0.9986
9	0.9956	0.0017	0.9999	0.9918
12	0.9902	0.0027	0.9962	0.9833
15	0.9843	0.0043	0.9947	0.9742

Table 3.41 – Table of mean, standard deviation, maximum and minimum values of the sum of the choice probabilities prior to normalisation for the Extended Joe approximation on the first and second series of artificial networks.

The effectiveness of the averaging device put forward by Joe for avoiding possible inaccuracies due to unfavourable variate processing sequences is demonstrated in figs. 3.83 and 3.84 that compare the percentage errors obtained by averaging the approximation results over all the possible permutations with those obtained using a single processing order (here that resulting from the sequence in which the options have been coded in the input file to the computer program).

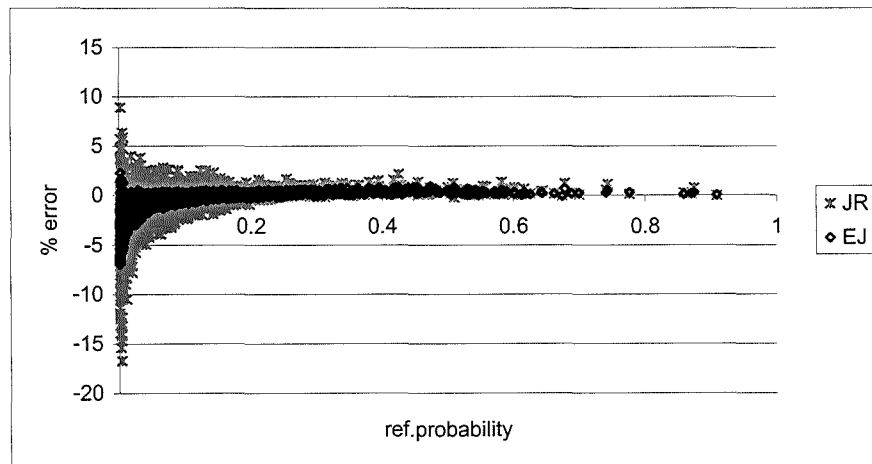


Fig. 3.83 – Percentage errors of the Extended Joe approximation (with averaging of the results from all possible permutations –EJ– and a single non-optimised calculation order – JR) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

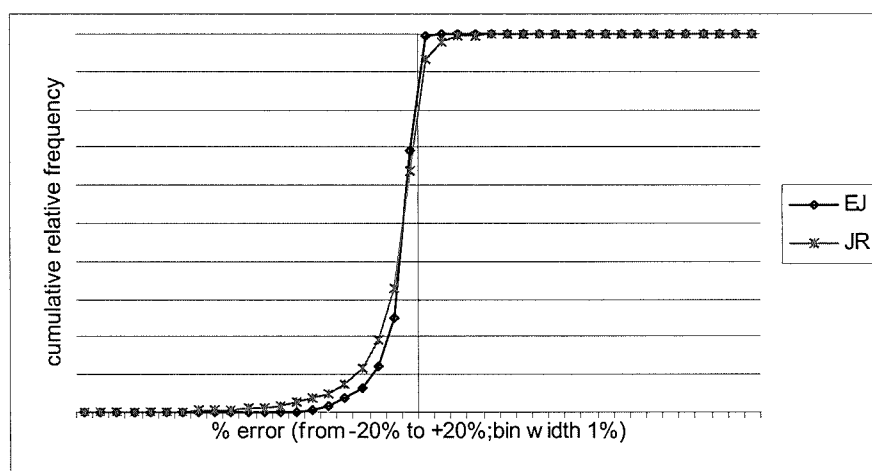


Fig. 3.84 – Cumulative relative frequencies of the percentage errors of the Extended Joe approximation (with averaging of the results from all possible permutations – EJ- and a single non-optimised calculation order – JR) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

Similarly to the case of the Solow-Joe approximation, the use of the simple processing sequence rules has been tested to try to obtain final results as precise as those from the averaging process without resorting to such device. Two possible orders were considered as heuristic methods (and were suggested by their effect with the Solow-Joe approximation): reordering the variates so that those with the smallest marginal integrals are processed first, or in the opposite order. Both gave very similar results that seem not better than those obtained using the instance of a random order considered here (see figs. 3.85 and 3.86 that compare results from one such order with those of the random one) and much more inaccurate than those obtained from the original approximation as shown in figure 3.87 and 3.88. Averaging the results obtained with those two opposite variate reorderings gives no particular advantage as is expected observing their common trend and, more importantly, does not give an envelope of errors as good as that of the original approximation. Thus no particular processing order can be put forward, even as a heuristic, after the investigations carried out.

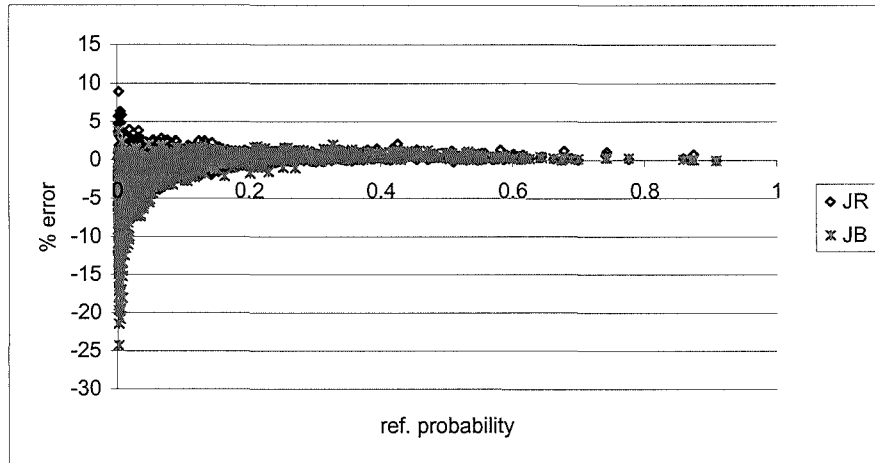


Fig. 3.85 - Percentage errors of the extended Joe approximation with the variate giving the largest marginal integral processed first (JB) and with an instance of variate processing order (JR) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

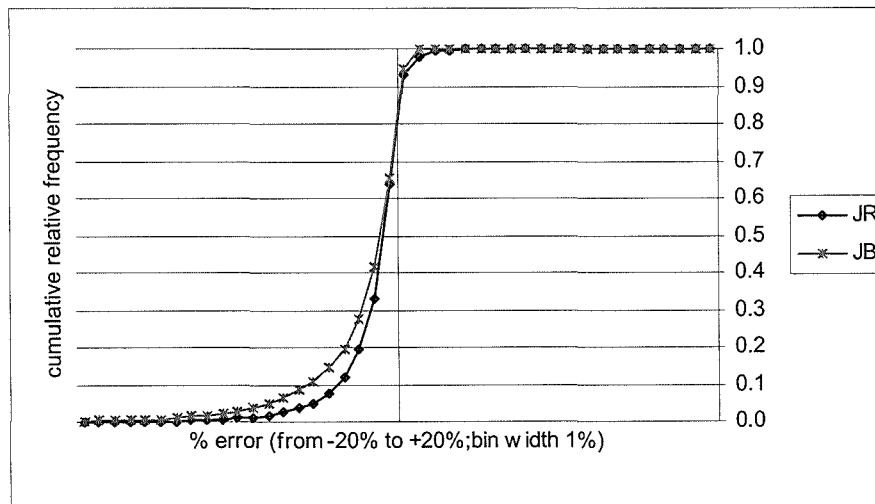


Fig. 3.86 - Cumulative relative frequencies of the percentage errors of the extended Joe approximation with the variate giving the largest marginal integral processed first (JB) and with an instance of variate processing order (JR) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

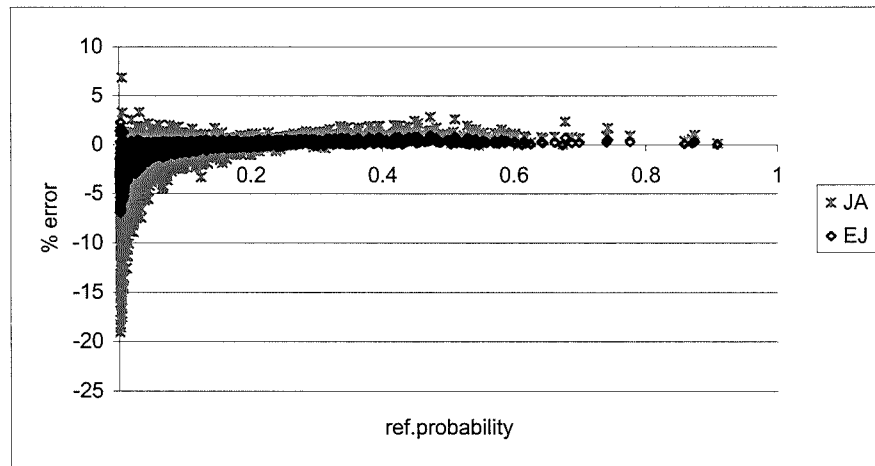


Fig. 3.87 – Percentage errors of the extended Joe approximation with full enumeration (EJ) and with the calculation orders processing the variates with the smallest marginal integral first (JA) on the 9 option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

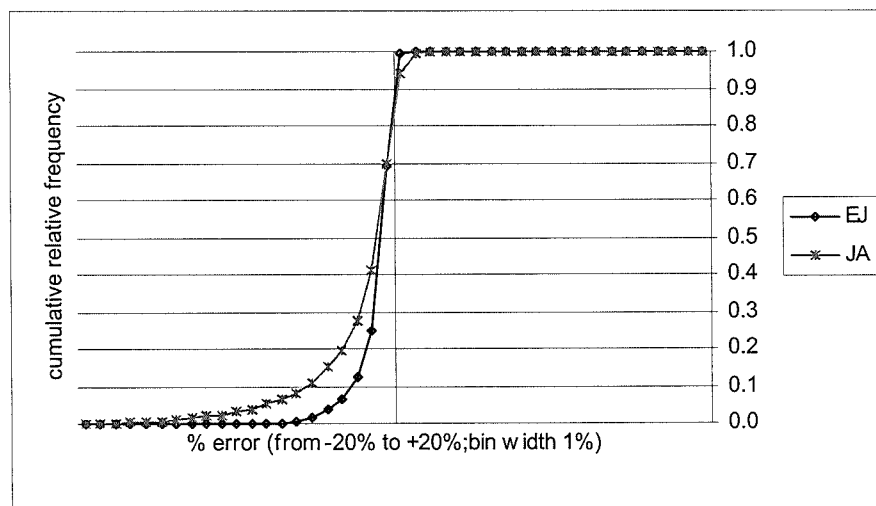


Fig. 3.88 – Cumulative relative frequencies of the percentage errors of the extended Joe approximation with full enumeration (EJ) and with the calculation orders processing the variates with the smallest marginal integral first (JA) on the 9 option cases from the first and second series of artificial networks. Data for reference probabilities smaller than 0.001 are not included.

3.6.5 Summary and Comparison of Accuracy and Calculation Time Results

The accuracy of the different MNP approximation methods, illustrated so far separately for each approximation or approximation group, is compared in this section using the same sort of graphs employed in the previous paragraphs and presenting some additional summary statistics on their precision along with time calculation summary data.

The graphs reported in this section are all relative to the cases of 12 paths in the first and second series of networks. Although it was mentioned that the actual number of paths used in traffic assignment could be smaller and the errors produced by the approximations for smaller choice sets are of lower importance, the figures are intended to provide a common and rather demanding ground of comparison between the approximations.

Tables 3.42, 3.43, 3.44 report summary statistics about the approximations' precision obtained from the results on the first and second series of networks: they list the average percentage error and, in brackets, its standard deviation, organised by approximation and by number of options. The three tables proposed exclude the errors for reference choice probabilities under three different thresholds to evaluate the approximations also considering that, in general, the biggest percentage errors are encountered for the smallest choice probabilities. The zero values for the approximations of the FOMN group, the Solow-Joe and the extended Joe in cases of three choice options are due to the fact that in those cases those approximations use simply a routine for numerical integration therefore the results agree with the reference ones.

The time calculation data gathered during the experiments are reported in aggregate form in tables 3.45 and 3.46 that list average and standard deviation of the calculation time for all the experiments on the first and second series of networks, organised by approximation and number of options in the choice set. The timings reported have been obtained on a Pentium II 350 MHz PC and are thus relevant to that computing equipment and the programs employed, but should give a fair comparison of the relative computation expenses of the approximations examined.

The generalised FOMN method with the heuristic optimised processing order proposed and the approximation of Mendell-Elston (especially with the heuristic processing order suggested by Rice *et al.*) and the extended Joe method can be characterised as the most precise approximations amongst those investigated both looking at the statistics reported and, more simply, looking at the graphs of their percentage error envelopes.

An example of how the Mendell-Elston approximation (with the Rice *et al.* order) compares with the optimised order generalised FOMN is given in figure 3.89. With fewer options the inaccuracies are smaller but the trends are similar. In general the Mendell-Elston method tends to have more disperse results and mainly overestimates the low probabilities whilst underestimating larger ones. The opposite trend of the generalised FOMN method comes with inaccuracies of even smaller percentage importance. It should be noted, though, that the FOMN method uses direct calculation of bivariate Normal probabilities whilst the Mendell-Elston does not. The employment of such a routine as part of an enhanced Mendell-Elston approximation could possibly improve its results further.

The better performance of the generalised FOMN approximation entails, however, a higher computational cost than that necessary for the Mendell Elston method: the average calculation times shown on table 3.45 are about five times longer than those for the Mendell-Elston (except in the three option case). The comparison with the calculation times of the Mendell-Elston approximation with the Kamakura order gives similar results.

The generalised FOMN approximation without optimised order gives results of similar precision to the Mendell-Elston method (see the example in fig. 3.90) although it presents underestimation of some low choice probabilities that is not present for the Mendell-Elston with Rice *et al.* order (see fig 3.91). The generalised FOMN approximation without variate reordering requires, as its optimised order

version, longer calculation times, between three and four times longer than the Mendell-Elston (except in the three option case). This is probably due to the need to solve the internal optimisation problems described in section 3.5.5 several times at each application of the approximation. Comparing the data reported in table 3.45 for the timings of the FOMN approximation with and without optimised order, suggests that reordering the options in the desired sequence implies calculation times that are about 1.4 times those without reordering. This proportion is confirmed when comparing the data for the crude FOMN method in the versions with and without optimised order. It should be remarked once more that the approximation without optimised order has been considered as this is the way it is presented in the relevant literature. However, it is preferable to perform the calculations according to a set order since a random order as that considered here could give results of varying precision depending on how favourable it is.

An example of the trend of inaccuracies given by the extended approximation of Joe is reported in figure 3.92, that compares it with the trend of percentage errors of the Mendell-Elston method with the Rice *et al.* order. The two approximations have opposite trends of the errors but the absolute values of the inaccuracies are similar. In fact, the extended Joe approximation has a trend of the errors similar to that of the generalised FOMN approximation with optimised processing order (it tends to underestimate low choice probabilities and overestimate larger ones) but with larger inaccuracies. However, the results of the extended Joe approximations come at a high computational cost: the times obtained in the experiments are, on average, three orders of magnitude larger than the Mendell-Elston ones for the cases of 6 options, and even larger for larger choice sets. Table 3.45 also shows that the extended Joe calculation times are much higher than those for the generalised FOMN approximation. Such long calculation times are due to considering results obtained from different permutations of the variates but also to the calculation of a number of up to four-variate marginal Normal integrals, as explained in the description of the method.

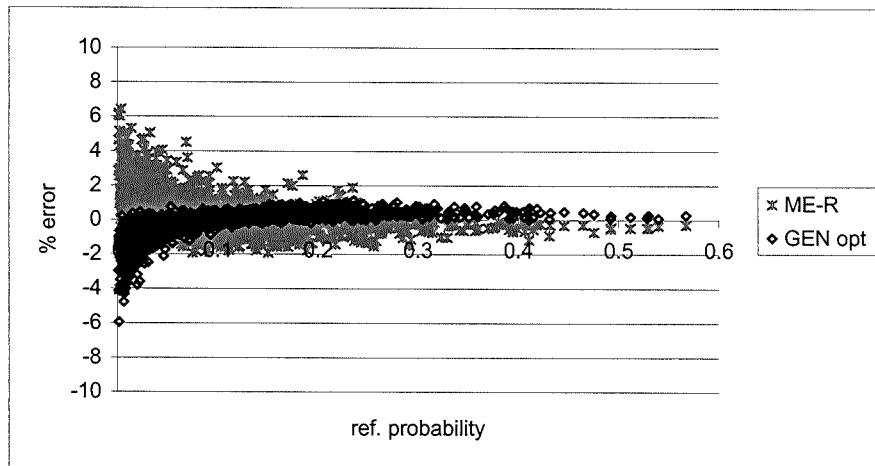


Fig. 3.89 - Comparison of the percentage errors due to the Mendell-Elston method with Rice et al. ordering (ME-R) and the optimised order generalised FOMN method (GEN opt) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

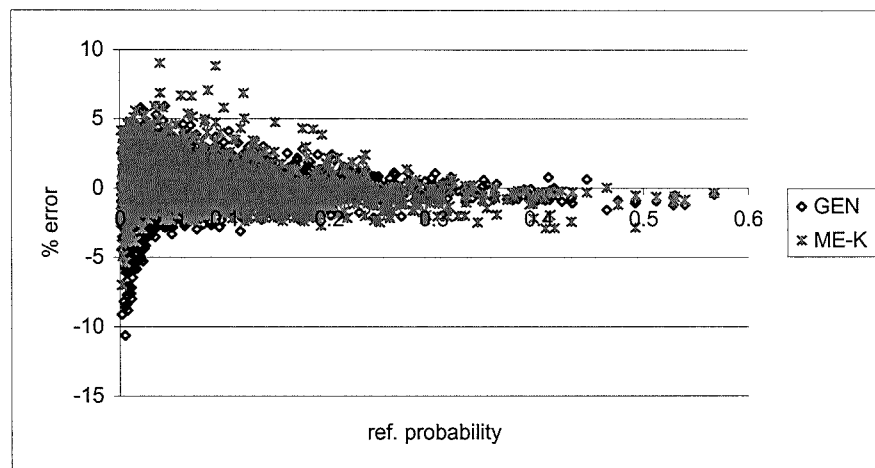


Fig. 3.90 - Comparison of the percentage errors due to the Mendell-Elston method with Kamakura ordering (ME-K) and the generalised FOMN method without optimised order (GEN) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

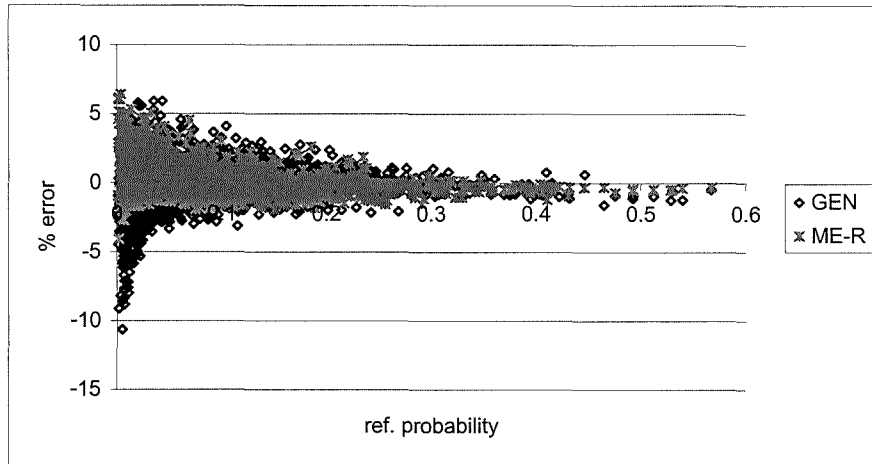


Fig. 3.91 - Comparison of the percentage errors due to the Mendell-Elston method with Rice et al. ordering (ME-R) and the generalised FOMN method without optimised order (GEN) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

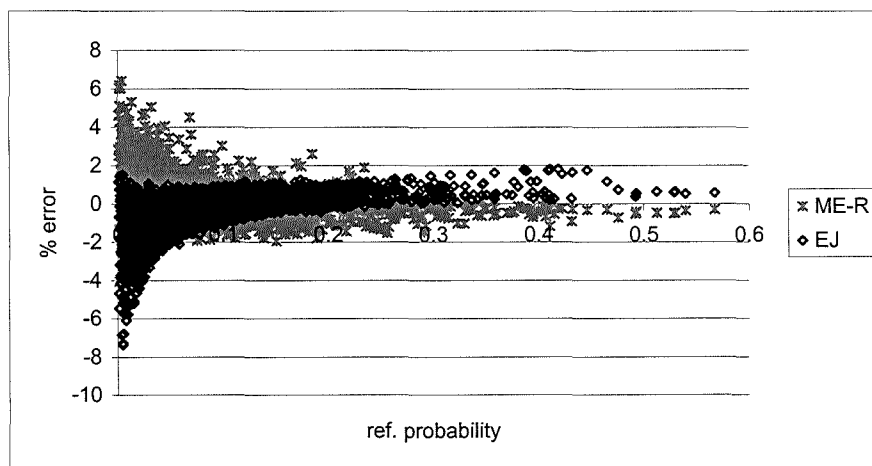


Fig. 3.92 - Comparison of the percentage errors due to the Mendell-Elston method with Rice et al. ordering (ME-R) and the extended Joe approximation with result averaging (EJ) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

There are a number of approximations that give rather satisfactory results but are rather less accurate than those mentioned above. Amongst these is the separated split approximation. An example of how it compares with the generalised FOMN approximation with optimised order is given in figure 3.93.

In the papers putting forward the separated split approximation, Langdon (1984a,b) warned about the exponential increase of calculation time resulting from the extension and complexity of the computation structure of the approximation for large choice sets. In fact, the calculation times recorded for all the approximations grow more than linearly with the number of options in the choice set but the times required by the separated split method have a much steeper increase than some others beyond a certain number of alternatives. For instance, from the data presented in table 3.45 it can be seen that this happens beyond 9 choice options when the Mendell-Elston or the Taylor series expansion approximation are compared with the separated split, or beyond 12 options when the comparison is against the optimised order generalised FOMN method.

The approximations based on the Taylor series give very similar results as seen in the section that reports their accuracy data. There it was noted, however, that the second order approximation should be preferred. In practice, both orders of the Taylor approximation require the same computational effort, as shown by table 3.45 and as should be expected since the difference in the operations carried out is minimal. It should also be noted that the Taylor series approximations require substantially the same computational times as the Mendell-Elston method (with either optimised order). This is consistent with the similarity of the calculation structures and with the minimal practical differences between the operations carried out each time the approximations are applied. The precision reached by the Taylor series methods is similar to that of the separated split method, as shown by figures as 3.94: the two methods have opposite error trends but, in absolute value the inaccuracies are rather similar.

The trend and the magnitude of the percentage errors of the Taylor series approximations are similar to that of the Solow-Joe approximation, although the latter gives larger percentage underestimation of low choice probabilities, especially for smaller choice sets. Figure 3.95 is an example of comparison of the trends of such approximations. However, the calculation times required by the Solow-Joe approximation when all or a sample of the results from different permutations of the variates are averaged to give the final result are much longer than those required by the Taylor series approximation, as can be seen from table 3.45. For instance in the 6 option cases (with averaging of the results for all possible variate permutations) they are, on average, about 50 times larger than the calculation times required by the Taylor approximation.

The difference in calculation times is more favourable when the proposed heuristic averaging of the results over two permutations only is considered. The times are, however, approximately 3 to 4 times larger than those required by the Taylor series approximation for a similar precision (and than those required from the Mendell-Elston approximation for a higher precision). An example of how the results from the heuristic limited averaging version of the Solow-Joe approximation compare with the Taylor ones is given in fig. 3.96.

To close the discussion on the relative precision of the Solow-Joe approximation it is interesting to note in plots as that in figure 3.97, the difference in accuracy between the Solow-Joe and the extended Joe approximation, both based on the same idea but refined to a different order.

The crude FOMN method with heuristic optimal processing order is much less precise than the corresponding generalised FOMN method, as seen in the section on their accuracy, and is also less precise than the generalised FOMN without optimal processing sequence. In fact, the accuracy and the trend of the crude FOMN method can be approximately compared with those of the Taylor series or the Solow-Joe method (see the example in figure 3.98), although the crude FOMN has a marked tendency to overestimate actual high probabilities that is not present in the other approximations. It should be noted, though, that the time required to carry out the

crude FOMN calculations with reordering is, on average, three or four times that required by an approximation using the Taylor series (and that of the Mendell-Elston, that is much more precise), except for the three option cases. However, the crude FOMN with reordering is, on average, slightly faster than the heuristic limited averaging Solow-Joe approximation (except in the 6 option cases).

The percentage inaccuracies of the methods seen above are moderate when compared with those of the very fast improved Clark method, that tends to give large percentage errors especially for options with real choice probability of about less than 10%, as shown in figure 3.99 (although it should be noted that the largest percentage errors are observed for very small actual choice probabilities). Not including variate reordering, the improved Clark method requires calculation times that are shorter than with other approximations and increase more than linearly with the number of options, but at a lower rate than in most other cases.

In fact, the only method whose increase in calculation time with the number of options is less pronounced is the heuristic simple Clark method, the fastest but also the least precise amongst the methods presented. The comparison with the improved Clark method in fig. 3.100 shows large percentage errors also for large choice probabilities. The particularly short calculation times and the fact that they increase only slightly more than linearly with the number of options are due to the absence of variate reordering (no optimised order is proposed in the literature or was tried here) and to the fact that the number of times the approximation calculations are carried out grows only linearly with the number of options whilst in many other cases (except with the separated split, the Solow-Joe and the extended Joe approximation) it grows approximately with the square of the number of options.

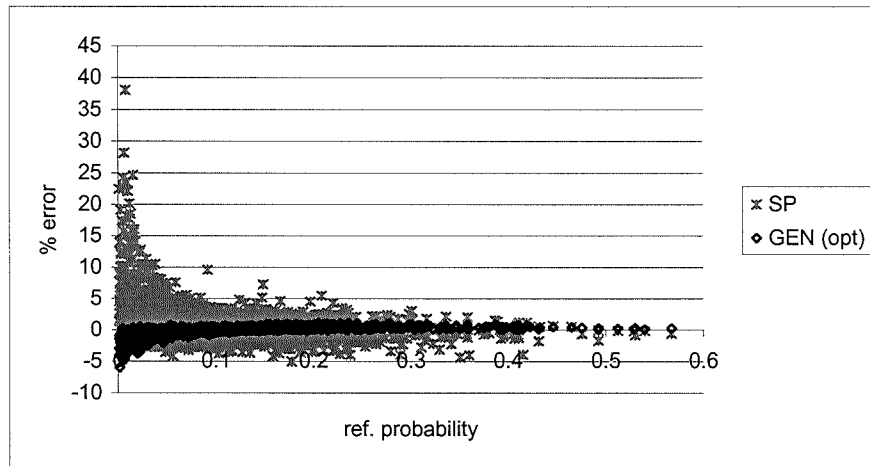


Fig. 3.93 - Comparison of the percentage errors due to the separated split method (SP) and the optimised order generalised FOMN method (GEN opt) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

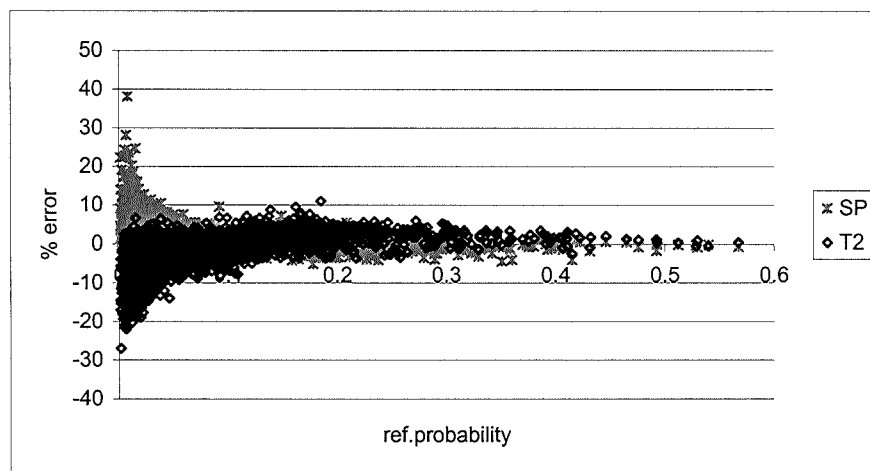


Fig. 3.94 - Comparison of the percentage errors due to the separated split method (SP) and the second order Taylor approximation (T2) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

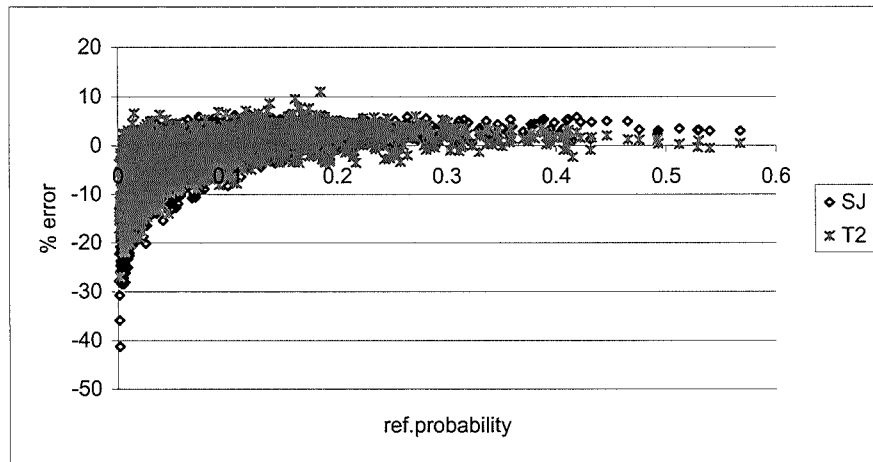


Fig. 3.95 - Comparison of the percentage errors due to second order Taylor approximation (T2) and the Solow-Joe approximation (SJ) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

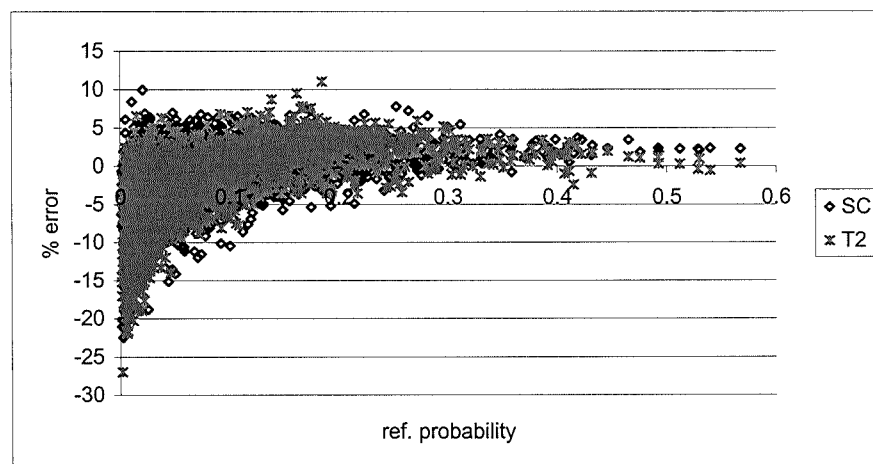


Fig. 3.96 - Comparison of the percentage errors due to second order Taylor approximation (T2) and the Solow-Joe approximation with heuristic limited reordering (SC) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

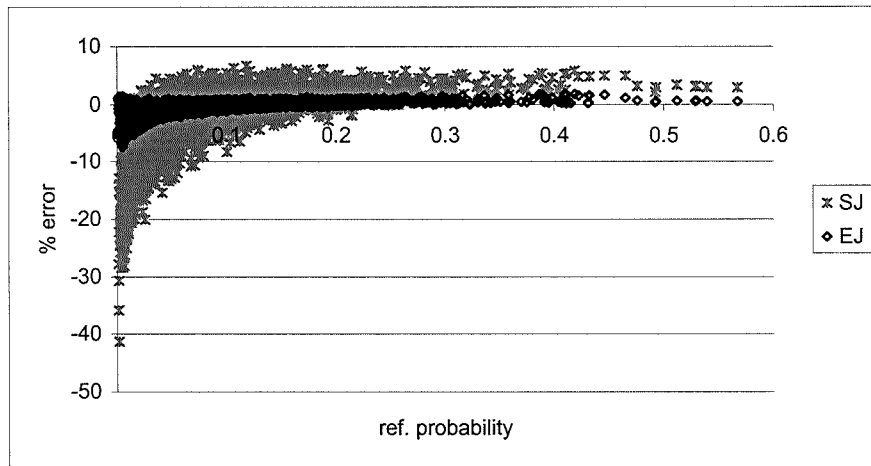


Fig. 3.97 - Comparison of the percentage errors due to the Extended Joe method (EJ) and the Solow-Joe approximation (SJ) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

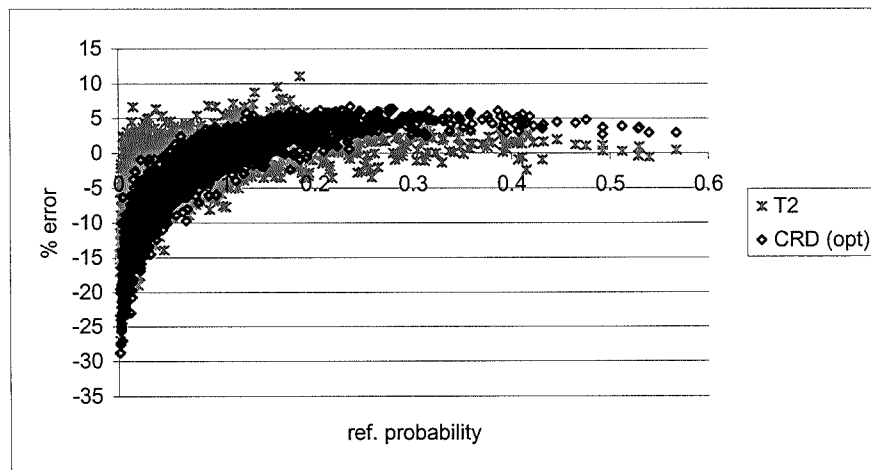


Fig. 3.98 - Comparison of the percentage errors due to the crude FOMN method with optimal reordering (CRD opt) and the second series Taylor approximation (T2) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

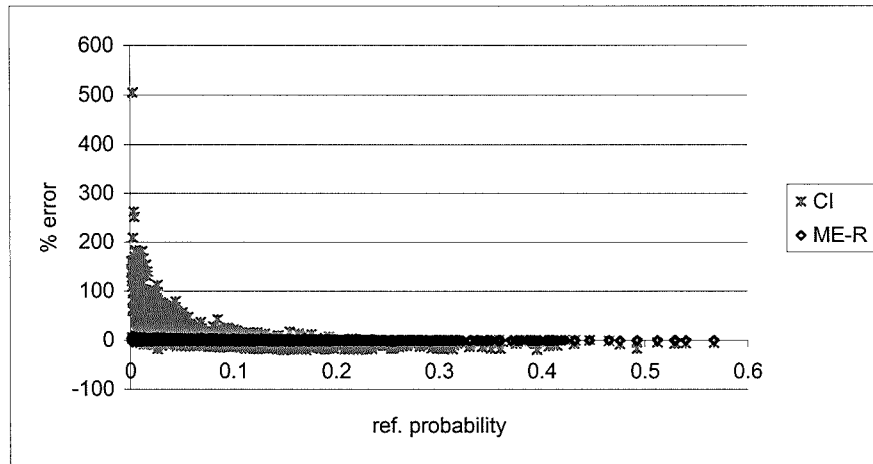


Fig. 3.99 - Comparison of the percentage errors due to the improved Clark method (CI) and the Mendell-Elston approximation with the Rice et al. ordering (ME-R) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

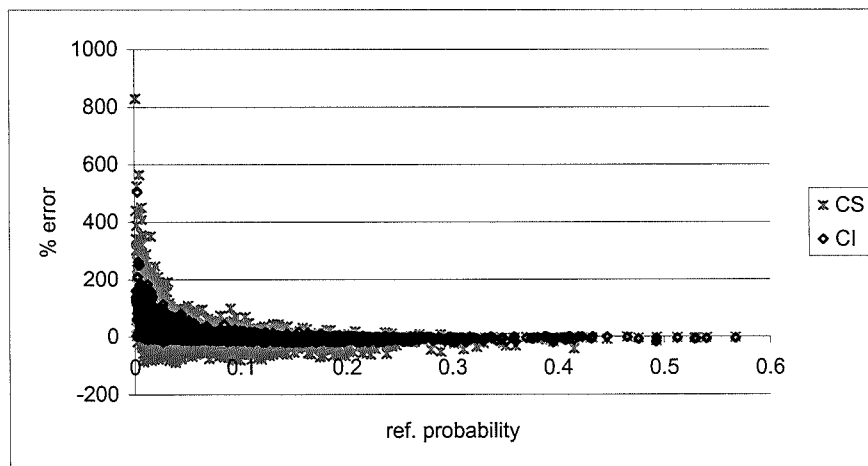


Fig. 3.100 - Comparison of the percentage errors due to the simple Clark method (CS) and the improved Clark approximation (CI) on the twelve option cases from the first and second series of artificial networks plotted against the reference probabilities. Data points for reference probabilities smaller than 0.001 are not depicted.

The accuracy results reported clearly point at the methods already mentioned (generalised FOMN with heuristic optimal processing sequence, Mendell-Elston and extended Joe) as the best ones to be employed for the practical calculation of the MNP choice function. The consideration of the calculation times, however, suggests that the generalised FOMN and the Mendell-Elston approximations should certainly be preferred. In particular, the shorter calculation times reported for the Mendell-Elston method characterise it as the approximation giving the best trade-off between accuracy and calculation time. Moreover, it should be noted that to limit the calculation times required by the extended Joe method with large choice sets, it is necessary to use only a sample of the results from the different possible permutations of the variates, thus introducing an element of non-repeatability, when part of the interest in using analytical approximations is to ensure that results are repeatable.

Amongst the approximations giving an intermediate quality of the accuracy results, the Taylor series one and the separated split seem to be those giving better results with limited calculation time. The heuristic reordering of the Solow-Joe approximation, the crude FOMN with optimal reordering and the original Solow-Joe approximation (listed in order of increasing typical calculation times) do not improve noticeably on the quality of the Taylor series and of the separated split approximations and require always longer calculation times than the Taylor series methods and often longer calculation times than the separated split approximation. Moreover, the same point raised above on the non-repeatability of the results due to averaging only a part of the probabilities from different possible permutations of the variates applies to the Solow-Joe method, although in this case it is possible to obviate it using the heuristic reordering proposed.

The very fast improved Clark approximation gives large errors for low choice probabilities, and its heuristic variation seems to give large percentage errors over the whole range of possible probabilities, though the worst results are obtained for low choice probabilities.

Numerical integration has been used here only to obtain reference data by specifying a high required precision of the results giving the high computation times reported in table 3.46. These are possibly also due to the computing equipment employed. In fact, the times reported by Genz (1992) using a different class of computer are much lower. The feasibility of the direct use of the numerical integration method of Genz on a computing equipment with higher specification and with a lower requested precision of the results should be verified. It should, however, be expected that approximations as the generalised FOMN and the Mendell-Elston would remain competitive both for precision and calculation time.

Opt.	CS	CI	ME-K	ME-R	SP	TF	TS
3	7.26 (40.50)	2.14 (3.92)	-0.03 (0.95)	0.08 (0.17)	0.35 (1.53)	-0.17 (0.71)	0.06 (0.18)
6	20.91 (102.99)	4.32 (18.16)	-0.04 (1.58)	0.23 (0.51)	0.70 (3.43)	-1.31 (2.57)	-0.88 (1.93)
9	25.88 (75.59)	11.23 (33.31)	-0.09 (1.94)	0.34 (0.76)	1.10 (4.65)	-2.31 (4.07)	-1.81 (3.52)
12	20.60 (59.14)	12.05 (31.92)	0.30 (1.32)	0.34 (1.03)	0.72 (3.10)	-3.04 (5.31)	-2.61 (4.89)
15	23.48 (66.18)	18.87 (51.61)	0.26 (1.76)	0.29 (1.26)	0.88 (3.76)	-4.03 (6.37)	-3.57 (5.99)

Opt.	CRD(opt)	GEN	GEN(opt)	SJ	SC	J1
3	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
6	-2.42 (3.81)	-0.02 (0.53)	-0.11 (0.21)	-1.66 (3.65)	-1.60 (3.56)	-0.12 (0.28)
9	-4.14 (5.42)	-0.08 (1.26)	-0.30 (0.49)	-4.35 (7.13)	-3.35 (5.81)	-0.70 (1.25)
12	-4.51 (6.45)	-0.29 (1.59)	-0.43 (0.76)	-3.23 (5.93)	-1.37 (3.84)	-0.53 (1.09)
15	-6.08 (7.63)	-0.62 (1.93)	-0.71 (1.09)	-5.10 (8.31)	-1.55 (4.60)	-1.03 (1.72)

Table 3.42 – Average percentage error and standard deviation (in brackets) for MNP approximations in cases with reference probability at least 0.001. The first column indicates the number of options in the choice set. The approximations are marked as in the rest of the chapter.

Opt.	CS	CI	ME-K	ME-R	SP	TF	TS
3	2.59 (12.44)	0.68 (2.39)	0.05 (0.29)	0.08 (0.17)	0.19 (1.02)	-0.16 (0.71)	0.06 (0.18)
6	9.20 (28.24)	2.13 (6.96)	0.11 (0.91)	0.21 (0.50)	0.27 (1.47)	-1.13 (2.30)	-0.76 (1.73)
9	10.83 (37.29)	4.86 (15.07)	0.15 (1.28)	0.28 (0.72)	0.43 (2.24)	-1.84 (3.65)	-1.46 (3.17)
12	12.47 (39.99)	7.41 (21.53)	0.30 (1.27)	0.27 (0.92)	0.46 (2.64)	-2.53 (4.87)	-2.16 (4.48)
15	12.58 (41.42)	9.17 (27.75)	0.27 (1.63)	0.25 (1.12)	0.48 (3.08)	-3.15 (5.63)	-2.75 (5.28)

Opt.	CRD(opt)	GEN	GEN(opt)	SJ	SC	J1
3	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
6	-1.92 (3.31)	0.01 (0.48)	-0.09 (0.17)	-1.02 (2.22)	-0.99 (2.17)	-0.08 (0.17)
9	-3.06 (4.56)	-0.01 (1.03)	-0.23 (0.41)	-2.46 (4.27)	-1.97 (3.76)	-0.42 (0.77)
12	-3.66 (5.61)	-0.20 (1.46)	-0.34 (0.66)	-2.43 (4.90)	-1.13 (3.58)	-0.45 (0.98)
15	-4.66 (6.49)	-0.37 (1.60)	-0.53 (0.87)	3.59 (6.71)	-1.35 (4.25)	-0.79 (1.48)

Table 3.43 – Average percentage error and standard deviation (in brackets) for MNP approximations in cases with reference probability at least 0.01. The first column indicates the number of options in the choice set. The approximations are marked as in the rest of the chapter.

Opt.	CS	CI	ME-K	ME-R	SP	TF	TS
3	1.41 (9.12)	0.20 (0.96)	0.03 (0.19)	0.04 (0.12)	0.06 (0.36)	-0.14 (0.71)	0.05 (0.16)
6	2.42 (16.79)	0.60 (3.52)	0.15 (0.64)	0.12 (0.47)	0.01 (0.99)	-0.60 (1.82)	-0.38 (1.37)
9	0.39 (18.79)	0.09 (5.62)	0.16 (0.98)	0.09 (0.55)	-0.04 (1.47)	-0.48 (2.51)	-0.35 (2.16)
12	-1.78 (21.60)	-0.89 (8.76)	0.12 (1.14)	0.03 (0.69)	-0.16 (1.86)	-0.13 (3.25)	-0.04 (2.95)
15	-4.08 (20.14)	-3.05 (10.28)	0.05 (1.33)	-0.04 (0.73)	-0.26 (2.19)	0.52 (3.34)	0.52 (3.13)

Opt.	CRD(opt)	GEN	GEN(opt)	SJ	SC	J1
3	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
6	-0.88 (2.39)	0.03 (0.39)	-0.04 (0.12)	-0.29 (1.13)	-0.31 (1.20)	-0.03 (0.09)
9	-0.78 (2.88)	0.09 (0.75)	-0.06 (0.24)	-0.39 (2.14)	-0.45 (2.21)	-0.09 (0.40)
12	-0.16 (3.33)	0.13 (1.01)	-0.01 (0.38)	-0.08 (3.08)	-0.15 (2.89)	-0.04 (0.63)
15	0.54 (3.69)	0.23 (0.94)	0.04 (0.46)	0.52 (3.85)	0.04 (3.31)	0.04 (0.86)

Table 3.44 – Average percentage error and standard deviation (in brackets) for MNP approximations in cases with reference probability at least 0.05. The first column indicates the number of options in the choice set. The approximations are marked as in the rest of the chapter.

Opt.	CS		CI		ME-K		ME-R		SP		TF		TS	
3	4.7	(0.1)	16.1	(0.5)	12.5	(0.3)	13.4	(1.0)	6.8	(0.2)	12.8	(0.4)	13.9	(0.3)
6	7.5	(0.2)	35.1	(0.7)	53.9	(0.6)	52.8	(1.9)	32.5	(2.8)	54.0	(0.6)	57.5	(0.6)
9	10.2	(0.2)	73.0	(1.5)	211.1	(2.2)	197.0	(12.6)	230.3	(11.6)	206.1	(2.1)	213.7	(2.0)
12	13.7	(0.4)	133.9	(3.2)	601.6	(6.6)	550.3	(46.7)	1740.1	(41.8)	585.1	(4.7)	599.2	(4.7)
15	17.9	(0.6)	227.6	(6.2)	1387.6	(14.6)	1264.7	(115.2)	13920.9	(291.1)	1355.0	(11.8)	1377.2	(11.9)

Opt.	CRD(opt)		GEN(non-opt)		GEN(opt)		SJ		SC		J1	
3	17.3	(2.1)	17.3	(2.1)	17.3	(2.1)	17.3	(2.1)	17.3	(2.1)	17.3	(2.1)
6	200.4	(11.1)	188.7	(9.2)	265.8	(17.6)	3012.2	(187.3)	254.6	(13.8)	$7.5 \cdot 10^4$	$(1.5 \cdot 10^4)$
9	869.9	(39.2)	809.0	(26.8)	1192.8	(65.1)	$2.3 \cdot 10^6$	$(4.1 \cdot 10^4)$	773.7	(46.1)	$2.7 \cdot 10^6$	$(2.5 \cdot 10^5)$
12	2379.4	(43.4)	2198.4	(57.5)	3242.8	(72.1)	$2.7 \cdot 10^5$	$(1.4 \cdot 10^3)$	1804.2	(35.9)	$1.7 \cdot 10^7$	$(1.1 \cdot 10^6)$
15	5346.3	(87.9)	5044.6	(80.0)	7232.8	(148.5)	$6.0 \cdot 10^5$	$(1.7 \cdot 10^3)$	3781.5	(78.3)	$8.6 \cdot 10^7$	$(4.7 \cdot 10^6)$

Table 3.45 – Average calculation times and standard deviations (in brackets) required by the MNP approximations for the first and second series of experiments for different sizes of the choice set. Times in seconds* 10^{-5} . Experiments carried out on a Pentium II 350 MHz desktop computer. The first column indicates the number of options in the choice set. The approximations are marked as in the rest of the chapter.

Opt.	Numerical Integration	
3	0.028	(0.006)
6	0.090	(0.026)
9	0.999	(0.538)
12	14.582	(7.140)
15	552.812	(571.667)

Table 3.46 – Average and standard deviation (in brackets) of the calculation times for the numerical integration method of Genz with a required precision for the probabilities of $5 \cdot 10^{-5}$. Times in seconds. Experiments carried out on a Pentium II 350 MHz desktop computer. The first column indicates the number of options in the choice set.

3.7 Issues in the Practical Application of the MNP Choice Model

3.7.1 Semidefinite Covariance Matrices

The presentation of the MNP problem and of the related MVN integral at the beginning of this chapter assumed that the MVN distribution of the utilities has positive definite covariance matrix (to allow its inversion in (2.7) and thus in (3.2), (3.5) and (3.8)). In MNP calibration the covariance matrices are restricted to be positive definite. However, in traffic assignment it is possible to encounter cases in which the covariance matrix of the utilities is positive semidefinite and so are the covariance matrices of the MVN distributions in difference.

Consider, for instance, the network depicted in fig. 3.101: next to each link is the variance of the Normal distribution of the link cost. Six paths can be enumerated between point A and point B as sketched in figure 3.102, that also gives the codes of the paths.

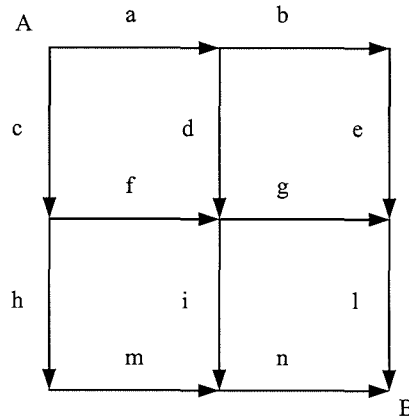


Fig. 3.101 – Simple network with 6 paths with semidefinite covariance matrix of the distribution of the path costs.

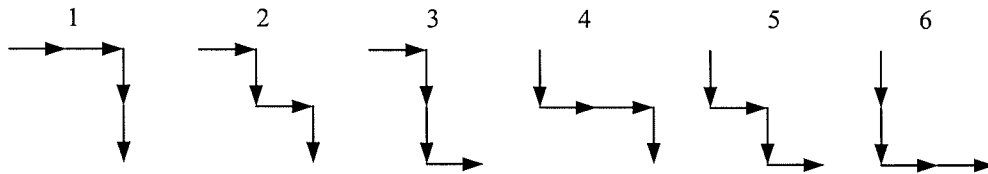


Fig. 3.102 – Schematic representation of the 6 possible paths between A and B in figure 3.101.

If the path costs are considered additive, the covariance matrix of their MVN distribution results in:

$$\begin{pmatrix} a+b+e+l & a+l & a & l & 0 & 0 \\ a+l & a+d+g+l & a+d & g+l & 0 & 0 \\ a & a+d & a+d+i+n & 0 & i+n & n \\ l & g+l & 0 & c+f+g+l & c+f & c \\ 0 & 0 & i+n & c+f & c+f+i+n & c+n \\ 0 & 0 & n & c & c+n & c+h+m+n \end{pmatrix} \quad (3.100)$$

(where line and column 1 correspond to path 1 and so on).

It can be seen that line (and column) 4 of this matrix can be expressed as a linear combination of lines 2, 5 and 3: line 4 is obtained summing lines 2 and 5 and subtracting line 3. Therefore, one of the eigenvalues of the matrix is zero and the matrix is positive semidefinite, rather than positive definite. In other words, the rank of the matrix is smaller than its dimension.

The networks used to test the precision of the approximations in the previous part of this chapter had positive definite covariance matrices but positive semidefinite covariance matrices may occur in real networks.

It is important to note that the presence of a positive semidefinite covariance matrix does not affect the calculations of the approximations of Clark, of Mendell-Elston, of the separated split method of Langdon, and of those of the FOMN family that have been examined. However, the covariance matrices need to be positive definite to solve the MNP problem with the numerical integration method of Genz and with the approximations based on the Taylor series expansion. Moreover, Joe in the code for the Solow-Joe and extended Joe approximations remarks that semidefinite matrices may cause problems.

In general, the a MNP choice problem with semidefinite covariance matrix can be reduced to an equivalent one with full rank covariance matrix as suggested for instance in Pattison and Gossink (1999), who recalled that, by definition, a Normal variate \mathbf{X} with covariance matrix with rank r smaller than its dimension n is related to at least a Normal variate \mathbf{Y} of dimension r and with full rank covariance matrix with a transformation like:

$$\mathbf{X} = \mathbf{A}\mathbf{Y} + \mathbf{b} \tag{3.101}$$

where \mathbf{A} is a $[n \times r]$ matrix. This transformation can be used to obtain the image of the original variate in the space generated by the eigenvectors of its covariance matrix corresponding to non-zero eigenvalues. The limits of integration for \mathbf{Y} can be found using again the transformation (3.101).

Alternatively, heuristic methods can be used to eliminate the issue. For instance, Clark and Watling (2002) in a work on sensitivity analysis of SUE probit traffic assignment models avoided the issue in an approximate way by excluding from the path sets the path or paths causing the semidefiniteness of the covariance matrix. Such paths were chosen so that the remaining paths carry most of the traffic between the relevant OD pair.

However, an exact method to solve the problem with the numerical integration method of Genz has been proposed by Genz and Kwong (2000). This method consists of reducing the dimension of the covariance matrix by using the generalised Cholesky decomposition carried out by the algorithm of Healy (1968) and rewriting the integration limits. The generalised Cholesky decomposition of Haley allows us to decompose singular matrices and produces triangular matrices with columns of zeros corresponding to the null eigenvalues, thus eliminating the redundant variates from the MVN integral expressed as the combination of a number of univariate Normal distributions. The integration limits are then redefined in the space of the necessary univariate Normal distributions, proceeding then to write the redefined MVN integration problem so that it results in an integration problem over a unit hypercube, following the method of Genz (1992, 1993) described in section 3.4.

3.7.2 The Evaluation of the Satisfaction Function

3.7.2.1 Introduction

An aggregate measure of the utility of a set of options is given by the mean of the distribution of the maximum of their utilities. This is referred to as satisfaction (see *e.g.* Daganzo, 1979; Sheffi, 1985; Ben Akiva and Lerman, 1985) and can be used in choice models applied to traffic assignment as a measure of the cost to reach a destination. It is part of the equivalent objective function that allows us to solve for the Stochastic User Equilibrium traffic pattern on a network, as explained in detail in chapter 4, and of the objective function and the elastic demand functions (that determine the number of trips made between each origin and destination on a network) used in the elastic demand models considered in chapter 5.

The satisfaction, S , due to the J options in a choice set is:

$$S = E\left[\max_j(U_j)\right] \quad (3.102)$$

In traffic assignment models, where disutilities or costs rather than utilities are used, the satisfaction is, correspondingly, the expected minimum value of the costs of the options in a choice set (the J paths between an OD pair):

$$S = E[\min_j(C_j)] \quad (3.103)$$

The satisfaction has a relative rather than an absolute meaning, unless the scale of the model is somehow fixed, since it is based on the utilities, that are scalable. Sheffi (1985) reports a number of properties of the satisfaction function.

In the multinomial probit model case (3.102), similarly to the choice function, cannot be written in closed form and needs to be evaluated approximately (Daganzo, 1979; Sheffi, 1985).

There is limited literature on the evaluation of the satisfaction function with the probit choice model. No method relying on numerical integration has been suggested and simulation is not normally used. Two of the approximation methods seen above allow the direct, though approximate, calculation of the satisfaction. They can also be used to form hybrid algorithms as outlined below. A further method to approximate analytically the satisfaction function is also suggested.

3.7.2.2 Evaluation of the Satisfaction Using the Approximation of Clark

The method of Clark can be applied recursively to calculate the maximum of a number of Normal (or approximated as Normal) variates as explained in section 3.5.2. It was also noted that the simple application of the Clark approximation one more time than strictly needed for the calculation of the choice probabilities allows us to obtain the value of the satisfaction with either the simple Clark method or the improved Clark method.

With the simple Clark method one value of the satisfaction is obtained directly for the calculations, whilst with the improved Clark method a number of satisfaction evaluations equal to the number of options can be obtained, and possibly the average value can be used.

The simplicity of the Clark approximation and its very high computational efficiency suggests the use of hybrid algorithms to calculate the satisfaction with MNP approximations that do not calculate it directly: the choice probabilities can be calculated with the relevant approximation and the satisfaction with the method of Clark.

3.7.2.3 Evaluation of the Satisfaction Using the Approximation of Langdon

The method suggested by Langdon (1981) for the calculation of the satisfaction is based on his approximation for solving the MNP integral. It relies on the fact that, to obtain the choice probabilities, the separated split approximation calculates the means, variances and covariances of the reduced utility distributions (the utilities of the different options as perceived by the decision makers who have made a certain choice, in the terminology used by Langdon).

The satisfaction for each set of options can be obtained by carrying out one more reduced utility distribution calculation after those necessary for the choice probabilities, thus computing the means of the reduced utility distributions of each option for those who have chosen it. Summing the mean of the utility of each finally chosen option as perceived by those who chose it, weighted by its choice probability, the satisfaction of the set of options is obtained.

Langdon (1981) underlined that such satisfaction value is obtained without assumptions on the distribution of the maximum of the utilities. In fact, as in the case of the probability calculations, the method assumes that the reduced utility distributions are Normal.

Also in this case, hybrid algorithms can be proposed *e.g.* for the Mendell-Elston and the Taylor series approximations to carry out, in parallel to the approximation's probability calculations, the calculations for obtaining the utility of each option for those who chose it, as in the approximation of Langdon. The calculation of the reduced utility proceeds similarly to that in the original approximation along one of the outer branches of the pictorial example of computational structure in figure 3.1. The satisfaction is obtained summing each of the final reduced utilities weighted by the relevant choice probability obtained with the approximation used.

3.7.2.4 Evaluation of the Satisfaction Using the Normal Approximation

A further method to obtain the value of the satisfaction, which may be used with any MNP choice function evaluation method, can be developed by assuming that the distribution of the maximum of the utilities can be approximated as Normal. This is

similar to the assumption made by Clark but only in the assumption on the distribution of the total maximum.

The approximate values of the mean and of the variance of the maximum distribution may be calculated considering, separately, two dummy choice options and performing the relevant two additional choice calculations.

First, a dummy alternative, not correlated with the real ones, with any mean and variance, is considered along with the real ones and its choice option is calculated with the relevant approximation. Then a second uncorrelated dummy alternative, with a mean different from that of the first dummy alternative but with the same variance, is considered along with the real ones and its choice probability is calculated.

The mean and variance of the maximum of the utilities, assumed to be Normal, can be obtained by considering that the choice probability of each of the dummy alternatives is the same as the probability that they are chosen over the maximum of all the real ones. Thus, the approximate mean and variance of the maximum can be calculated considering that the dummy alternatives are not correlated with it (since they are not correlated with any of the other real alternatives) and considering two binary choices between the maximum of all the real alternatives and each of the dummy ones.

If V_a , V_b and σ^2 are respectively the utilities of the dummy alternative a , of the dummy alternative b and their common variance, their separate probabilities of choice over the real options are P_a and P_b and can be written as:

$$P_a = P(U_a > U_{\max}) = \Phi \left(\frac{V_a - V_{\max}}{\sqrt{\sigma^2 + \sigma_{\max}^2}} \right) \quad (3.104)$$

$$P_b = P(U_b > U_{\max}) = \Phi \left(\frac{V_b - V_{\max}}{\sqrt{\sigma^2 + \sigma_{\max}^2}} \right) \quad (3.105)$$

Therefore it is also possible to write:

$$\frac{V_a - V_{\max}}{\sqrt{\sigma^2 + \sigma_{\max}^2}} = \Phi^{-1}(P_a) = k_a \quad (3.106)$$

$$\frac{V_b - V_{\max}}{\sqrt{\sigma^2 + \sigma_{\max}^2}} = \Phi^{-1}(P_b) = k_b \quad (3.107)$$

The system of equations (3.106) and (3.107) can be solved to give:

$$\sigma_{\max}^2 = \left(\frac{V_a - V_b}{k_a - k_b} \right)^2 - \sigma^2 \quad (3.108)$$

$$V_{\max} = V_a - \left(\sqrt{\sigma^2 + \sigma_{\max}^2} \right) k_a \quad (3.109)$$

Although it is rather simple, this method may be rather time consuming as it introduces two additional choice calculations of one dimension larger than those in the original choice problem.

3.8 Conclusions

This chapter has discussed the problem of the evaluation of the multinomial probit choice function. A number of different methods have been described, dividing them into methods relying on simulation, methods using numerical integration and methods using analytical approximations. The analytical approximation methods that have recently received limited attention in the transportation literature have been described and examined in more detail. The ground breaking numerical integration method of Genz (1992, 1993), that allows us to calculate MVN integrals of large dimension in relatively limited computing times, has been used to obtain the reference results for testing the accuracy of most of the approximation methods described. Experiments have been carried out with the approximation of Clark (in the heuristic simple Clark implementation and in the exact improved Clark method), the approximation of Mendell-Elston, the separated split method of Langdon, the crude and generalised FOMN methods, the approximation derived from first and second order Taylor series expansions, the Solow-Joe approximation and the extended approximation of Joe. These approximations have been developed in different disciplines and, in particular, the FOMN methods, the Taylor

approximations, the Solow-Joe and the extended Joe approximation, as well as the numerical integration method of Genz, had been applied previously only for the evaluation of MVN integrals and were introduced here in the calculation of the MNP choice function. The accuracy tests have been carried out on artificial test networks intended to replicate choice situations encountered in traffic assignment. Besides allowing us to compare the quality of the results of the approximations and their typical inaccuracies, such tests have also allowed the collection of information on the computational time they require.

An analysis of the possible inaccuracies given by each approximation as well as an overall comparison of the methods based on accuracy and computational cost have been presented. The analysis of the results of each approximation showed that they all tend to give larger percentage error in the evaluation of small choice probabilities and underlined the importance of using optimal processing orders of the variates within the approximation calculations to maximise the accuracy of the results. In particular, heuristic optimal processing orders were proposed and tested for the FOMN methods examined and for the Solow-Joe approximation and a further heuristic processing order, suggested by a remark in the literature, has been tested for the Mendell-Elston approximation.

The approximation of Mendell-Elston (with the processing order suggested by Kamakura and with the heuristic processing order of Rice *et al.*), the generalised FOMN approximation (with heuristic optimal processing order) and the extended approximation of Joe resulted in the most precise methods amongst those examined. The extended Joe method, however, showed long calculation times due to calculating up to four-variate integrals and to the averaging device suggested by its author to obviate the inaccuracies due to variate ordering. The generalised FOMN method with optimal processing order is the most precise method amongst those examined but required calculation times that were about five times those required by the Mendell-Elston method which, thanks to its good precision and limited calculation time, is the method offering the best trade-off between precision and computational time.

The approximation of Clark, in either of its implementations, resulted in the fastest but the least precise of the methods, whilst the other methods are more precise than

the Clark ones but less than those mentioned above and are, generally, of satisfactory precision except for small choice probabilities.

The results obtained can be summarised in the following table 3.47 that reports a qualitative judgement on the precision and the calculation time of the approximations examined.

<i>Approximation</i>	<i>Accuracy</i>	<i>Calc. Time</i>
Generalised FOMN (heuristic optimised order)	Very good	Medium
Mendell-Elston (heuristic Rice <i>et al.</i> order)	Very good	Low
Mendell-Elston (Kamakura order)	Very good	Low
Extended Joe (complete averaging)	Very good	Very high
Taylor series (optimised order)	Medium	Low
Langdon (optimised order)	Medium	Medium-low
Solow-Joe (heuristic order)	Medium	Medium-low
Solow-Joe (complete averaging)	Medium	High
Crude FOMN (heuristic optimised order)	Medium-low	Medium
Improved Clark (non-optimised order)	Low	Very low
Simple Clark (non-optimised order)	Very low	Very low

Table 3.47 – Qualitative summary of precision and calculation time results for the MNP approximations examined. The methods are listed following an approximate decreasing order of accuracy. The Mendell-Elston is the method giving the best trade-off between accuracy and calculation time.

Two issues in the application of MNP solution methods to transportation problems have also been mentioned: the possibility that the utilities have a semidefinite covariance matrix and the evaluation of the satisfaction function, the expected maximum utility of a choice set. On the first subject it has been remarked that the calculations with most approximations examined are not affected by positive semidefinite covariance matrices and that for the numerical integration method of Genz a solution method has been proposed in the literature. On the second subject, it has been recalled that the satisfaction can be evaluated directly with the MNP approximations of Clark and of Langdon, and it has been suggested that these can be used to implement hybrid algorithms when the choice probabilities are calculated with other methods. A further approximate method to calculate the satisfaction has been suggested.

4. MULTINOMIAL PROBIT STOCHASTIC USER EQUILIBRIUM PATH-BASED TRAFFIC ASSIGNMENT

4.1 Introduction

This chapter is concerned with the solution of the Stochastic User Equilibrium (SUE) traffic assignment problem using two of the analytical methods for the solution of the multinomial probit (MNP) choice model discussed in chapter 3 but considering methods that can be employed also with other approximations or choice models solved analytically.

The chapter opens with a brief introduction to the traffic assignment problem, followed by a note on the specification of the path sets and a review of link-based and path-based models and algorithms for User Equilibrium (UE), Stochastic Network Loading (SNL) and Stochastic User Equilibrium.

The core of the chapter deals with the solution of the multinomial probit path-based Stochastic User Equilibrium problem when the path sets are fixed at the outset of the assignment, explicit account of both link and path flows is taken and the equivalent program due to Sheffi and Powell (1982) is used. The efficiency of solution algorithms with different line search methods and search directions is compared using the multinomial probit approximations of Mendell and Elston (1974) and of Clark (1961) in the implementation described in chapter 3 as improved Clark.

4.2 The Traffic Assignment Problem

4.2.1 Generalities

The traffic assignment problem consists of calculating the traffic pattern resulting from the route choices made by a known population of drivers travelling between their origins and destinations (OD) on a road network of known characteristics. As

results from this definition and as mentioned in chapter 2, traffic assignment is an applied choice problem.

The traffic assignment problem has been widely studied in the literature and can be posed in several different ways depending on the theoretical framework used. The models and the related algorithms proposed in the literature to solve it differ on several crucial points, amongst which:

- the choice model employed;
- the way routes through the network are considered;
- the set of costs used;
- whether the interaction between drivers is considered;
- whether the system is modelled in a static situation representative of a period of time or the evolution of the system during a period of time is explicitly represented.

Following the framework set out in chapter 2, a description of traffic assignment models as choice models should start from the specification stage (definition of the theoretical choice framework, of the choice set and of the actual functional specification of the model).

The models discussed here and used in practice for traffic assignment deal with route choices within the random utility framework outlined in section 2.2. Drivers are assumed to be utility maximising decision makers. In assignment, the choice problem is posed in the equivalent form of disutility (cost) minimisation, rather than in terms of utility maximisation, since the costs of travelling along alternative routes between the same origin destination pair are considered.

The choice set is formed by paths between the origin-destination (OD) pairs in the network. It is generally not feasible to characterise and consider in the calculations all such paths, even only all the elementary paths, which are those without loops,

between each OD pair. Moreover, not all the possible paths between an OD pair are necessarily relevant choice options for the travellers. Traffic assignment models differ on the actual sets of paths included in the choice sets and on how they are represented. Paths can be considered in the models (and represented in the solution algorithms) either directly, using path related data, or implicitly, using only link data. Models of the first sort are known as **path-based** whilst the others are referred to as **link-based**. The situations modelled on a network, like User Equilibrium and Stochastic User Equilibrium, are defined referring directly to the paths between the OD pairs as these are the actual choice options. However, algorithmic considerations, including difficulties of enumerating suitable path sets and storing and manipulating them in the computer memory, traditionally led researchers to prefer link-based models both in research and in applications. Recently, some of the issues related to the use of path-based models have been partially overcome and path based models are increasingly being used.

Route disutilities or costs are derived considering the criteria on which, as it results from empirical studies, drivers choose their routes such as, for instance, minimising expected journey time and route length (see *e.g.* Ben-Akiva *et al.*, 1984; Thomas, 1991; Ortúzar and Willumsen, 1994, for lists of possible criteria). Such criteria are taken into account using suitable quantitative descriptions that are transformed into homogeneous units and combined into a single measure of travel disutility known as generalised cost, which usually is a weighted sum of the most important factors only (journey time and cost, for instance).

Route costs are commonly assumed to be additive, *i.e.* the cost of a route is the sum of the costs of the links that it traverses. Also link costs are often assumed to be separable, that is the cost on a link is a function of the traffic on that link only. The first assumption needs to be relaxed when path-related costs are taken into account which, except in special cases, can be done only using explicit path treatment (see Gabriel and Bernstein, 1997). The assumption of separable link costs is relaxed when *e.g.* explicit account of costs at intersections is taken, using either formulations

of the problem able to include it or algorithmic modifications as the heuristic diagonalisation algorithm (see *e.g.* Sheffi, 1985).

The specification of a particular **choice model** within the random utility framework entails assumptions about how users perceive route disutilities, consistently with the discussion on choice models in chapter 2.

Using the **deterministic model** implies that users make choices on the systematic value of the route disutilities as specified by the analyst, therefore that they have perfect information on network conditions and at the same time that the analyst has fully accounted for all the factors affecting route choice in the generalised cost expression.

With **stochastic route choice models**, drivers' decisions are modelled as made on perceived travel costs including the random effects mentioned in section 2.2, rather than on their systematic value only. However, different choice models will have different abilities to capture the topology of the network and therefore different suitability for traffic assignment, as noted in chapter 2.

The costs on the network, however defined, can be considered fixed or assumed to vary with the traffic.

When the **costs on the network** are considered **constant** independently of the traffic on it and the deterministic choice model is used, all the traffic between each OD pair is assigned to one path only (the path with the minimum deterministic disutility) and the resulting model is therefore known as “**all or nothing**” (AoN). Models considering constant costs but stochastic choice split the traffic amongst a number of paths between each OD pair and are referred to as **Stochastic Network Loading** (SNL) models.

However, considering constant network costs is not realistic, as in reality the limited capacity of the elements of a network makes their costs vary with their usage, causing the interaction between the choices known as congestion. In the models considered here, this phenomenon is accounted for in the **static equilibrium** framework, which entails assuming that, over the analysed period, the costs on the network and the traffic pattern are consistent. Nevertheless, network loading models remain important since, rather than on their own, they are used as building blocks for the algorithms calculating the equilibrium flows.

Accounting for congestion within the static equilibrium framework implies modelling a single representative situation of the traffic system during a representative period of time, usually the peak hour or the peak period. Alternatively the congestion can be accounted for following its evolution during a given time period with **dynamic models**.

When the deterministic choice model is used within the static equilibrium framework the resulting assignment model is known as **User Equilibrium (UE)**. **Stochastic User Equilibrium (SUE)** models result when a stochastic discrete choice model is employed with the static equilibrium framework. Using different stochastic choice models gives different traffic patterns and thus different traffic assignment models.

The User Equilibrium condition over a network is described by the first principle of Wardrop (1952):

no user can improve his journey time by unilaterally changing route

which is equivalent to saying:

the journey times on all routes used are equal, and less than or equal to those which would be experienced by a single vehicle on any unused route.

The definition of the SUE concept has been given by Daganzo and Sheffi (1977):

at SUE no user can improve his *perceived* travel cost by unilaterally changing route.

SUE can be seen as a generalisation of UE and, dually, UE can be seen as a particular case of SUE.

UE models are often used on the grounds that, in practice, the networks usually modelled are congested and in such conditions the results given by a SUE model should be very close to those of the simpler UE one (see *e.g.* Sheffi and Powell, 1981), as the systematic component of the path costs prevails over the stochastic effects. However, it can be argued that a SUE model is more flexible and, since congestion on networks is not necessarily uniform, SUE models can deal correctly with areas of a network with different levels of congestion, accounting mainly for the deterministic effects where these prevail and including stochastic effects when they are important. Thus, in general, SUE models should be preferred.

Traffic assignment models differ also on the mathematical formulation that determines which effects can be accommodated and which solution algorithms can be used. The traffic assignment problems considered in this thesis are analysed with equivalent mathematical programmes, but in the literature they have also been expressed *e.g.* as fixed point problems or using variational inequalities (see respectively *e.g.* Daganzo, 1983, and Smith, 1979)

A discussion of the sets of paths actually accounted for in link and path-based models in the literature is reported in the next section and is followed by a review of models and algorithms for UE, SNL and SUE.

4.2.2 A Review of Traffic Assignment Models in the Literature

4.2.2.1 A Discussion on the Specification of Path Sets

Before giving a survey of traffic assignment models it is interesting to consider separately the path specification issue. As seen above, the definitions of the UE and SUE conditions are expressed in terms of paths but it is practically infeasible, and

very likely not useful, to enumerate all possible paths on a real network. In path-based models, path enumeration techniques are employed to obtain limited but relevant and sufficient sets of paths. In link-based models there is often less emphasis on the implicit rule that defines which paths are actually considered but different models have different enumeration rules. However, both in path-based and link-based models, the matter is equally important and delicate as the results of the models depend on the paths considered.

Until recently, implicit representation of the paths has been preferred for both deterministic and stochastic models. This was due to the problems brought about by dealing directly with paths on a network of realistic size and complexity:

- the number of paths between each OD pair increases very rapidly with the size of the network, even when only the elementary paths are considered, and their complete enumeration is possible only on small networks. This can be seen for instance considering the formula to obtain the maximum number of paths between each pair of nodes as a function of the number of nodes in the network and the number of links building the paths reported by Huang and Bell (1998);
- typically, many of the possible paths between an OD pair are not actually relevant to the traffic travelling between that OD pair: if enumerating all the possible paths were practically possible, many of the routes that would result would probably be too involved or illogical;
- path data storage and manipulation require more computer memory and time than link data, and large networks path data may require very large quantities of them.

Over the last ten years much interest and research effort has been concentrated on path-based models because of the possibility to obtain directly additional traffic data (as *e.g.* path-related information, turning movements) and because of possible computational advantages (see *e.g.* Jayakrishnan *et al.* 1994).

Moreover, the issues related to path-based algorithms listed above have become less stringent. Techniques to enumerate a limited but relevant and sufficient number of paths between each OD pair have been developed and tested, although research in

this particular area is still on-going. On computer memory grounds, path-based algorithms are nowadays employable on medium sized networks thanks to less expensive computer RAM and to devices such as virtual memory. The practical feasibility of these methods for algorithms designed to work on desktop computers is shown, for instance, by the introduction of path based methods in the commercial software SATURN (see Kupiszewska and Van Vliet, 1999).

It is worth mentioning in more detail which are the methods to explicitly enumerate paths currently in the literature although, in some cases, it is difficult to separate such techniques from the assignment algorithms in which they are implemented. Path enumeration for stochastic loading is carried out prior to the loading whilst for equilibrium traffic assignment it can be carried out before or during the assignment.

Path enumeration carried out during the assignment calculations is referred to as column generation and consists of running a shortest path algorithm at each iteration of the traffic assignment algorithm on the current set of generalised costs or augmented link costs (the latter used to account for constraints as in the PFE of Bell *et al.*, 1997, or to account for the entropic part of the objective function as in the Frank-Wolfe logit SUE model of Bell *et al.*, 1993). The path thus characterised is added to the current set of paths, if it is new, and considered by the choice model to assign the traffic.

Several algorithms have been considered in the literature to define prior to the traffic assignment a set of paths that is kept fixed during the assignment calculations. Examples of this sort of algorithm are the label approach of Ben-Akiva *et al.* (1984) using a simple shortest path algorithm, the essentially least cost path algorithm of Hunt and Kornhauser (1996), the heuristics of Huang (1995) to filter the paths resulting from a STOCH enumeration (Dial, 1971), the method of Huang and Bell (1998) to enumerate elementary paths excluding those using nodes not visited by UE flows (to reduce their total number), the reasonable path algorithms of Park and Rilett (1997) that can include the heuristic of De La Barra *et al.* (1993), the exact k-shortest path algorithms with filtering of the results to avoid excessive overlapping of paths (Cascetta *et al.*, 1997), the STOCH3 rule proposed by Leurent (1997a,b) to enhance and stabilise the STOCH rules of Dial, the algorithm for non-additive path

costs of Scott and Bernstein (1997), the methods proposed by Dial (1996) and by Leurent (1994, 1995, 1996b) to enumerate paths when the value of time of the users is a random variable, and the paths that can be obtained using a shortest path algorithm on costs sampled from their distributions as in the assignment algorithms of Burrell (1968) and of Sheffi and Powell (1981).

Each of these algorithms is defined on a particular cost set: generalised cost (free flow, randomly drawn, UE, current during an assignment) or special costs to characterise individually different criteria that define route choice (*e.g.* minimum travel time or minimum distance) as in the label approaches of Ben-Akiva *et al.* (1984) and of Cascetta *et al.* (1997).

The variety of techniques to enumerate sets of paths suggested in the literature and the ongoing research to test such algorithms might suggest that it is still convenient to use link-based algorithms. However, it is important to remark that there are differences and limitations also amongst implicit path enumeration techniques used with well known link-based models.

Few link-based models consider all the possible paths (including those with loops) between each OD pair and leave it to the choice model to define which are actually relevant: the logit models of Bell (1995) and of Akamatsu (1996) and the probit SAM model of Maher and Hughes (1997a). The other link-based models in some cases consider implicitly all the elementary paths (*i.e.* the paths without loops) between the OD pairs (as in Burrell-like SNL and SUE models or in link-based UE models) whilst in other cases only subsets of the elementary paths between the OD pairs are considered. This is the case, for instance, in the STOCH model of Dial (1971). In particular the definition of “efficient paths” put forward by Dial (1971) limits the set of possible paths using a rule with a behavioural meaning but is devised to allow the algorithm to work and can be quite restrictive, as remarked by Maher (1992).

4.2.2.2 Link-Based Models and Algorithms for UE

Heuristic methods such as the hard speed change method, the soft speed change method and the incremental loading method were initially used to model the UE traffic pattern (see *e.g.* Thomas, 1991). Those methods were ineffective and the exact calculation of the UE link flow pattern over a complex network became possible after Beckmann *et al.* (1956) posed the UE problem as the following mathematical program whose objective function is minimised by the UE link flows:

$$z_{UE} = \sum_I \int_0^{x_i} c_i(u) du \quad (4.1a)$$

subject to:

$$\sum_p f_{rsp} = q_{rs} \quad \forall rs \quad (4.1b)$$

$$f_{rsp} \geq 0 \quad \forall rs, p \quad (4.1c)$$

and with:

$$x_i = \sum_{rsp} f_{rsp} \delta_{ip}^{rs} \quad \forall i \quad (4.1d)$$

where c_i is the cost of travelling along link i which is assumed separable and therefore depends only on the flow x_i on the same link, q_{rs} is the flow between the OD pair rs and f_{rsp} is the flow along p , one of the paths between the OD pair rs . Finally, δ_{ip}^{rs} is 1 when link i is part of path p between the OD pair rs and zero otherwise.

The solution to the program of Beckmann *et al.* (1956) can be efficiently found with link-based techniques employing the Frank-Wolfe algorithm (Frank and Wolfe, 1956), also known as the convex combination algorithm, or one of its variations.

The Frank Wolfe algorithm has the basic structure of many optimisation algorithms: it starts from a feasible set of flows and progresses towards the solution by iterations. At each iteration the current solution is updated by moving by a suitable step length along a descent direction given by the vector linking the current and the auxiliary

solution. The latter is the feasible flow vector that minimises the linearisation of the programme's objective function at the current solution point and, in the case of the UE programme, it is the result of an All or Nothing loading on the network with the current costs. The search vector also bounds the step length whose actual value is found minimising the programme's objective function along the search direction.

The UE point is described by a set of unique link flows but the UE path flows are not unique (see *e.g.* Sheffi, 1985; Bell and Iida, 1997).

4.2.2.3 Path-Based Models and Algorithms for UE

Path-based models for UE have been developed mainly in the last ten to fifteen years and implement a number of different algorithms, many of which are more efficient than the link-based Frank and Wolfe, which becomes slow near the optimum (see *e.g.* Larsson and Patriksson, 1992, Sun *et al.*, 1996).

Schittenhelm (1990) proposed the "equilization" algorithm based on the iterative direct check of the UE conditions for all the elementary paths between each OD pair which are enumerated by means of column generation. Larsson and Patriksson (1992) solved for the UE flows using column generation and the disaggregate simplicial decomposition algorithm. Their approach was further considered by Hicks and Ham (1997) who studied its application to large networks. Jayakrishnan *et al.* (1994) and Sun *et al.* (1996) proposed using column generation and a gradient projection algorithm (with different variants) for real time applications. Kupiszewska and Van Vliet (1998, 1999) put forward a path-based improvement of the Frank-Wolfe algorithm, the Social Pressure algorithm, which is to be used in the commercial software package SATURN (Van Vliet and Hall, 1993). Gabriel and Bernstein (1997) considered path-based UE for the first time with non-additive path costs (*i.e.* path costs that are not simply the sum of the costs of the links traversed by the path), addressed the problem with a non-linear complementarity formulation and put forward a solution algorithm involving column generation. Bell (Bell and Iida, 1997, pp. 99-102) noted how, knowing the set of used paths and using the result of a link-based assignment, the most likely path flows can be obtained using a maximum entropy formulation.

4.2.2.4 Link-Based Models and Algorithms for SNL and SUE

Several link-based logit loading methods assuming fixed path costs and differing for the set of paths implicitly considered have been put forward in the literature. The best known logit loading method is due to Dial (1971) and includes only paths complying with his algorithmically effective, but quite restrictive, definition of “efficient paths”. Efficient paths, as defined by Dial, are either those including only links taking us further away from the origin and closer to the destination (according to the rule named STOCH1) or, in a less restrictive definition, those including only links taking us away from the origin (rule STOCH2). The algorithm, in a forward pass moving away from the considered origin calculates the link “weights” which contain all the information on upstream efficient links that are necessary for the assignment. Traffic destined to or leaving each node is assigned in a backward pass proportionally to the weights calculated for each efficient link entering each node.

More recently new logit link-based loading techniques have been put forward by Bell (1995) and Akamatsu (1996) whose models consider all the paths between each OD pair, including also those with infinite loops, and Maher and Hughes (1996b), who proposed methods to account implicitly for paths without loops, with loops of some order or with infinite loops.

The logit choice model is possibly the most widely used in traffic assignment because it is convenient and simple to use and because of the availability and efficiency of its implementations although it has been known since the 1970s (see *e.g.* Florian and Fox, 1976; Sheffi and Daganzo, 1977) that it is not entirely suitable for traffic assignment. In fact, as mentioned in chapter 2, assuming that the random parts of the path costs are independently and identically distributed results, respectively, in not taking account of route overlapping and in route choices made only on absolute differences of costs thus causing biased loading.

Stochastic loading methods with logit model “extensions” proposed in the literature are path-based and are described in the next section.

There are, however, link-based algorithms for the probit model which, although traditionally less used than the logit, has a sound theoretical basis in the Central

Limit Theorem (see Daganzo and Sheffi, 1977) and is suitable to model route choice since route overlapping and non-identically distributed random parts of the path costs are considered through the path-cost covariance matrix. In particular, when path costs are considered additive, the variance of the cost of a path is equal to the sum of the variances of the costs of the links that it traverses (since link costs are assumed to be independent) and the covariance between the costs of two paths is the sum of the variances of the costs of the common links.

As mentioned in chapter 2 the most widely used method to solve the probit model is simulation. Probit SNL link-based simulation models have been put forward by Burrell (1968) (in an approximate implementation), Sheffi and Powell (1981) and Nielsen (1997), all considering implicitly all the elementary paths between each OD pair.

These methods perform a SNL by drawing the costs of the network's links from their distribution a number of times and each time loading the demand between each OD pair on the resulting shortest path. The SNL is then obtained by averaging the flows obtained over the number of draws, thus performing something similar to a crude frequency simulation algorithm. However, such a complete SNL is not carried out when calculating SUE, for which a "streamlined" procedure, described later in this chapter, has been devised by Sheffi and Powell (1981).

Probit loading models using the approximation of Clark (1961), rather than simulation, have been put forward by Daganzo and Sheffi (1977), whose model was not included in a practical algorithm, and Maher (1992), whose SAM model can account implicitly for paths without loops, with loops of some order or with infinite loops.

The Stochastic Assignment Model (SAM) of Maher has been developed using the Clark approximation to the MNP model, but it is not strictly based on it. In fact, it can embed any approximation to the MNP able to solve the choice function and give the value of the satisfaction function for a choice set. SAM loads the network in two stages. In a "forward step", carried out for each origin in turn, the distribution of the costs to reach each node in the network and the expected value of its minimum (the satisfaction) are calculated along with the probability that the flow reaching a node

will travel along each entering link, as it results from the distribution of the costs. In a “backward step”, for each origin in turn, the nodes are processed in an order opposite to that followed in the forward step, moving from the one furthest away from the origin considered, towards such origin, and the flow destined to each node is calculated (summing the flow actually destined to the node and those leaving it through the exit links) and loaded on the entering links according to the splits obtained in the forward step. An important feature of the model is how it resolves “deadlocks”. These occur in the forward step when the distribution of the costs reaching each node cannot be obtained because of cycles in the networks. Four ways to deal with the issue have been put forward, ranging from a method eliminating all cycles according to the technique proposed by Dial (1971) for logit loading to a method to include infinite cycles, and are treated in detail in Hughes (1998).

Two mathematical programs to calculate SUE flows have long been established in the literature. Fisk (1980) devised the following objective function for logit based SUE:

$$z_{SUE \text{ logit}} = \frac{1}{\theta} \sum_{RS} \sum_P f_{rsp} \log f_{rsp} + \sum_I \int_0^{x_i} c_i(u) du \quad (4.2a)$$

where f_{rsp} is the flow on path p , belonging to the set P of paths between the OD pair rs , θ is the logit dispersion parameter, x_i is the flow on link i belonging to the set I of links of the network and c_i is the separable cost on link i . This objective function is minimised at the logit SUE point for a network and is subject to the following constraints:

$$\sum_P f_{rsp} = q_{rs} \quad \forall rs \quad (4.2b)$$

$$x_i = \sum_{RS} \sum_P f_{rsp} \delta_{ip}^{rs} \quad \forall i \quad (4.2c)$$

$$f_{rsp} \geq 0 \quad \forall rsp \quad (4.2d)$$

Modifications to (4.2) have been proposed by Bekhor and Prashker (1999) to extend the same formulation of the SUE problem to choices modelled using the cross-nested and the paired combinatorial logit model.

Sheffi and Powell (1982) proposed to solve for SUE by minimising the following objective function:

$$z_{SUE}(\mathbf{x}) = -\sum_I \int_0^{x_i} c_i(u) du + \sum_I x_i c_i(x_i) - \sum_{RS} q_{rs} S_{rs}(\mathbf{x}) \quad (4.3)$$

where the symbols are as for (4.2) and q_{rs} and S_{rs} are respectively the total flow and the satisfaction between the OD pair rs . This programme, which assumes separable and strictly increasing link costs and additive path costs, can be used with any choice model and does not include constraints since the non-negativity of path and link flows and the consistency between flows on the set of paths between each OD pair and the total flow are directly satisfied at the solution.

Alternatively, the program (4.3) can be rewritten with a change of variable as a function of the link costs:

$$z_{SUE}(\mathbf{c}) = \sum_I \int_0^{c_i} c_i^{-1}(u) du - \sum_{RS} q_{rs} S_{rs}(\mathbf{c}) \quad (4.4)$$

This second version of the objective function has been used by Sheffi and Powell (1982) and Daganzo (1982) to show that, given separable and strictly increasing cost functions, at SUE link flows and costs are unique.

The discussion on the uniqueness of the SUE solution has been taken forward by Bell (Bell and Iida, 1997) showing that, at SUE, also path flows are unique.

Several link-based algorithms have been proposed for calculating the SUE point for a network. The most used is the Method of Successive Averages (MSA) introduced as an heuristic by Sheffi and Powell (1981) and later theoretically justified again by Sheffi and Powell (1982) proposing their objective function. They used the MSA both in the probit and in the logit case. Leurent (1997a,b) used the MSA along with a modification of the loading method of Dial (1971) devised to make such logit loading method stable in SUE calculations.

The MSA can be summarised in the following 5 steps:

1. Initialisation. Set the iteration number n to 0. Obtain an initial flow pattern $\mathbf{x}^{(0)}$ (usually by carrying out a stochastic loading on the network's free flow costs).
2. Calculation of the search direction. Increment the iteration number n by 1. Calculate the costs on the network due to the current flows and carry out a stochastic loading on such costs. The resulting flows $\mathbf{y}^{(n)}$ are known as "auxiliary" flows. The search direction is the vector $\mathbf{y}^{(n)} - \mathbf{x}^{(n)}$ linking the current and the auxiliary flows.
3. Calculation of the step length. Obtain the step length as $1/(1+n)$.
4. Calculation of the new current flow pattern. The new current flow pattern $\mathbf{x}^{(n)}$ is obtained by moving along the search direction by the above step length.
5. Convergence check. If the convergence check is satisfied terminate the calculation: the current flows solve the problem. Otherwise return to step 2.

The MSA is the only method suitable with simulation. In fact it converges also when the search direction is a descent direction only on average, as is the case with a simulation model (see Sheffi, 1985). This is relevant in particular when the "streamlined" version of the probit SNL proposed by Sheffi and Powell (1982) is employed. This modification of the algorithm assumes that in step 2 above a simple all or nothing loading on the sampled shortest path between each OD pair substitutes for the full stochastic loading of the network to generate the auxiliary solution.

In case of loadings calculated analytically, the MSA is a convenient method of performing the optimisation, as it is simple and does not require the direct use of the objective function. However, the MSA is not fast to converge because it reduces the step during the optimisation whilst in SUE there is no *a priori* reason for the step length to decrease since, as the algorithm gets close to the optimum, the current and the auxiliary solution get near to each other until they coincide. In fact, the distance between the current and the auxiliary solution can be used to perform the convergence check at step 5 above. This was remarked by Maher and Hughes (1997a).

A number of more efficient algorithms, alternative to the MSA, have been proposed in the literature. Chen and Alfa (1991b) retained the search direction used in the MSA algorithm but proposed to calculate the step length for the logit SUE problem by performing a line search using the deterministic part of the Fisk (1980) objective function. They also suggested a way to use the complete objective function but Bell *et al.* (1993) pointed out that such a method may result in path flows that are not consistent with the OD demand constraints. Akamatsu (1996) proposed a link-based method using the Fisk objective function that can be employed only with his logit loading method considering all the possible paths. Maher and Hughes (1996b, 1997a) used the search direction of the MSA but calculated the step lengths using the Sheffi and Powell (1982) objective function both in the probit and in the logit case, putting forward the most efficient methods in the literature. They proposed to carry out the line search by approximating the function along the search vector as a quadratic or as a cubic and take the optimal step as the point where the approximating function is minimised. They noticed that line searches carried out once only or until an improved objective function or gradient are found are more efficient for the overall convergence than line searches refined in a subinterval to a pre-set precision.

Maher (1998) investigated, in the logit case, the performance of algorithms using the same line search methods but along with the unconstrained and the constrained quasi-Newton DFP-BFGS search directions (see *e.g.* Scales, 1985). Although quasi-Newton methods are expected to be the most efficient optimisation techniques, in this case they resulted less efficient than those using the MSA search direction.

4.2.2.5 Path-Based Models and Algorithms for SNL and SUE

Several path-based techniques for stochastic network loading have been proposed in the literature. Ben-Akiva *et al.* (1984) used pre-assignment path-enumeration for a nested-logit model of uncongested interurban route choice. De la Barra *et al.* (1993) put forward a modified logit SNL model that tries to account for path overlapping. Yai *et al.* (1997) considered probit loading with the GHK simulator (Borsch-Supan and Hajivassiliou, 1993) on small sets of routes enumerated in advance.

A number of other path-based SNL methods have been proposed as part of SUE models.

All the path-based SUE models in the literature use the logit model or one of its extensions that account for path overlapping. Bell *et al.* (1993) proposed a variation of the Frank and Wolfe algorithm for UE to solve the Fisk logit SUE program enumerating the routes by means of column generation. Huang (1995) proposed a method to calculate logit SUE using paths enumerated in advance and an algorithm using the MSA search direction but performing line searches using the Fisk objective function. Leurent (1996a) calculated the logit SUE with two path-based formulations, one using the MSA technique and one using an extension to SUE of the UE “equilization” algorithm of Schittenhelm (1990) using paths fixed prior to the assignment. Damberg *et al.* (1996) considered column generation and extended the use of the disaggregate simplicial decomposition approach already employed by Larsson and Patriksson (1992) for UE, to solve for SUE using the Fisk objective function. Huang and Bell (1998) proposed a SUE logit model based on the Van Vliet (1981) formula to calculate link choice probabilities and employing the MSA to find the equilibrium solution.

Cascetta *et al.* (1996, 1997) proposed the C-logit model, a modified logit SNL model that accounts effectively for path overlapping with an added path cost that is a non linear function of the overlapping among the paths between the same OD. The C-logit model has been introduced in section 2.3.3.7. They also solved the C-logit SUE problem using MSA and paths enumerated in advance.

Bell *et al.* (1996, 1997) developed the PFE (Path Flow Estimator) a quasi-dynamic model for real-time use, based on the C-logit model. It divides the modelled time period into “slices” in which the traffic condition is approximated with SUE calculated with column generation and with the Fisk objective function with added constraints accounting for available data on the network situation (such as actual link flows). A second version of the PFE (Bell and Cassir, 1998), for the calculation of SUE without using real time information, uses MSA and column generation.

Vovsha and Bekhor (1998) and Prashker and Bekhor (1998) adapted for traffic assignment the cross-nested logit model of Vovsha (1997), introduced in section 2.3.3.4. In their application, the model is referred to as a link-nested logit model since the nests are the links and the elemental options are the paths that are enumerated in advance. They also obtained a link-based version of their model and used it with the MSA to solve the related SUE problem.

4.3 Algorithms for Multinomial Probit Path-Based Stochastic User Equilibrium

4.3.1 Introduction

The remaining part of this chapter is concerned with solving the multinomial probit path-based SUE problem (MNP PB SUE) using an analytical approximation method and considering the set of paths between each OD pair given and fixed at the beginning of the calculations. Several algorithms, some taken from the literature and some new in the traffic assignment context, are described and tested.

The algorithms studied have been obtained building on the work of Maher and Hughes (1997a) for solving for SUE using the analytical SAM model.

In particular, the MNP PB SUE flow and cost patterns are calculated minimising the objective function of Sheffi and Powell (1982):

$$z_{SUE}(\mathbf{x}) = -\sum_I \int_0^{x_i} c_i(u) du + \sum_I x_i c_i(x_i) - \sum_{RS} q_{rs} S_{rs}(\mathbf{x}) \quad (4.3)$$

where i is one of the I links of the network, x_i is the flow on the link and $c_i(\cdot)$ is the strictly increasing and separable cost function for link i returning the travel cost on it; q_{rs} is the flow between rs , one of the RS OD pairs and S_{rs} is the satisfaction, the expectation of the minimum cost of travel between the OD pair rs .

The path costs have been considered additive and no path-specific costs have been included. The assumption of separable and strictly increasing link costs allows us to show that the solution to (4.3) is equivalent to the SUE condition on a network. In fact, as shown by Sheffi and Powell (1982) a term of the gradient of (4.3) results:

$$\frac{\partial z_{SUE}}{\partial x_i} = (x_i - y_i) \frac{dc_i}{dx_i} \quad (4.5)$$

which is zero only when the current and the auxiliary solutions coincide, since the performance functions are strictly increasing, that is when the SUE conditions are verified. The objective function (4.3) is not generally convex, as discussed by Sheffi and Powell (1982), although it is so at the solution point. The uniqueness of its solution has been proved by rewriting it with a change of variable in terms of link costs (see (4.4)) and showing that the gradient of (4.4) and (4.3) vanish at the same point but the Hessian of (4.4) is positive definite everywhere. Because of the one to one correspondence between link flows and costs the minimum of (4.3) as well as that of (4.4) is a global one.

While Sheffi and Powell (1982) established the uniqueness of the SUE solution in terms of link flows and costs, Bell proved that at the SUE point also path flows are unique (Bell and Iida, 1997).

The uniqueness of the path flows and costs, assuming a given set of paths, can also be seen considering that, given the unique link flows and costs and a set of paths between each origin-destination pair, the path costs are also uniquely defined, as they are the sum of uniquely defined costs. For a given set of costs of the options, the stochastic choice model used will return a unique set of choice probabilities and, as the OD flows are fixed, they will result in a unique set of path flows.

Although (4.3) does not express the problem in terms of paths it has been considered to solve the problem with explicit consideration of the paths because it is convenient to use and has a proven unique solution and because of the correspondence of the solution in path and link terms.

The rest of this section is organised as follows. After discussing the methodology and the test cases used, the algorithms already proposed in the literature and on which this work builds are described and tested. Then the results from the two strands of work that have been followed to try to obtain more efficient algorithms are reported. First algorithms using alternative step calculation methods are described and their efficiency is assessed. Then the rationale for a family of possible alternative search directions is given. Finally, algorithms employing alternative search directions based on the rationale illustrated are described and tested.

4.3.2 Test Methodology and Test Bed

4.3.2.1 Algorithm Test Methodology

The efficiency of the algorithms illustrated in the following sections has been tested by applying them to calculate the SUE traffic pattern on some test networks. During the calculations the progress of a convergence statistic has been recorded as well as the number of iterations, the number of loadings and the time required to obtain such progress. The algorithms have been compared on the computational cost required to reach a pre-set level of convergence.

The convergence statistic employed has been adapted from Maher and Hughes (1997a) to consider path flows, and is given by:

$$\ln RMSnd = \ln \sqrt{\frac{1}{P} \sum_P \frac{(x_p - y_p)^2}{(0.5(x_p + y_p))^2}} \quad (4.6)$$

where x_p and y_p are respectively the current and auxiliary path flow for path p and P is the number of paths included in the statistic which are those for which either the current or the auxiliary solution is at least 0.1% of the relevant OD flow. Such limitation has been introduced to avoid conditioning the convergence on paths carrying very small amounts of the total OD flow.

The $\ln\text{RMSnd}$ is the natural logarithm of a non-dimensional measure of distance between the current and the auxiliary solution, the latter resulting from a stochastic network loading on the current network costs. As the equilibrium is approached the two flow patterns tend to get near to each other until they coincide when the equilibrium is reached. Thus the measure of their distance will tend to zero and its logarithm to $-\infty$. The logarithmic transformation is used to make the convergence trend more evident, especially as the algorithms tend to the solution, as discussed in Hughes (1998) and as verified for this work.

The convergence performance of an algorithm is assessed by evaluating the number of stochastic loadings it requires to reach the solution defined by a target level of the RMSnd statistic that here is set at 10^{-4} , corresponding to a $\ln\text{RMSnd}$ value of -9.21 .

Iterations cannot be used as a reference for evaluating the computational effort since with different algorithms they require different number of stochastic network loadings and time, depending on the algorithms used.

The number of stochastic loadings has been chosen as the reference elementary operation to avoid referring directly to the computational time that is computer-related and as they seems representative of the computational effort required by the algorithms. This has been verified by comparing the time taken by different algorithms when the time used for obtaining and writing ancillary data is eliminated. Also when comparing algorithms with different search directions it has been verified that the additional computational time required to obtain search directions different from that referred to in the following as the traditional search direction (the MSA one) is of very limited importance compared to the time to carry out a loading.

Therefore the smaller the number of loadings required by an algorithm to reach a pre-set threshold of the statistics $\ln\text{RMSnd}$, the more efficient the algorithm.

The tests have been carried out using two MNP approximation methods: the improved Clark (1961) approximation and the Mendell-Elston (1974) approximation with the optimised calculation order of Kamakura (1989).

4.3.2.2 Algorithm Test Bed

The algorithms investigated in this study have been tested on three networks: the network used by Chen and Alpha (1991a), the Sioux Falls network (used *e.g.* in LeBlanc, 1975; Vythoukcas, 1990) and the central Headingley network (used *e.g.* in Maher *et al.*, 1999). Their main characteristics are summarised in table 4.1.

	Links	Nodes	Centroids	Active ODs
Chen and Alpha	34	12	4	4
Sioux Falls	76	24	24	528
Headingley	188	73	29	240

Table 4.1 - Characteristics of the test networks. The active ODs column indicates the number of OD pairs between which the flow is positive.

The links on the networks have BPR performance functions (US BPR, 1964) that can be written as:

$$c_i(x_i) = c_{0i} + \beta_i \left(\frac{x_i}{cap_i} \right)^{\alpha_i} \quad (4.7)$$

where c_{0i} is the link free flow cost, cap_i is referred to as the capacity of the link, β_i and α_i are further coefficients of the function. However, the Headingley network includes some connectors (links with fixed costs) therefore, on such network, the application of the algorithms investigated here should be considered heuristic.

On the network of Chen and Alpha, the set of routes fixed at the outset of the assignment is given by the ten simple paths between each OD. The original OD matrix has been multiplied by 0.5.

For the Sioux Falls and Headingley networks, two sets of paths have been enumerated on each network using a simple heuristic enumeration method applied on the free flow network costs (the costs of the links when there is no traffic on them) and on the costs resulting from a user equilibrium assignment.

The k -shortest path heuristic employed consists of repeating a number of times (300 in the cases considered) a procedure similar to the operations required to carry out stochastic network loading by simulation (see *e.g.* Sheffi, 1985): the link costs are sampled from their distributions and the shortest path between each OD pair, according to the sampled costs, is calculated with the algorithm of Dijkstra (1959) and recorded. At the end of the calculations, a list of shortest paths and the number of times they have been recorded is obtained. The n paths between each OD required at the outset of the enumeration (here $n=10$) are obtained as the n most sampled paths. Of course, if less than the required paths have been enumerated between an OD pair, those obtained are considered. This path enumeration technique is not intended to replicate exactly the methods used in the practical applications reviewed above but rather to obtain in a simple way sets of paths to test the algorithms.

Table 4.2 summarises the data for the sets of routes considered for the Sioux Falls network whilst those for Headingley are reported in table 4.3. It can be readily noted that the use of the UE costs reduced the size of the path sets.

Name	Link costs	Max no. of paths/OD	Aver. No. of paths/OD
S1	free flow	10	2627
S2	UE	10	2009

Table 4.2 – Names and characteristics of the path sets on the Sioux Falls network used to test the assignment algorithms.

Name	Link costs	max no. of paths/OD	Aver. No. of paths/OD
H1	free flow	10	830
H2	UE	10	545

Table 4.3 – Names and characteristics of the path sets on the Headingley network used to test the assignment algorithms.

4.3.3 Application of Algorithms in the Literature

4.3.3.1 The Method of Successive Averages (MSA)

The simplest and most common technique to minimise program (4.3) is the Method of Successive Averages (MSA) put forward by Sheffi and Powell (1981, 1982). It moves by a predetermined step along the search direction determined at each iteration as it has been detailed in section 4.2.2.4 and thus requires carrying out one stochastic loading only at each iteration.

The search vector is limited by two feasible flow patterns, the current and the auxiliary flows and, because of the convexity of the flow set, is always a feasible search direction.

Sheffi and Powell (1982) noted that the MSA is the only method that can be used to solve the SUE problem when the flows are obtained by simulation and therefore the search direction is a descent direction only on average.

Even not using a simulation-based loading method, the MSA is a convenient method of performing the optimisation, as it does not require the direct use and calculation of the objective function: the SUE point can be obtained using only network loadings.

However, when the choice model is solved analytically, alternative methods not using a fixed step generally converge faster. This has been shown in the link-based logit and probit case *e.g.* by Maher and Hughes (1997a). They remarked that this is

also because, in the MSA, the step is reduced during the optimisation whilst in SUE there is no reason for the step length to decrease since, as the algorithm gets close to the optimum, the current and the auxiliary solution get near to each other until they coincide.

Typical convergence behaviours of the MSA algorithm are shown in the figures included in the next and the following sections: e.g. in fig. 4.3 for the Chen and Alpha network, in fig. 4.6 for a case on the Sioux Falls network and 4.8 for a set of paths on the Headingley network. Those figures compare the MSA with some of the algorithms described in the next section, showing that, as found *e.g.* in Maher and Hughes (1997a), the MSA is outperformed (except at the beginning of the calculations) by other algorithms in the literature.

In fact, here the MSA is considered as a base algorithm against which to compare alternative ones and also as it can offer advantages during the first iterations of algorithms employing other step calculation methods, as suggested again by Maher and Hughes (1997a), and as confirmed by the examples proposed in the figures.

4.3.3.2 Quadratic and Cubic Interpolation Methods

Maher and Hughes (1997a), in their work on the solution of SUE with the SAM network loading model, devised a number of algorithms for optimising (4.3) using the same search direction as the MSA (referred to in the following as the traditional search direction) but finding the steps by minimising the objective function (4.3) along it, rather than using a fixed step length as in the MSA. Such a minimisation of the function along the search direction is carried out by fitting a quadratic or a cubic function to it and taking as the estimate of the optimal step the point where the fitted function is minimised.

A quadratic can be fitted to the objective function along the search direction using the values of its gradient at the two extremes of the search direction (these two points directly bracket the minimum).

Considering the search direction at iteration n , $(\mathbf{y}^{(n)} - \mathbf{x}^{(n)})$, a point $\mathbf{x}^{(\lambda)}$ along it can be written as:

$$\mathbf{x}^{(\lambda)} = \mathbf{x}^{(n)} + \lambda(\mathbf{y}^{(n)} - \mathbf{x}^{(n)}) \quad (4.8)$$

(this expression is valid referred to both link and path flows). The gradient of the objective function (4.3) at $\mathbf{x}^{(\lambda)}$ along the search direction, for separable link costs functions, can be obtained as:

$$\frac{dz_{SUE}}{d\lambda} = \sum_I \frac{\partial z_{SUE}}{\partial x_i^{(\lambda)}} \frac{dx_i^{(\lambda)}}{d\lambda} \quad (4.9)$$

where x_i is the flow on link i . As shown by Sheffi and Powell (1982) and reported in (4.5):

$$\frac{\partial z_{SUE}}{\partial x_i^{(\lambda)}} = (x_i^{(\lambda)} - y_i^{(\lambda)}) \left(\frac{dc_i(x)}{dx} \right)_{x=x_i^{(\lambda)}} \quad (4.10)$$

where $y_i^{(\lambda)}$ is the auxiliary solution obtained considering the costs consistent with the flows $x_i^{(\lambda)}$. A component of the derivative of (4.8) w.r.t. λ is:

$$\frac{dx_i^{(\lambda)}}{d\lambda} = (y_i^{(n)} - x_i^{(n)}) \quad (4.11)$$

Substituting (4.10) and (4.11) in (4.9) gives:

$$\frac{dz_{SUE}}{d\lambda} = g_\lambda = \sum_I (x_i^{(\lambda)} - y_i^{(\lambda)}) \left(\frac{dc_i(x)}{dx} \right)_{x=x_i^{(\lambda)}} (y_i^{(n)} - x_i^{(n)}) \quad (4.12)$$

The value of the gradient g_0 at the current solution $\mathbf{x}^{(n)}$, and therefore at $\lambda=0$, can be obtained by substituting the current and the auxiliary solution in (4.12) respectively for $x_i^{(\lambda)}$ and $y_i^{(\lambda)}$ thus using the same data employed to determine the search direction. At the auxiliary solution $\mathbf{y}^{(n)}$, for $\lambda=1$, the gradient g_1 , can be obtained by considering in (4.12) the auxiliary solution as $x_i^{(\lambda)}$ and the flow resulting from a further loading on the costs consistent with the auxiliary solution as $y_i^{(\lambda)}$.

Fitting a quadratic to the objective function corresponds to fitting a linear function to the gradient between the two extremes of the search direction. The point λ^* at which

the function fitted to the gradient is zero is then taken as the estimate of the point where the objective function is minimised, and is given by:

$$\lambda^* = \frac{-g_0}{g_1 - g_0} \quad (4.13)$$

Maher and Hughes (1997a) proposed to fit a cubic to the objective function along the search direction by using the values of the objective function z_0 and z_1 at the extremes of the search vector along with the values of the gradient. This corresponds to fitting a quadratic to the gradient. The estimate λ^* of the point where the objective function is minimised is again as the point where the gradient of the fitted function is zero, which is given by:

$$\lambda^* = \frac{3(z_0 - z_1) + 2g_0 + g_1 + \sqrt{(3(z_0 - z_1) + g_0 + g_1)^2 - g_0g_1}}{3(2(z_0 - z_1) + g_0 + g_1)} \quad (4.14)$$

These interpolation methods can be simply applied once along the search vector, in which case the information needed to use them requires carrying out two stochastic loadings, one at the current solution (which is also the loading used to determine the search direction) and one at the auxiliary solution, as also discussed above when illustrating how to determine the gradient at the two extremes of the search direction. Maher and Hughes (1997a) and Hughes (1998) tested also the effect of refining the line search in subintervals chosen so that they contain the minimum. The refinement of the line search in subintervals is carried out with the same method used in the original interval until the newly found point corresponds to an improved gradient (when compared to the one at the current solution) in the quadratic interpolation case or to a smaller value of the objective function in the cubic interpolation case. Alternatively the line search can be refined to a given precision, that is until the improvement in the refinement of the step is under a certain threshold. Refining the step requires carrying out a further stochastic loading each time a new subinterval is considered.

Considering the possibility of refining the step in subintervals, the line search methods proposed by Maher and Hughes in their work on SAM for SUE can be summarised in the following six:

- quadratic interpolation, simple;
- quadratic interpolation, refined in subintervals until an improved gradient is found;
- quadratic interpolation, refined in subintervals to a given precision;
- cubic interpolation, simple;
- cubic interpolation, refined in subintervals until an improved objective function is found;
- cubic interpolation, refined in subintervals to a given precision.

The quadratic interpolation method requires only flow data to calculate the gradients, therefore when used with the probit model it can be used directly with any approximation method, including those not providing the value of the satisfaction function.

With the cubic interpolation, it is necessary to know the value of the satisfaction function, along with the link flow data, to calculate the value of the objective function (4.3). As seen in chapter 3, only with the method of Clark, in either of its implementations, or with the method of Langdon can the satisfaction be calculated. Therefore the cubic interpolation can be used directly with these approximations. Moreover, the method used in the Langdon approximation to obtain the satisfaction can be adapted for use with the Mendell-Elston approximation. To use the cubic interpolation method with other approximations, one of the methods to approximate the satisfaction suggested in chapter 3 should be used.

In the remainder of the present chapter the hybrid method suggested in chapter 3 is used with the Mendell-Elston approximation: the path choice probabilities are calculated with the Mendell-Elston approximation but the expected value of the minimum of the costs is obtained with the approximation of Clark. This device has no effect on the final solution since the algorithm switches from the cubic interpolation method to the quadratic one as soon as the solution is in the vicinity of the SUE solution and the result is fully consistent with the Mendell-Elston approximation.

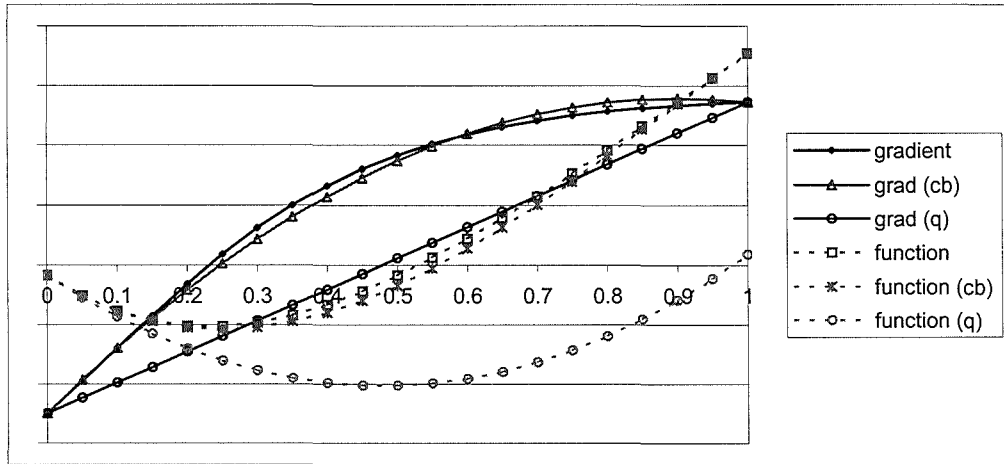


Fig. 4.1 – SUE objective function and gradient (on two different scales) with the quadratic (q) and cubic (cb) approximations to the SUE objective function and the resulting approximated gradients along a search direction close to the start of the optimisation (the function is approximated evaluating the satisfaction with the method of Clark). Sioux Falls network, path set S1, Mendell-Elston approximation.

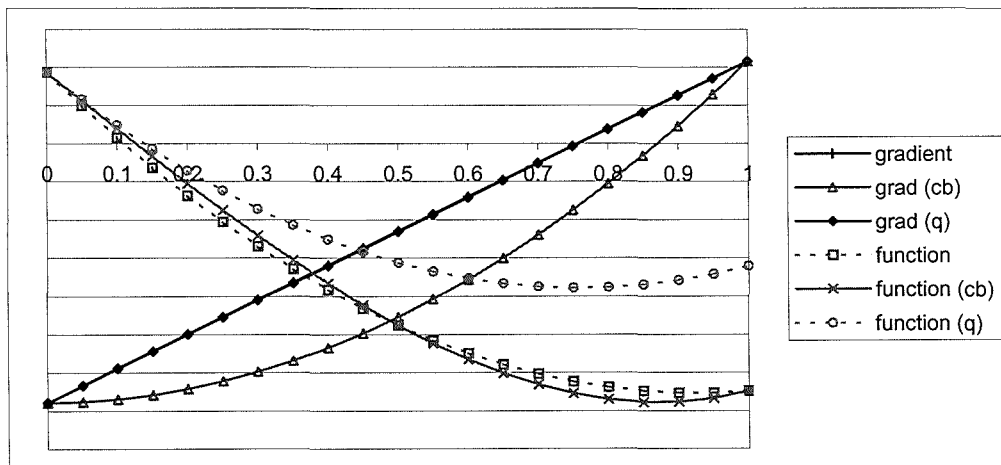


Fig. 4.2 – SUE objective function and gradient (on two different scales) with the quadratic (q) and cubic (cb) approximations to the SUE objective function and the resulting approximated gradients away from the start of the optimisation (the function is approximated evaluating the satisfaction with the method of Clark). Notice the inconsistency between the SUE objective function and its gradient. The actual gradient and the gradient obtained by quadratic interpolation coincide. Sioux Falls network, path set S1, Mendell-Elston approximation.

In fact, at the beginning of the optimisation, the SUE objective function is well approximated by a cubic function (see the example in fig. 4.1). However, as the

solution is approached, the objective function and its gradient become inconsistent, due to the approximate evaluation of the satisfaction, as shown in fig. 4.2. Maher and Hughes (1997a), noting this inconsistency, suggested to complete the optimisation using gradient information only, and thus the quadratic interpolation, to ensure that the point at which current and auxiliary solutions are identical is found (see also Hughes, 1998).

The results of Maher and Hughes (1997a) and Hughes (1998) showed that the rate of convergence improves dramatically when an algorithm with an optimised line search is used instead of the MSA. Their experiments showed clearly that algorithms with quadratic and cubic line searches carried out once or respectively until an improved gradient or objective function is found improve on the convergence performance of the MSA. With the networks they used, the cubic search gave the better overall performance and they also found that refining the line searches to a pre-set level of precision is less efficient than moving to the next search direction after obtaining a step giving an improved objective function or gradient, although it is generally faster than the MSA.

The application of the algorithms of Maher and Hughes to the cases considered here gave results similar to those in the literature.

An inspection of the convergence trends of the algorithms with one interpolation only and refined until a better gradient or objective function is found, showed that they coincide, both with the Clark and with the Mendell-Elston approximation, except in some cases on the Headingley network.

In fact, the trends given by the two approximation methods on the same set of paths are rather similar and the hybrid device for the cubic interpolation with the Mendell-Elston approximation seems to work well.

In accordance with the results of Maher and Hughes, refining the line searches to a pre-set level of precision gives convergence trends more efficient than the MSA but not as efficient as with the corresponding interpolation when the step is calculated with one interpolation only or checking for improvement. Examples of this behaviour are provided in figure 4.3 for the Chen and Alfa network and in figure 4.4 for a set of paths enumerated on the Sioux Falls network. The behaviour on the Headingley network is less clear: in some cases algorithms with refined step do as well as the algorithms checking for gradient improvement but not as well as those performing one line search only. In other cases the algorithms with refined line search do less well than the others. An example of the results on the Headingley network is given in fig. 4.5.

Considering all the results it seems therefore worth focussing on the algorithms not refining the step search, consistent with what was suggested by Maher and Hughes.

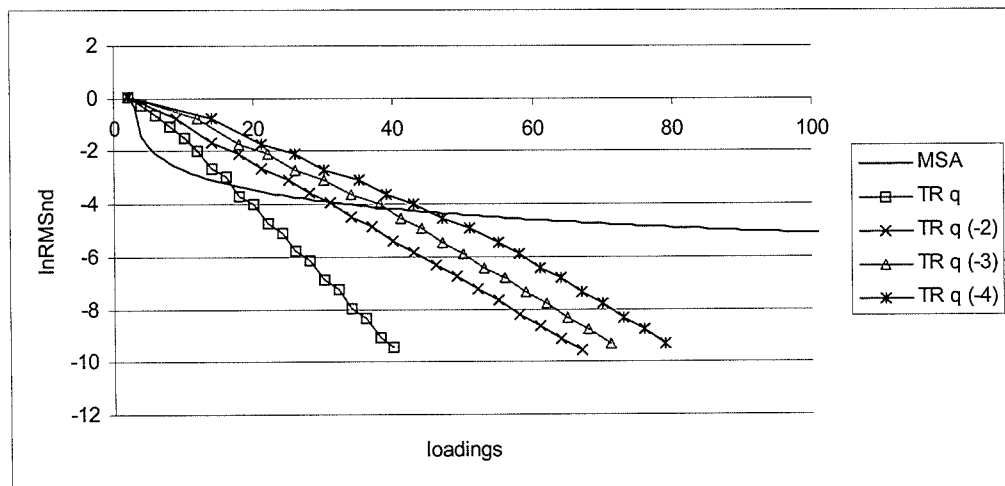


Fig. 4.3 – Comparison of the MSA, the quadratic (TRq) and the quadratic with refined step (TRq(power of ten of the precision required)) interpolation methods. Chen and Alfa network, Mendell-Elston approximation.

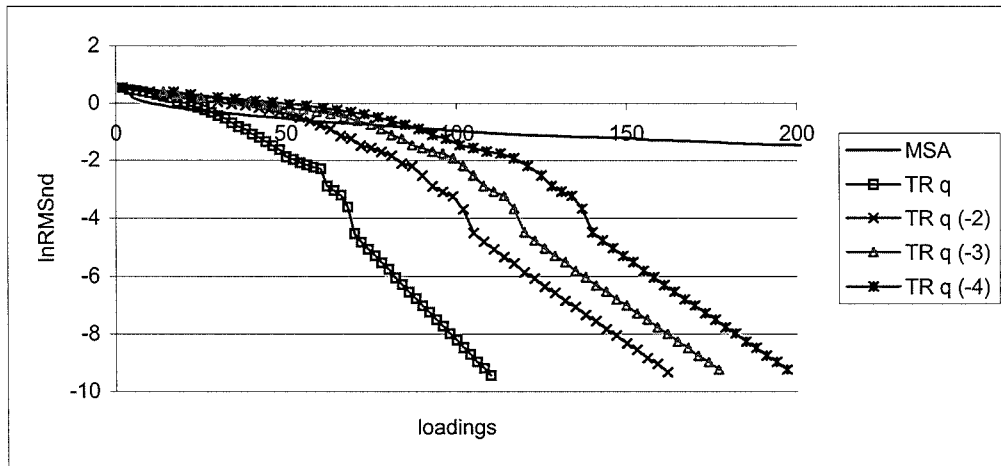


Fig. 4.4 - Comparison of the MSA, the quadratic (TRq) and the quadratic with refined step (TRq(power of ten of the precision required)) interpolation methods. Sioux Falls network, path set S1, Clark approximation.

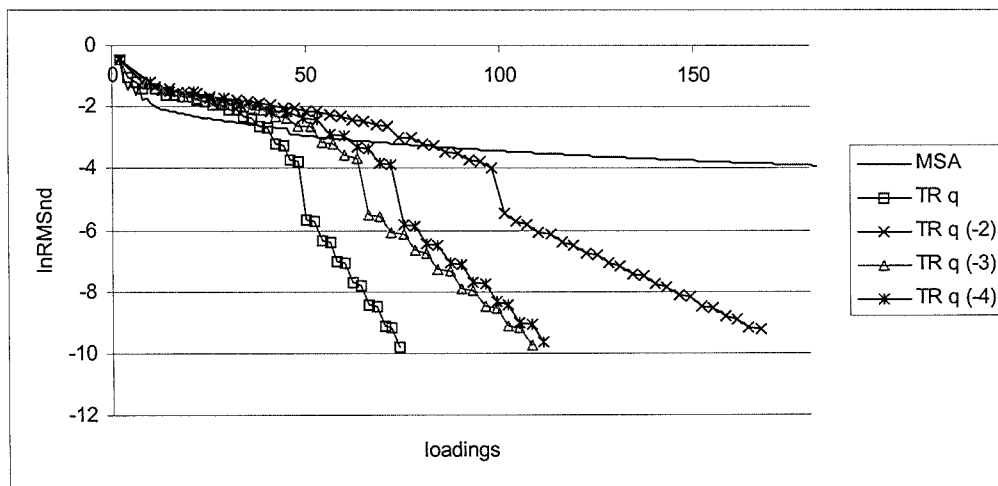


Fig. 4.5 – Comparison of the MSA, the quadratic (TRq) and the quadratic with refined step (TRq(power of ten of the precision required)) interpolation methods. Headingley network, path set H2, Clark approximation.

Algorithms carrying out the quadratic line search once or until an improved objective function is found are more efficient than the MSA. Examples of the results obtained are reported again in figs. 4.3, 4.4 and 4.5 and also in figs. 4.6 and 4.7.

The cubic search algorithms show a similar behaviour: examples of their convergence trends are provided in figure 4.6 for the Sioux Falls network and 4.7 and 4.8 for the Headingley network. Those figures also compare the convergence trends of the corresponding cubic and quadratic line search methods and show that

which is the best performing is less clear cut than in the cases examined by Maher and Hughes. According to the results obtained here, on the Chen and Alfa and the Sioux Falls network (see fig. 4.6) the cubic does better, whilst on the Headingley network the quadratic method provides the most efficient convergence trend on one set of paths (compare figs. 4.7 and 4.8).

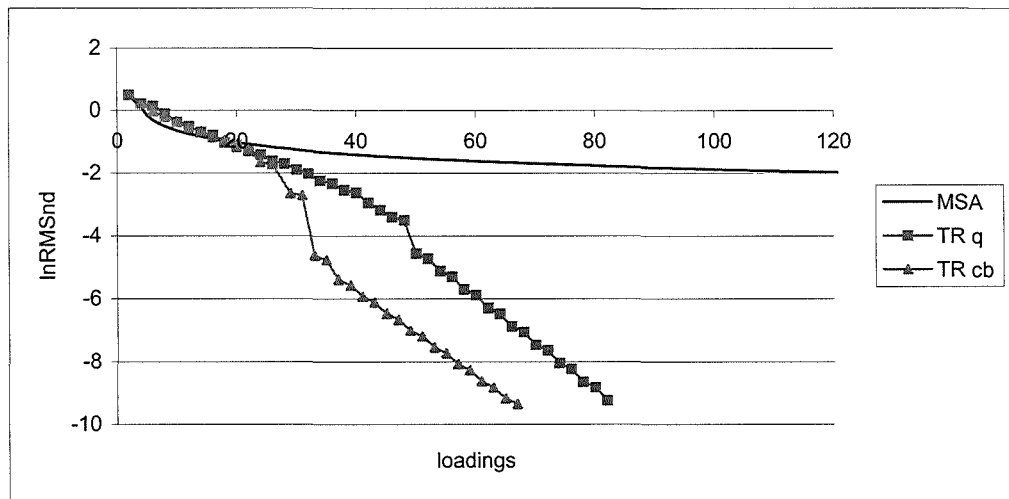


Fig. 4.6 – Comparison of the MSA and the quadratic (TRq) and cubic (TRcb) interpolation methods. Sioux Falls network, path set S2, Clark approximation.

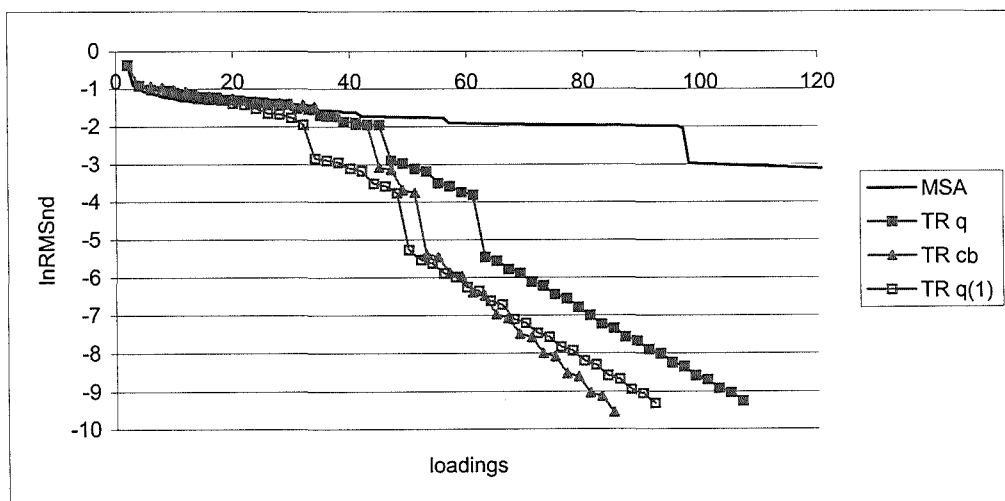


Fig. 4.7 – Comparison of the MSA and the quadratic (TRq(1) one interpolation, TRq with gradient improvement) and cubic (TRcb) interpolation methods. Headingley network, path set H1, Mendell-Elston approximation.

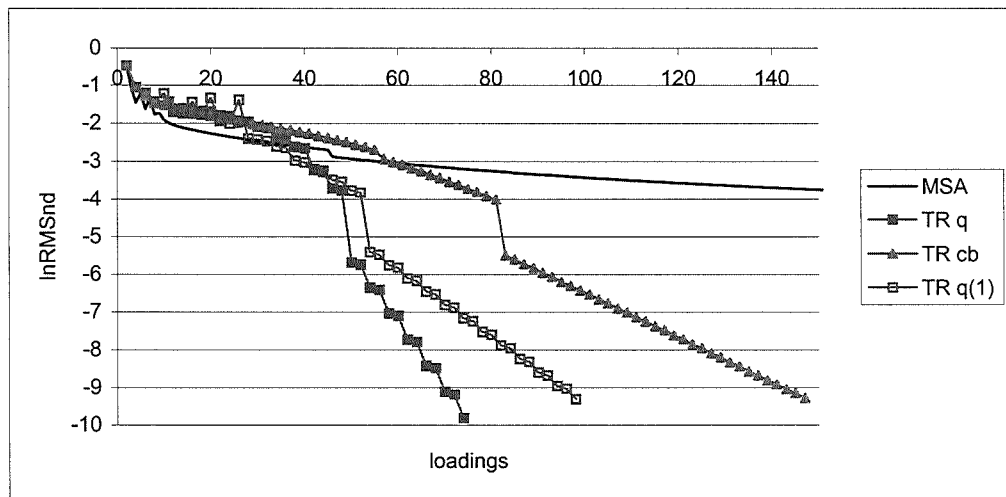


Fig. 4.8 – Comparison of the MSA and the quadratic (TRq(1) one interpolation, TRq with gradient improvement) and cubic (TRcb) interpolation methods. Headingley network, path set H2, Clark approximation.

The data for all networks examined show that the MSA tends to be more efficient than the optimised step methods at the start of the calculations, although after a few iterations its convergence trend, as shown in figures 4.3 to 4.8, tends to flatten out or, anyway, becomes less efficient than the optimised line search methods.

The initial efficiency of the MSA is due to the fact that it requires a single loading per iteration whilst the other methods require at least two. After a number of iterations, spending a higher computational effort in determining the step becomes more effective whilst the MSA runs out of steam as it gives too small a step.

This behaviour was characterised also by Maher and Hughes who suggested to obtain more efficient algorithms by using hybrid algorithms, starting with some MSA iterations before employing optimised line search. Such hybrid algorithms have also been tested here (a selection of results is reported in figs. 4.9, 4.10, 4.11) and, consistently with the results of Maher and Hughes, some of them are more efficient than those using optimised search directions from the start. Again the trends for hybrid algorithms with step refined until an improved gradient (for the quadratic case) or objective function is found (for the cubic case) coincide and do so also for the Headingley network suggesting that the differences between the methods are concentrated in the behaviour during the first iterations.

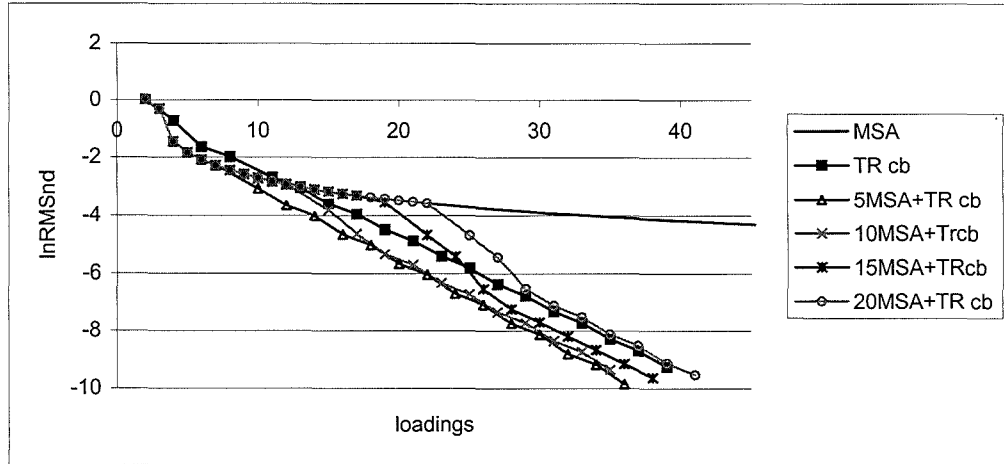


Fig. 4.9 – Comparison of the MSA, the cubic interpolation method (TRcb), and hybrid algorithms based on TR cb started with the number of MSA indicated. Chen and Alpha network, Mendell-Elston approximation.

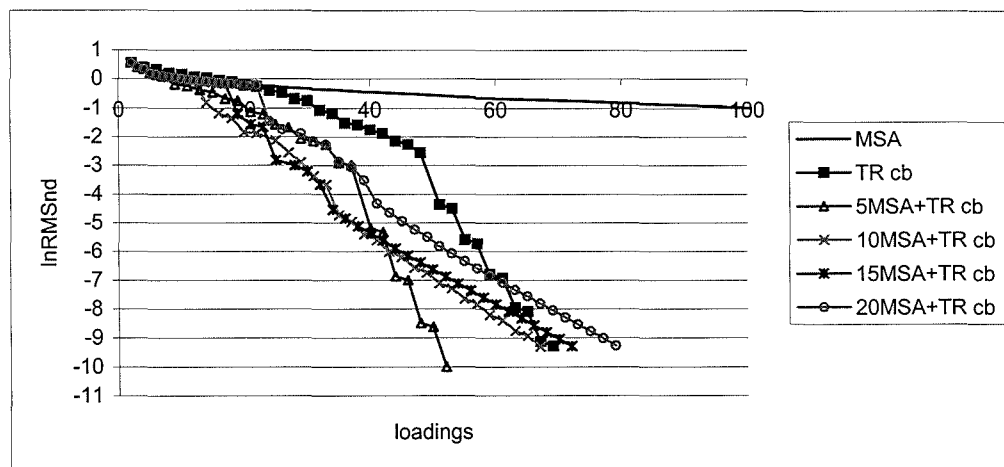


Fig. 4.10 – Comparison of the MSA, the cubic interpolation method (TRcb), and hybrid algorithms based on TR cb started with the number of MSA indicated. Sioux Falls network, path set S1, Clark approximation.

The tests on hybrid algorithms have been carried out starting with 5, 10, 15 and 20 MSA iterations. It is difficult to suggest a generally valid best number of initial MSA iterations, since the same number can have different effects on different networks, as can be seen from the figures. It seems, however that in many cases few MSA iteration (5-10) are enough to give a good improvement over the algorithms using lines search from the start.

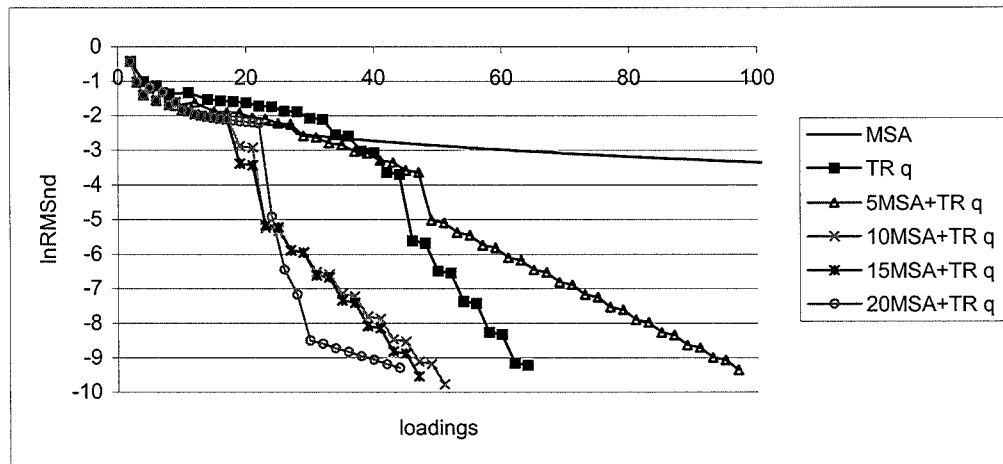


Fig. 4.11 – Comparison of the MSA, the quadratic interpolation method (TRq), and hybrid algorithms based on TR q started with the number of MSA indicated. Headingley network, path set H2, Mendell-Elston approximation.

4.3.3.3 Conclusions

The results obtained here with the MSA and with the algorithms of Maher and Hughes confirm, in general, those of Maher and Hughes (1997a). The MSA, widely used to solve the SUE problem, is outperformed by algorithms using the same search direction as the MSA (the traditional search direction) but with an optimised step. Refining the step determination to a pre-set precision is not effective in most cases examined and it is better to use a less expensive estimate of the step length, interpolating only once or until a better gradient or objective function is found, and moving to the next search direction, as in the investigations reported in Hughes (1998).

Interpolating once or seeking a gradient or objective function improvement gave the same results, except on the Headingley network, where results are less clear cut, and generally not refining the step gives a better convergence behaviour.

Although generally less efficient than other methods, the MSA typically outperforms them at the beginning of the calculations. The hybrid algorithms suggested by this behaviour, started with some MSA iterations before turning to seek optimal steps along the search direction, often give more efficient convergence trends, although different number of initial MSA iterations may have a different effect on the overall efficiency of the algorithms on different networks.

4.3.4 Algorithms with Alternative Step Calculation Methods

4.3.4.1 Introduction

The investigation about possible alternative step calculation methods to be used with the traditional search direction concentrated on pre-determined step methods and alternative gradient interpolation methods since, as in the SUE case gradient information is readily available, interval reduction methods such as the bisection algorithm should not be of advantage.

One pre-determined step calculation method has been considered, the Method of Weighted Averages (MWA), whose performance is evaluated in the next section. The following sections discuss three alternative optimised step methods: the *regula falsi*, the rational interpolation and the partial step quadratic methods.

4.3.4.2 The Method of Weighed Averages (MWA)

The MSA can be modified considering a different but as simple and inexpensive mechanism to obtain the step length. In fact, as remarked by Sheffi (1985, p.328), the MSA is a particular case of a general algorithm whose step, at each iteration, is:

$$\lambda^* = \frac{a}{b+n} \quad (4.15)$$

where a is a positive number, b a non-negative number and n is the number of the current iteration. The coefficients a and b should be such that the sequence of steps obtained complies with the following requirements:

$$\sum_{n=1}^{\infty} \lambda_n = \infty \quad (4.16)$$

$$\sum_{n=1}^{\infty} \lambda_n^2 < \infty \quad (4.17)$$

where (4.16) ensures that the algorithm does not stop because of the step length and (4.17) is a condition required in a simulation based algorithm (where the search

direction is a random variable) to minimise the variance of the results (see Sheffi, 1985).

The MSA is obtained when a and b are set to 1, whilst if a and b take different values the step determination rule referred to here as the Method of Weighed Averages is obtained. The MWA name is suggested from the fact that the different auxiliary solutions will have different weights in the final solution whilst in the MSA case they all have the same importance.

A step calculation method of the MWA sort has also been used for the MNP calibration problem by Sheffi *et al.* (1982) with $a=10$ and $b=4$.

The interest of the MWA lies in that it retains the inexpensive determination of the step typical of the MSA (it requires carrying out one stochastic loading only at each iteration) but could obviate its slow convergence speed.

Using a larger than b is not considered here as the resulting steps, during the first iterations would be larger than 1, whilst it is known that the current and auxiliary solutions bracket the minimum. Having a smaller than b may result, in early iterations, in steps smaller than those returned by the MSA. A number of cases with $a = b$ are investigated. Nine different steps, and therefore nine different algorithms, are tested by considering a and b to take each integer value between 2 and 10 included. The higher is a , the smaller is the effect of n on the step and the more the algorithm will take new solutions close to the auxiliary solution, especially at the beginning of the calculations. As the algorithm progresses and the step becomes smaller, the effect on the step length of having chosen a different $a=b$ becomes less important, but the step remains longer than the one given by the MSA would be for the same iteration.

The tests on the networks gave mixed results: in general the MWA converges more efficiently than the MSA but which particular MWA performs better seems to depend on the characteristics of the problem considered. For instance, observing figure 4.12 it is clear that in that case (and, indeed in general on the Sioux Falls network) the MWAs with large values of $a=b$ like 8-10 give very good convergence

trends, that are practically as efficient as those given by optimised line search algorithms. MWAs with similar coefficients are also the most efficient ones on the small Chen and Alfa network. It is also interesting to note that the convergence trends for different MWA with large values of $a=b$ are very close to each other. However, as shown in fig. 4.13, on the Headingley network the best convergence trends are obtained with smaller $a=b$ like 2-4 and the convergence trend oscillates noticeably. Because of this and because in the Sioux Falls case the MSA does better at the beginning of the calculations or there is little to choose between MWA and MSA it seems that the MWAs examined do not provide a better starting method than the MSA for employment in hybrid algorithms of the sort suggested by Maher and Hughes.

Thus MWAs, in general, are more efficient than the MSA and in some cases also of optimised step algorithms but which particular MWA performs best seems to depend on the network on which it is applied.

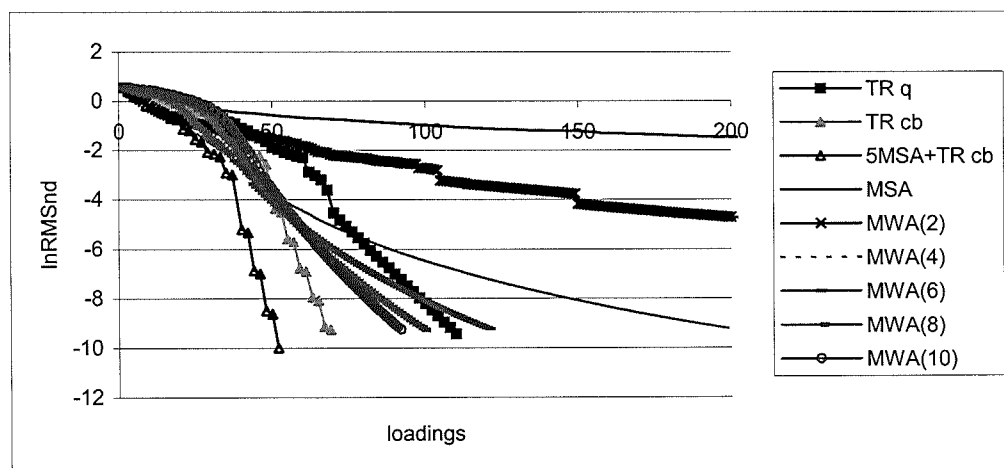


Fig. 4.12 – Comparison of the MSA, the quadratic interpolation method (TRq), the cubic interpolation methods (TRcb), a hybrid algorithms based on TR cb and MWA algorithms with the coefficients indicated in brackets. Sioux Falls network, path set S1, Clark approximation.

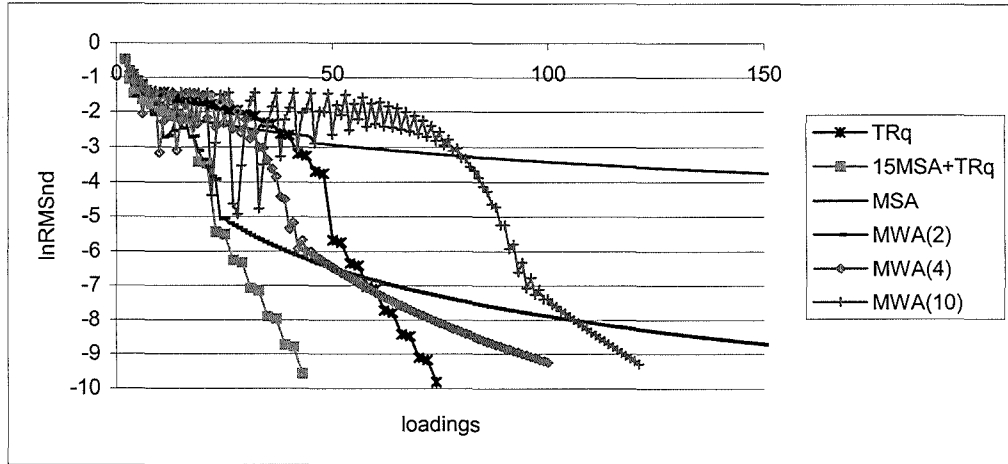


Fig. 4.13 – Comparison of the MSA, the quadratic interpolation method (TRq), a hybrid algorithm based on TR q and MWA algorithms with the coefficients indicated in brackets. Headingley network, path set H2, Clark approximation.

4.3.4.3 The Regula Falsi Method

The *regula falsi* method is an alternative to using the quadratic interpolation method of Maher and Hughes for approximating the objective function as a quadratic in the vicinity of the current point and determining an estimate of the step length.

This method is an approximate version of the Newton line search: it uses g_0 , the gradient of the objective function along the search direction at the current point, and an estimate of the second derivative h_0 to approximate the point λ^* where the objective function is at a minimum (and its derivative is 0) as:

$$\lambda^* = -\frac{g_0}{h_0} \quad (4.18)$$

The reason for using an estimate of h_0 rather than its exact value is that calculating the latter involves obtaining the Hessian of the objective function, which in turn requires the calculation of the Jacobian of the path-choice probabilities (as detailed in Sheffi, 1985, p.319 and reported in 4.3.5.2). This can be carried out in the path-based case considered here but it is rather computationally demanding, especially if the calculations are carried out numerically. The analytical derivation procedure suggested in chapter 6 could be applied to this case, although this is not done here.

As with the quadratic interpolation method the estimate of the step can be refined to make sure that an improved gradient is found at the new solution point or until the step is determined to a given precision. The latter method is not considered as it has already been checked that it results in algorithms that are typically less efficient than those spending less computational effort in the step determination (this was also confirmed for the *regula falsi* interpolation case by investigations not reported here). Therefore this algorithm can be implemented as:

- *regula falsi*, simple;
- *regula falsi*, refined in subintervals until an improved gradient is found.

The estimate of h_0 , the second derivative of the objective function, is obtained by calculating g_2 , the gradient of the objective function along the search direction at a point close to the current solution by using a second loading at that point (in the algorithm considered here at a distance δ of 0.001 the length of the search direction), and considering it along with the gradient at the current point g_0 to obtain:

$$h_0 = \frac{g_2 - g_0}{\delta} \quad (4.19)$$

As the step to obtain the second derivative is fixed, this algorithm is partially a heuristic: a precise method should determine the optimal distance for the second derivative calculation at each iteration on the basis of the relevant error. But this has not been considered here, as it would involve possible additional loadings, thus undermining the efficiency of the algorithm.

Thus, if the interpolation is applied once at each iteration, it requires two loadings. Two more loadings are required each time it is refined in a subinterval.

The trend for the algorithms performing only one iteration and those checking that an improved gradient has been found before moving to the next search direction coincide in both the Clark and the Mendell-Elston cases, except in some cases on the Headingley network.

On the Chen and Alpha network, the *regula falsi* is seen to be slightly more efficient than the quadratic interpolation method in the literature at the beginning of the assignment, as can be seen from fig. 4.14, suggesting that using information taken at the present solution captures better the non linear trend of the objective function gradient at the beginning of the assignment. However, there is little to choose between the optimised step algorithms in the literature and the *regula falsi* on the overall convergence trend.

Compared to the quadratic method in the literature, the *regula falsi* does slightly better with one set of paths on the Sioux Falls network and slightly worse with the other, the data for the latter being reported in figure 4.15. However, the differences are limited, and there seems to be little to choose between the original quadratic method and the *regula falsi*. The same figure shows that, on the Sioux Falls network, the *regula falsi* does not reach the efficiency of the cubic interpolation method.

On the Headingley network the algorithms with step calculated once and those obtaining it after checking that it gives an improved gradient do not coincide and the algorithm returning the step without refining it gives a faster convergence. Especially the latter, used from the start performs better than or as well as the algorithms in the literature, as shown in fig. 4.17, except on one set of paths.

The use of a hybrid algorithm starting with a number of MSA iterations before using the *regula falsi* method improves on the original algorithm, as it happens with other interpolation methods. However, although some MSA iterations to start are beneficial, the number to obtain the best overall efficiency seems to vary from case to case.

This can be seen for instance from the results on the Chen and Alpha network: five and ten MSA iterations improve the overall efficiency of the algorithm while more make it less efficient. Also with some MSA to start there is little to choose between the optimised step algorithms in the literature and the *regula falsi*, as shown again in fig. 4.14.

The convergence trends for the Sioux Falls network, as the one reported in fig. 4.16, show that the *regula falsi* does as well as the quadratic interpolation method. With a

large number of MSA iterations to start it has a convergence trend that cannot be distinguished from that of the cubic or of the other linear interpolation methods, but in some cases the cubic prevails.

On the Headingley network, some MSA iterations to start make the algorithm with one interpolation only and those checking on the gradient improvement coincide. In hybrid algorithms started with more than 10 MSA iterations the *regula falsi* step gives a convergence trend that is as efficient as the other methods based on the quadratic interpolation. With less MSA iterations it gives variable results, being sometimes less sometimes more efficient than the quadratic and cubic method in the literature (see fig. 4.17). The use of the MSA to start improves on the efficiency of the algorithm except in one case where the use of the algorithm without improvement check from the start gives a very fast convergence trend (this is again the case depicted in fig. 4.17).

Thus the *regula falsi* method, although it gives interesting convergence trends on the Headingley network, seems to add very little to the quadratic method already in the literature and can be proposed as a possible alternative interpolation method, but does not deliver an improved performance.

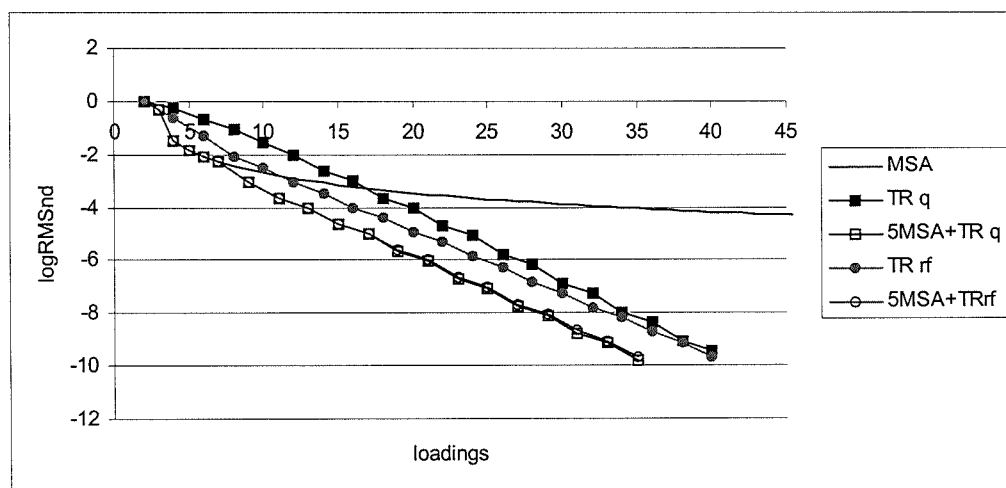


Fig. 4.14 - Comparison of the MSA, the quadratic (TRq), the regula falsi (TR rf) and hybrid algorithms based on TR q and TR rf started with the number of MSA indicated. Chen and Alpha network, Mendell-Elston approximation.

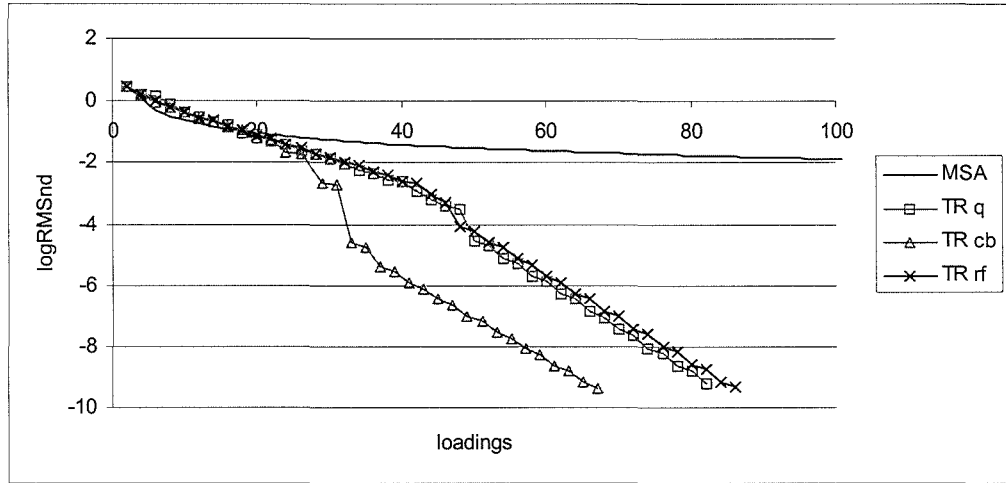


Fig. 4.15 – Comparison of the MSA, the quadratic (TRq), the cubic (TR cb) and the regula falsi (TR rf). Sioux Falls network, path set S2, Clark approximation.

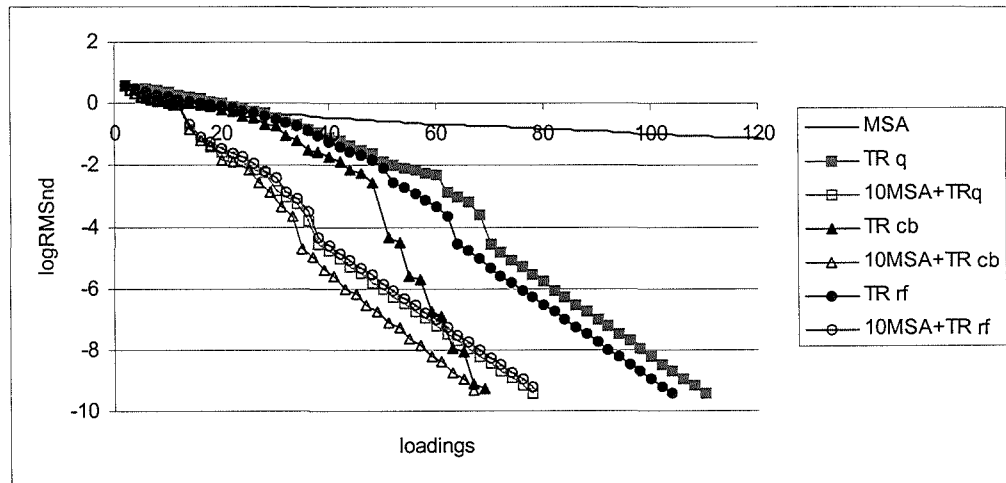


Fig. 4.16 – Comparison of the MSA, the quadratic (TRq), the cubic (TR cb), the regula falsi (TR rf) and hybrid algorithms based on TR q, TR cb and TR rf started with the number of MSA indicated. Sioux Falls network, path set S1, Clark approximation.

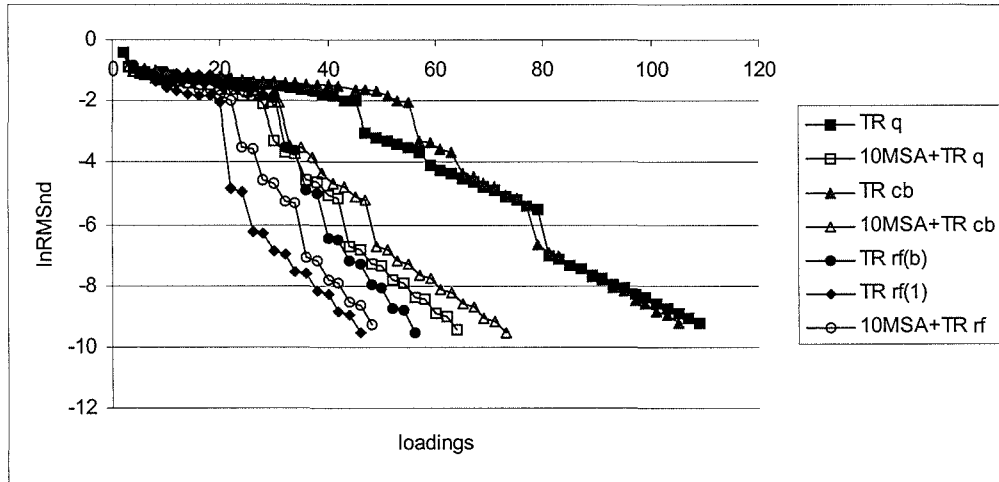


Fig. 4.17 – Comparison of the MSA, the quadratic (TRq), the cubic (TR cb), the regula falsi (TR rf) and hybrid algorithms based on TR q, TR cb and TR rf started with the number of MSA indicated. Headingley network, path set H1, Clark approximation.

4.3.4.4 The Rational Interpolation Method

A further way to try to fit closely the actual trend of the objective function's gradient during the first iterations of the calculations consists of fitting a rational function to the gradient of the objective function along the search direction. Visual inspection of the gradient and the rational function trends confirm their good agreement (see the example in fig. 4.18). Moreover, as the actual gradient trend becomes linear approaching the solution the rational function follows it, as shown in fig. 4.19.

Calculating the gradient at three points in the search interval (here at the two extremes and at the middle point) by performing a stochastic loading at each of those points the coefficients of a rational function such as:

$$r(\lambda) = \frac{a + \lambda}{b + c\lambda} \quad (4.20)$$

can be retrieved and the estimate of the step length is obtained as the zero of such function, namely as:

$$\lambda^* = -a = -g_0 \frac{g_1 - g_{0.5}}{g_1 g_{0.5} - 2g_0 g_1 + g_0 g_{0.5}} \quad (4.21)$$

where g_0 , $g_{0.5}$ and g_1 are respectively the values of the gradient of the objective function (4.3) at the beginning, at half length and at the end of the search interval.

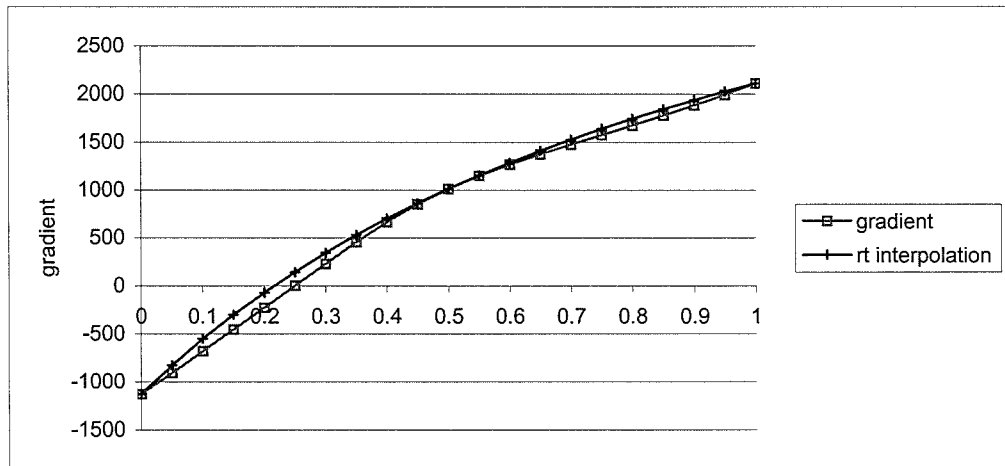


Fig. 4.18 – Comparison of trend of the SUE objective function gradient along the search direction and of its rational function approximation when the SUE algorithm is away from the solution. Sioux Falls network, path set S2, Mendell-Elston approximation.

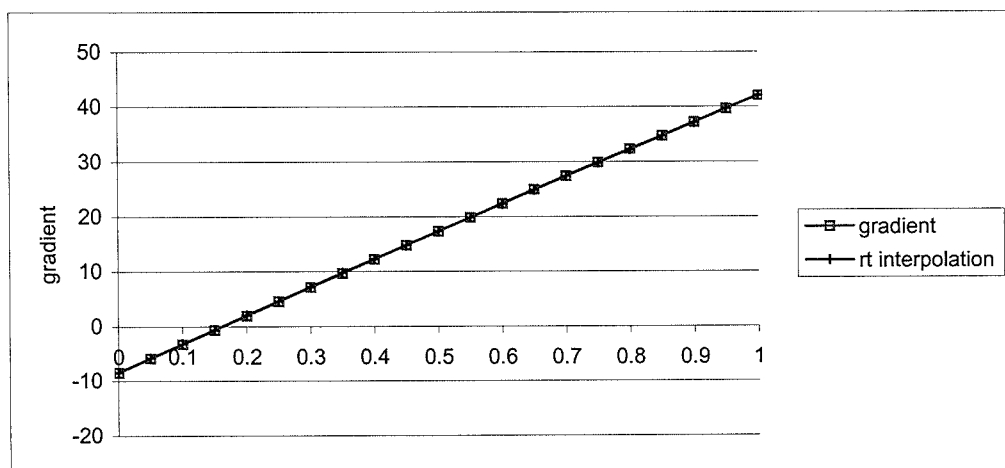


Fig. 4.19 – Comparison of trend of the SUE objective function gradient along the search direction and of its rational function approximation when the SUE algorithm is close to the solution. Sioux Falls network, path set S2, Mendell-Elston approximation.

As with other optimised step algorithms, the calculations can be refined in subintervals. Refining the step to a given precision gives convergence trends similar to those for the line search methods suggested by Maher and Hughes (1997a) and is not considered further. Therefore there are 2 different possible implementations of this algorithm:

- Rational, simple;
- Rational, refined in subintervals until an improved gradient is found.

In the present implementation, the refinement in subintervals is carried out using a quadratic interpolation to limit the number of loadings at each iteration, as it requires only one additional loading per interpolation rather than the two needed for the rational function. Moreover, although the rational function can adapt to a linear trend, to save on computational effort, the search in the main interval is switched to the quadratic one described in section 4.4.3 as soon as the gradient is detected to be almost linear. The gradient is considered almost linear when the value of $g_{0.5}$ is within 5% of the linear interpolation of the gradients at the extremes of the search interval.

The tests on the networks suggested that the rational interpolation method can be almost seen as a variation of the quadratic interpolation method as, except in the Chen and Alpha network, after a few iterations with the rational interpolation the gradient satisfies the condition above and the algorithm switches to the quadratic method. This near linearity is consistent with the trends observed when plotting the gradient of the objective function when an algorithm is away from the start.

The trend for the algorithms performing only one iteration and those checking that an improved gradient has been found before moving to the next search direction coincide in both the Clark and the Mendell-Elston cases on all networks employed.

On the Chen and Alpha network there seems to be little to choose between the quadratic interpolation and the rational interpolation method except at the beginning,

when it does marginally better, as it was intended. An example of this behaviour can be appreciated in fig. 4.20. On the Sioux Falls network the rational interpolation gives mixed results: its performance is approximately similar to that of the quadratic but in some cases better, in others slightly worse. Figure 4.21 is relative to a case when they are equivalent whilst fig. 4.23 depicts a case where it does better. However convergence trends such as the ones reported in figs. 4.22 and 4.24, show that this is not the case on the Headingley network.

Investigating hybrid algorithms on the Chen and Alpha network shows that starting the rational interpolation method with 5 or 10 MSA iterations gives a more efficient convergence trend than when it is used from the beginning. However, as can be seen in fig. 4.20, in those cases the rational interpolation method does marginally worse than the quadratic method of Maher and Hughes (due to the extra loading needed to obtain the step).

Hybrid algorithms typically improve the efficiency of the rational interpolation algorithm applied to the Sioux Falls and the Headingley network as well. Figs. 4.22 and 4.23 show such an improvement but also that, in most cases, there is little to choose between the quadratic interpolation and the rational one when they are used within hybrid algorithms. Altogether it seems that the intended better gradient interpolation to be obtained with the rational interpolation is not effective in improving on the quadratic convergence trend.

Thus, although the rational interpolation method can be proposed as a method alternative to those in the literature, it does not deliver any general efficiency improvement on those and in some cases it does worse than them.

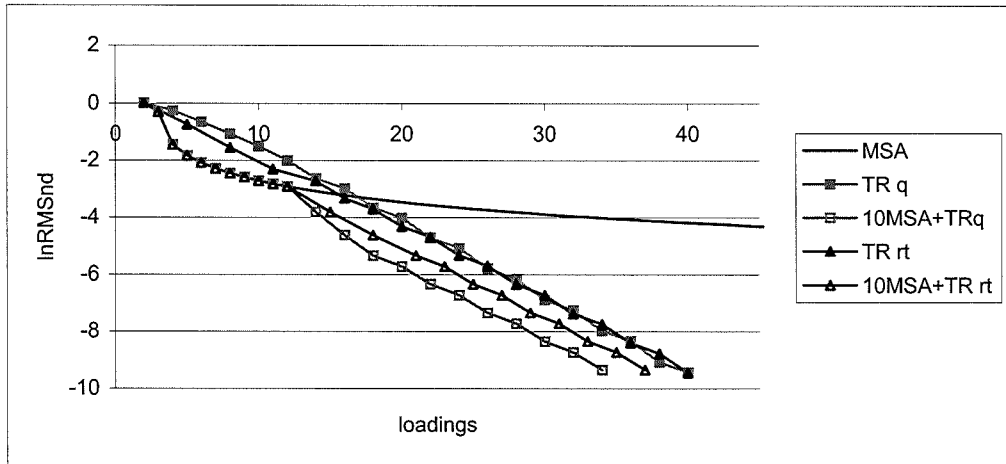


Fig. 4.20 – Comparison of the MSA, the quadratic (TRq), the rational interpolation (TR rt) and hybrid algorithms based on TR q, TR rt started with the number of MSA indicated. Chen and Alpha network, Mendell-Elston approximation.

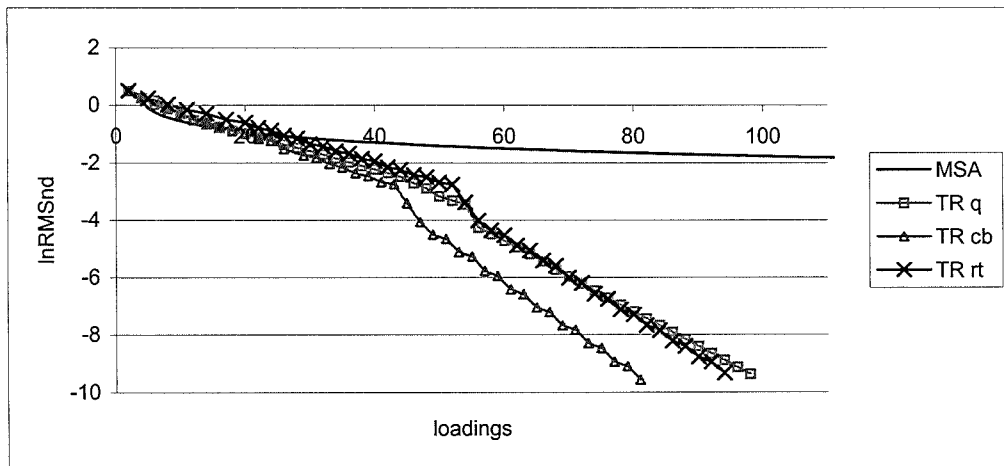


Fig. 4.21 – Comparison of the MSA, the quadratic (TRq), the cubic (TR cb), the rational (TR rt) interpolation algorithms. Sioux Falls network, path set S2, Mendell-Elston approximation.

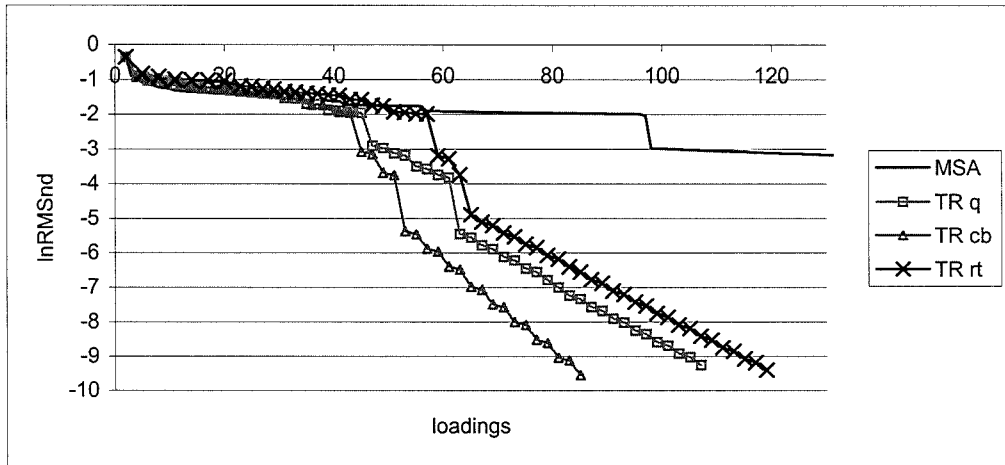


Fig. 4.22 – Comparison of the MSA, the quadratic (TRq), the cubic (TR cb), the rational (TR rt) interpolation algorithms. Headingley network, path set H1, Mendell-Elston approximation.

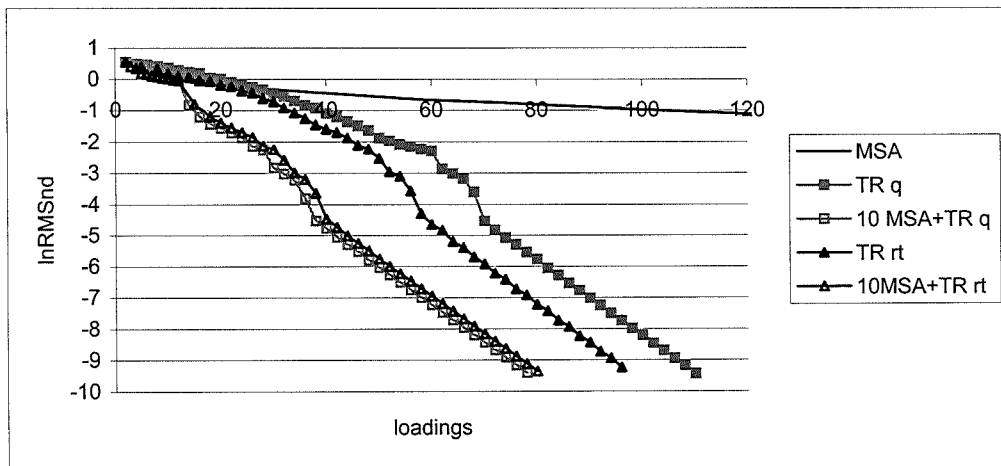


Fig. 4.23 – Comparison of the MSA, the quadratic (TRq), the rational (TR rt) interpolation algorithms and hybrid algorithms based on TRq and TR rt started with 10 MSA iterations. Sioux Falls network, path set S1, Clark approximation.

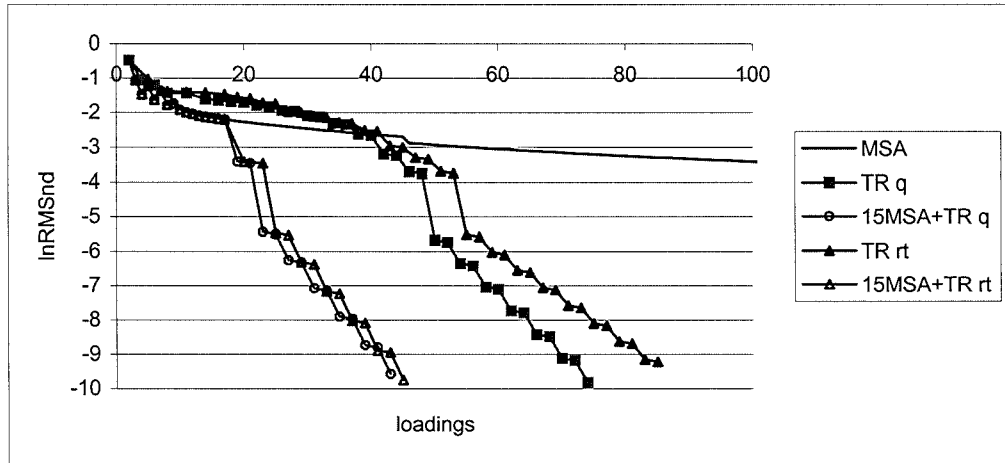


Fig. 4.24 – Comparison of the MSA, the quadratic (TRq), the rational (TRrt) interpolation algorithms and hybrid algorithms based on TRq and TR rt started with 15 MSA iterations. Headingley network, path set H2, Clark approximation.

4.3.4.5 Partial Step Quadratic Interpolation Methods

The last type of line search method tested in this series of experiments is a modification of the quadratic line search of Maher and Hughes (1997a) consisting of fitting a linear function to the gradient of the objective function (4.3) using the gradient at the current solution g_0 and the gradient g_β at an intermediate point β between the current and the auxiliary solution, rather than at the auxiliary solution.

Testing such an algorithm is suggested again by the observations on the trend of the gradient along the search direction when the algorithm is away from the solution. This motivated also the idea of trying fit to the gradient with a rational function.

When the gradient is not linear, using information taken at an intermediate point along the search direction could be of help in obtaining a better first estimate of the step length. Closer to the solution, the gradient tends to become linear, thus using the information at the two extremes to fit a linear function or at an extreme and at an intermediate point is irrelevant.

The estimated step length using the partial step quadratic interpolation method results:

$$\lambda^* = \frac{-\beta g_0}{g_\beta - g_0} \quad (4.22)$$

which for $\beta=1$ reduces to (4.14). Values of β of 0.33 and 0.50 have been used in the tests performed.

In the algorithm actually implemented, if the line search returns a point out of the present search vector such a solution is discarded and a quadratic search using the gradient at the present and at the auxiliary solution is carried out instead.

As with the other optimised line search algorithms, the step can be refined in subintervals chosen to contain the minimum, by using the gradients at the extremes of such intervals.

Algorithms refining the search until a given precision is met are not considered as it has already been ascertained that limiting the effort to characterise the step sizes gives more efficient trends. Thus, two types of line search methods result, depending on whether the gradient improvement is checked:

- Partial step quadratic simple;
- Partial step quadratic refined in subintervals until an improved gradient is found.

The first type of algorithm requires only two loadings per iteration. The second type requires one additional loading each time the step is refined in a subinterval, except if the subinterval is bracketed for the first time also by the auxiliary solution, when two additional loadings are needed (one at the point resulting from the previous interpolation and one at the auxiliary solution).

In general the numerical tests showed that the partial step quadratic methods perform better than the MSA and in some cases significantly better than the cubic and quadratic interpolation algorithms in the literature, but this improvement is not consistent across the tests investigated.

The trend for the algorithms performing only one iteration and those checking that an improved gradient has been found before moving to the next search direction coincide in both the Clark and the Mendell-Elston cases, except in a few cases on the Headingley network. The latter network, once more, gives different and less clear cut results than those obtained on the Chen and Alpha and the Sioux Falls network.

The tests on the Chen and Alpha network suggest that there is little to choose between different β s. In particular, observing data such as those depicted in figure 4.25, it seems that the algorithms using either intermediate point do as well as the cubic interpolation method at the beginning of the assignment but there is little to choose between the cubic interpolation and partial step quadratic interpolation as the solution gets close. It is however interesting to note that they perform better than the original quadratic interpolation method.

On the Sioux Falls network the partial step quadratic methods do as well or better than the original quadratic method at the beginning of the assignment. Altogether, on one set of paths both partial steps improve on the original quadratic method (though not outperforming the cubic interpolation method) but on the other set of paths only the partial step 0.5 improves or does as well as the original quadratic method (see fig. 4.26) whilst the other does as well or worse than the original quadratic step method (depending on the approximation used).

On the Headingley network the results given by the two different partial step lengths are mixed. Fig. 4.29 reports a case in which the partial step 0.33 does better than the other and of the quadratic. But in other cases they are less efficient than the quadratic methods.

From these results it is not possible to characterise the best intermediate point at which it is better taking the gradient information, although considering only the Chen and Alpha and Sioux Falls cases using 0.5 should be better.

Employing the partial step quadratic methods in hybrid algorithms (started with a number of MSA iterations before determining the step by interpolation) on the Chen and Alpha network, improves the overall efficiency of the algorithms when 5 or 10

MSA iterations are used but it worsens it if more MSA iterations are used (a set of results showing this is reported in figure 4.28). Moreover, when hybrid algorithms are considered there is little to choose between the quadratic interpolation method of Maher and Hughes and those using a partial step started with the same number of MSA iterations.

On the Sioux Falls network, when the algorithm is started with some MSA iterations there is, as well, little to choose between the two partial step quadratic methods and the original quadratic method and with few MSA iterations they are all outperformed by the cubic method, as can be seen in the example reported in fig. 4.29.

On the Headingley network, the device of starting the algorithms with some MSA steps improves on the efficiency of the original algorithms, and in some cases started with 5 and 10 MSA iterations the partial step quadratic methods do better than both the corresponding cubic and quadratic method in the literature (one of these cases is reported in fig. 4.30). When more than 10 MSA iterations are used to start the algorithms the convergence trends due to different optimised steps practically coincide.

Although designed to capture the non-linear trend of the gradient at the beginning of the optimisation, the partial step method does not outperform consistently the methods in the literature when applied directly, and there is often little to choose between them and the original quadratic method when the calculations are started with a number of MSA iterations, which is not surprising since, close to the solution, the gradient tends to have a linear trend. Therefore, with the results obtained here, the partial step quadratic methods do not seem to add much to the efficiency of the quadratic method in the literature.

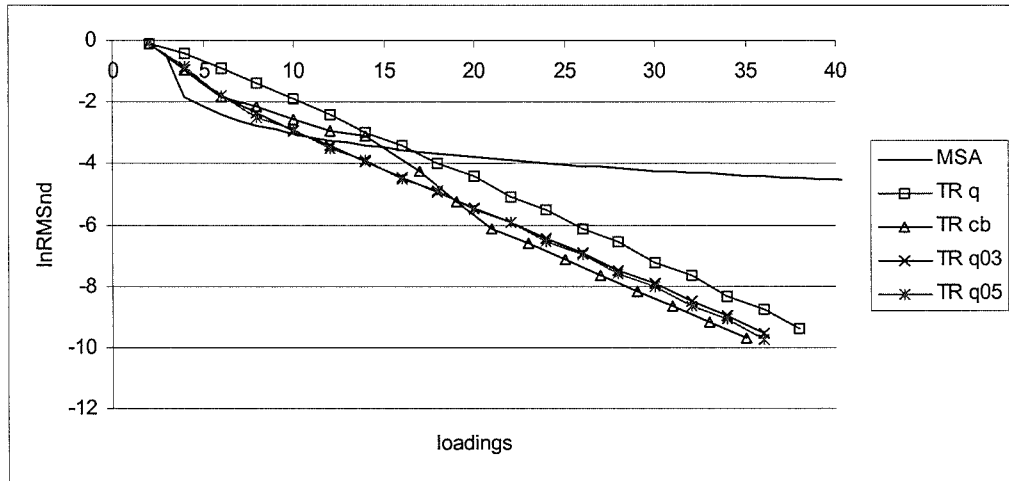


Fig. 4.25 – Comparison of the MSA and algorithms using the quadratic (TRq), the cubic (TRcb), the quadratic partial step $\beta=0.33$ (TRq03) and the quadratic partial step $\beta=0.5$ (TRq05) interpolations. Chen and Alpha network, Clark approximation.

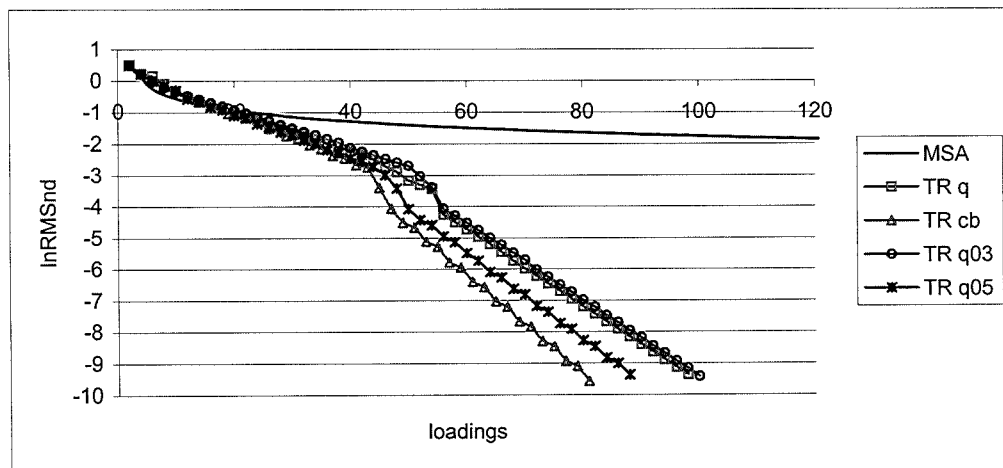


Fig. 4.26 – Comparison of the MSA and algorithms using the quadratic (TRq), the cubic (TRcb), the quadratic partial step $\beta=0.33$ (TRq03) and the quadratic partial step $\beta=0.5$ (TRq05) interpolations. Sioux Falls network, paths set S2. Mendell-Elston approximation.

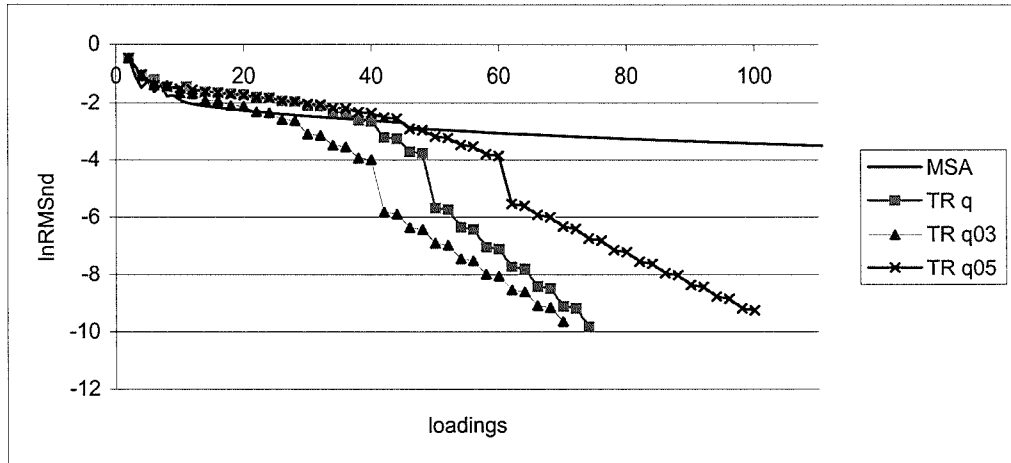


Fig. 4.27 – Comparison of the MSA and algorithms using the quadratic (TRq), the quadratic partial step $\beta=0.33$ (TRq03) and the quadratic partial step $\beta=0.5$ (TRq05) interpolations. Headingley network, path set H2, Clark approximation.

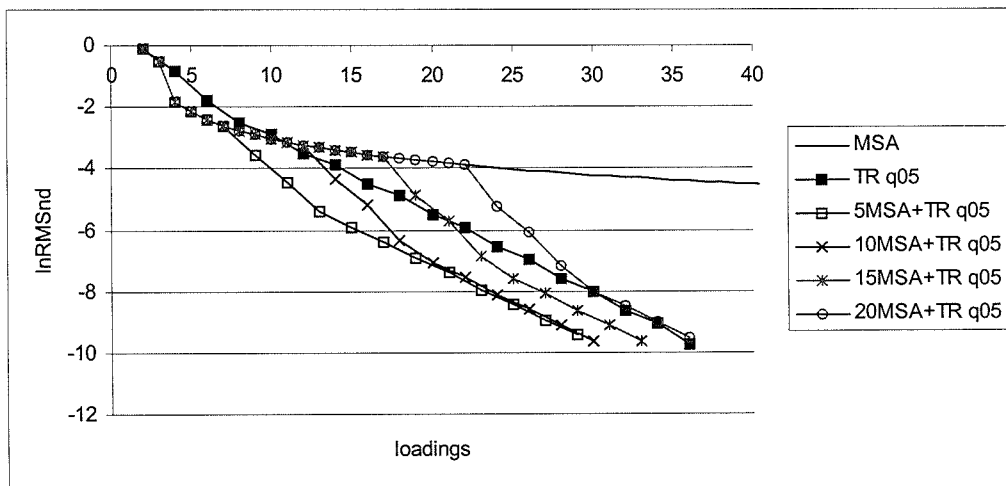


Fig. 4.28 – Comparison of the MSA an algorithm using the quadratic partial step $\beta=0.5$ (TRq05) interpolation and hybrid algorithms based on TRq05 started with the number of MSA iterations indicated. Chen and Alpha network, Clark approximation.

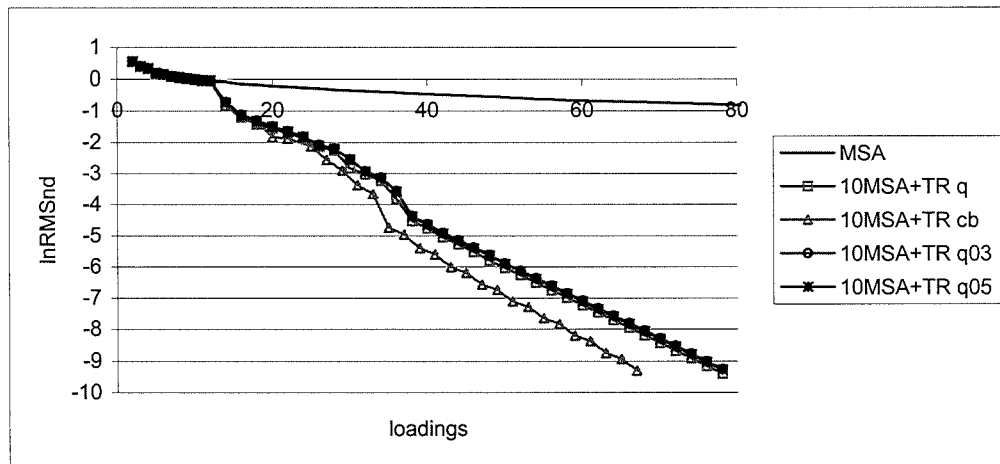


Fig. 4.29 – Comparison of the MSA and hybrid algorithms started with 10 MSA iterations using the quadratic (TRq), the cubic (TRcb), the quadratic partial step $\beta=0.33$ (TRq03) and the quadratic partial step $\beta=0.5$ (TRq05) interpolations. Sioux Falls network, path set S1, Clark approximation.

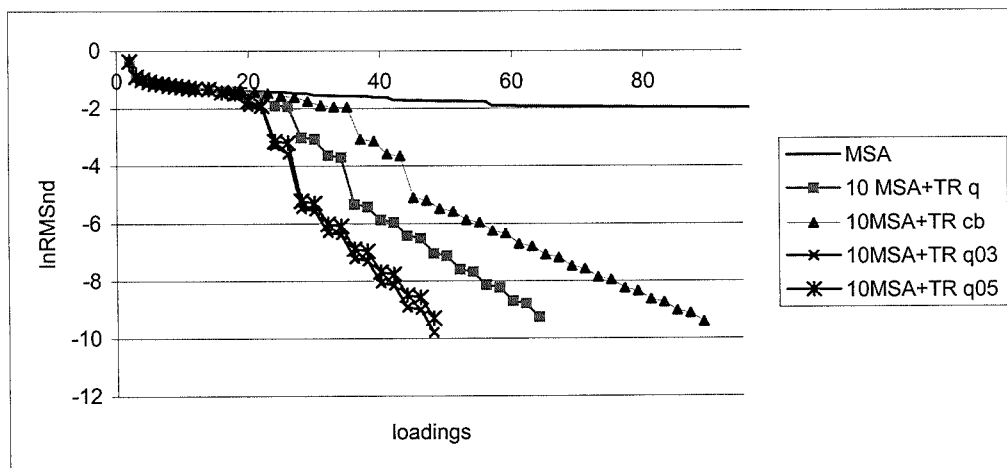


Fig. 4.30 – Comparison of the MSA and hybrid algorithms starting with 10 MSA iterations before using the quadratic (TRq), the cubic (TRcb), the quadratic partial step $\beta=0.33$ (TRq03) and the quadratic partial step $\beta=0.5$ (TRq05) interpolations. Headingley network, paths set H1, Mendell-Elston approximation.

4.3.4.6 Conclusions

The Method of Weighted Averages gives convergence trends that are in all cases more efficient than those of the MSA and, in some cases, outperform the algorithms optimising the step along the search direction although not their hybrid versions. The tests on the Headingley network gave, as with other step calculation methods, results that are not in accordance with those on the other networks. In particular they suggest the possibility of important oscillations of the convergence trend and characterise different MWA as the best performing ones. The MWA with a low value for the coefficient $a=b$ seems, however, to present small oscillations and improve on the MSA in all the cases examined although not necessarily on the algorithms optimising the step, and can be suggested as a valid alternative to the MSA.

The tests on the three interpolation methods considered allow us to discard the rational interpolation method as it does only as well or worse than the methods in the literature.

The *regula falsi* and the partial step methods can be used as alternatives to the interpolation methods in the literature but do not bring substantial and consistent improvements over those.

The *regula falsi* seems to perform well on the Headingley network, where also the quadratic method does well, but there seems to be little to choose between the original quadratic method and the *regula falsi* on the other networks. The inclusion in hybrid algorithms results in a similar situation.

The partial step quadratic methods give satisfactory results but, again, do not improve consistently on the methods in the literature, also when started with some MSA iterations. Moreover, the tests on the Chen and Alpha and Sioux Falls network suggest that a partial step of 0.5 could be the best choice but those obtained on the Headingley network are less clear.

4.3.5 Algorithms with Alternative Search Directions

4.3.5.1 Introduction

An alternative way to try to develop efficient SUE algorithms is to use search directions different from the traditional one, used in the MSA.

The alternative search directions proposed and tested in the following sections have been devised interpreting the traditional search direction as steepest descent direction for a preconditioned problem. For clarity, before describing the new search directions proposed, the subject of preconditioning is briefly introduced and the proposed interpretation of the traditional search direction is explained.

4.3.5.2 Preconditioning Optimisation Problems

Preconditioning is a technique used in optimising quadratic and non-quadratic functions to improve the behaviour of the solution algorithms. It corresponds to a change of coordinates aimed at reducing the condition number of the Hessian of a quadratic objective function or of the Hessian of the local quadratic approximation when non-quadratic functions are optimised (see *e.g.* Gill *et al.*, 1981, or Shewchuck, 1994, on which much of the following discussion on preconditioning is based).

The condition number of a matrix is the ratio between the magnitudes of the largest and the smallest of its eigenvalues. It gives summary information about the shape of the function, if this is a quadratic, or of its local quadratic approximation, and, more importantly, about the likely efficiency of solution algorithms.

Taking the case of a scalar quadratic function that can be written as:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c \quad (4.23)$$

a Hessian matrix \mathbf{A} with condition number equal to one implies that all the eigenvalues are equal and that the function is an hypersphere. In this case a steepest descent algorithm (with precise line search) would find the solution in one iteration.

The more distant is the condition number from one, the more the function is different from an hypersphere (and, in two dimensions, the contour lines look like ellipses rather than circles) and, consequently, the more difficult it is for an algorithm employing the steepest descent direction to locate the minimum. In fact, when the condition number is large, a steepest descent algorithm will be subject to a large amount of zigzagging unless it is started along one of the eigenvectors of the Hessian matrix and is used together with a precise line search. The difficulty of solving a problem as a function of the condition number of the Hessian matrix is discussed *e.g.* in Scales (1985) and Shewchuck (1994).

Preconditioning the problem (4.23) consists of applying a change of coordinates so that the condition number of the Hessian matrix in the transformed space is as small as possible and, when feasible, as close as possible to one.

Assuming that a positive definite matrix \mathbf{P} is defined as a preconditioning matrix, the transformation of coordinates is obtained by writing \mathbf{P} as:

$$\mathbf{P} = \mathbf{Q}\mathbf{Q}^T \quad (4.24)$$

by Cholesky decomposition.

\mathbf{Q} defines the change of coordinates that, writing the entities in the transformed space with a dash and those in the original space without, is characterised by the transformation:

$$\mathbf{x}' = \mathbf{Q}\mathbf{x} \quad (4.25)$$

and by the inverse transformation:

$$\mathbf{x} = \mathbf{Q}^{-1}\mathbf{x}' \quad (4.26)$$

In the transformed space the gradient of the function (4.23) results in:

$$\mathbf{g}' = \frac{d\mathbf{x}}{d\mathbf{x}'} \mathbf{g} = \mathbf{Q}^{-1}\mathbf{g} \quad (4.27)$$

and differentiating once more yields:

$$\mathbf{A}' = \mathbf{Q}^{-1} \mathbf{A} \mathbf{Q}^{-T} \quad (4.28)$$

If the eigenvalues of the matrix \mathbf{A}' are clustered and give a smaller condition number than those of \mathbf{A} , the problem in the modified space is simpler to solve, for the same algorithm, than in the original space.

The choice of a good preconditioning matrix is crucial to the simplification of the problem. Ideally, the preconditioning matrix should be the Hessian of the problem, so that a preconditioned Hessian with condition number one will result. In practical problems the diagonal of the Hessian or an approximation to it are often used as they are typically easier to obtain and work with.

When a quadratic function is preconditioned, the preconditioning matrix is constant, as is the Hessian of the function. In the case of non-quadratic functions solved considering them as locally quadratic, the preconditioning applies to the Hessian of the local approximation. Therefore, in the non-quadratic case both the approximated quadratic function and the preconditioner change at each iteration, but the advantage resulting from trying to solve the relevant optimisation problem with this modification is typically retained as the behaviour of the local quadratic approximation to the function is improved, making the problem locally simpler to solve.

4.3.5.3 An Interpretation of the Traditional Search Direction

The search direction used in the MSA and in the other algorithms tested in the previous sections is referred to, here, as the traditional search direction. It is not the steepest descent search direction for (4.3): this, in fact, would be the vector with components (4.5). However, as shown by Sheffi and Powell (1982) it is always a descent direction and its modulus reduces to zero when the solution is reached. Sheffi (1985) noted that the vector linking the current and the auxiliary solution can be read as the steepest descent vector for the program (4.4). However, that is correct when optimising the objective function in terms of costs as in one of the MSA algorithms proposed by Cantarella (1997) rather than, as in the case proposed by

Sheffi (1985), solving the program (4.3) that is the SUE equivalent program in terms of flows.

The traditional search direction for the SUE problem seems to be particularly effective. Even when compared with constrained and unconstrained quasi-Newton search directions based on the function (4.3), that are theoretically known to give good results, it is still the most efficient (Maher, 1998).

The efficiency of the traditional search direction for the SUE problem can be justified if it is seen as an approximate Newton search direction in which the actual Hessian of the local approximation to the objective function is approximated by the Jacobian of the link costs. Equivalently, but in operative terms more interestingly, the traditional search direction can be interpreted as a preconditioned steepest descent direction for the objective function (4.3) where the preconditioner is the Jacobian of the link cost functions, which, having assumed separable and increasing link costs, is diagonal and positive definite.

This interpretation can be explained by building on the discussion on preconditioning in Gill *et al.* (1981) and in Tronrud (1982) and considering that preconditioning with \mathbf{J}_n , the Jacobian of the link cost function at iteration n , is equivalent to introducing a change of coordinates defined by the matrix $\mathbf{J}_n^{\frac{1}{2}}$. Having assumed that the link costs are separable and increasing, their Jacobian is diagonal and positive definite and its inverse \mathbf{J}_n^{-1} exists. Writing the entities in the original space without dash and the entities in the preconditioned space with a dash, the preconditioning of the (local approximation to) the Hessian matrix with the matrix \mathbf{J}_n is equivalent to writing:

$$\mathbf{x}'_n = \mathbf{J}_n^{\frac{1}{2}} \mathbf{x}_n \quad (4.29)$$

$$\mathbf{g}'_n = \frac{d\mathbf{x}_n}{d\mathbf{x}'_n} \mathbf{g}_n = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{g}_n \quad (4.30)$$

$$\mathbf{H}'_n = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{H}_n \mathbf{J}_n^{-\frac{1}{2}} \quad (4.31)$$

The inverse transformations are:

$$\mathbf{x}_n = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{x}'_n \quad (4.32)$$

$$\mathbf{g}_n = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{g}'_n \quad (4.33)$$

$$\mathbf{H}_n = \mathbf{J}_n^{\frac{1}{2}} \mathbf{H}'_n \mathbf{J}_n^{\frac{1}{2}} \quad (4.34)$$

An iteration of an algorithm in the transformed space can be written expressing the next solution \mathbf{x}'_{n+1} as the sum of the present solution \mathbf{x}'_n and of the search vector \mathbf{d}'_n multiplied by the step length λ_n :

$$\mathbf{x}'_{n+1} = \mathbf{x}'_n + \lambda_n \mathbf{d}'_n \quad (4.35)$$

Substituting (4.29) in (4.35) yields:

$$\mathbf{J}_n^{-\frac{1}{2}} \mathbf{x}_{n+1} = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{x}_n + \lambda_n \mathbf{d}'_n \quad (4.36)$$

and multiplying by $\mathbf{J}_n^{\frac{1}{2}}$ gives:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \lambda_n \mathbf{J}_n^{-\frac{1}{2}} \mathbf{d}'_n \quad (4.37)$$

Assuming that the search direction in the transformed space is the steepest descent direction, that is taking:

$$\mathbf{d}'_n = -\mathbf{g}'_n = -\mathbf{J}_n^{-\frac{1}{2}} \mathbf{g}_n \quad (4.38)$$

and substituting it in (4.37) gives:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \lambda_n \mathbf{J}_n^{-1} \mathbf{g}_n \quad (4.39)$$

Thus, rewriting the algorithm working along the steepest descent direction in the modified space as a function of entities of the original space the search direction is:

$$\mathbf{d}_n = -\mathbf{J}_n^{-1} \mathbf{g}_n \quad (4.40)$$

The gradient of the SUE function (4.3) is:

$$\mathbf{g}_n = \mathbf{J}_n (\mathbf{x}_n - \mathbf{y}_n) \quad (4.41)$$

which, substituted in (4.40), gives:

$$\mathbf{d}_n = -\mathbf{J}_n^{-1} \mathbf{g}_n = -\mathbf{J}_n^{-1} \mathbf{J}_n (\mathbf{x}_n - \mathbf{y}_n) = \mathbf{y}_n - \mathbf{x}_n \quad (4.42)$$

which is the traditional search direction for SUE in the original space of the flows.

The Hessian matrix of the SUE objective function (4.3), given by Sheffi and Powell (1982), is:

$$\frac{\partial^2 z_{SUE}(x)}{\partial x_i \partial x_j} = \begin{cases} -\sum_{RS} q_{rs} \sum_K \sum_L \frac{\partial P_{rsk}}{\partial e_{rsl}} \left(\frac{dc_j}{dx_j} \delta_{jl}^{rs} \right) \left(\frac{dc_j}{dx_j} \delta_{jk}^{rs} \right) + \frac{dc_j}{dx_j} + & \text{if } i=j \\ \left(-\sum_{RS} \sum_K q_{rs} P_{rsk} \delta_{jk} + x_j \right) \frac{d^2 c_j}{dx_j} & \\ -\sum_{RS} q_{rs} \sum_K \sum_L \frac{\partial P_{rsk}}{\partial e_{rsl}} \left(\frac{dc_i}{dx_i} \delta_{il} \right) \left(\frac{dc_j}{dx_j} \delta_{jk} \right) & \text{if } i \neq j \end{cases} \quad (4.43)$$

The actual effectiveness of the Jacobian of the link costs, which, as can be seen from (4.43), is a component of the diagonal of the Hessian of the SUE objective function, as a preconditioner for the SUE problem has been investigated verifying numerically the evolution of the condition number of the Hessian of the SUE objective function and of its preconditioned version with the iterations of a SUE algorithm using the traditional search direction.

Table 4.4 reports such condition numbers for an assignment on the Chen and Alpha network and table 4.5 for an assignment on the Sioux Falls network with the set of paths referred to in table 4.2 as S2. The path choice probabilities have been calculated with the Mendell-Elston approximation and their derivatives have been obtained numerically with the method of Ridders (see Press *et al.*, 1992).

Iteration	c.no. of Hessian	c.no. of local preconditioned Hessian
1	*	*
2	*	*
3	$3.350 \cdot 10^4$	10.481
4	$2.087 \cdot 10^4$	6.440
5	$1.632 \cdot 10^4$	4.923
6	$1.487 \cdot 10^4$	4.551
7	$1.410 \cdot 10^4$	4.333
8	$1.376 \cdot 10^4$	4.251
9	$1.354 \cdot 10^4$	4.194
10	$1.344 \cdot 10^4$	4.169
11	$1.340 \cdot 10^4$	4.149
12	$1.355 \cdot 10^4$	4.145
13	$1.333 \cdot 10^4$	4.146
14	$1.333 \cdot 10^4$	4.146

*Table 4.4 – Condition number (c.no.) of the Hessian of the SUE problem and of the locally preconditioned Hessian at each iteration of an assignment carried out with the traditional search direction on Chen and Alpha network, ME approximation. The * indicate the cases in which the Hessian is not definite.*

In both cases it is evident the difference of the condition numbers between the two cases at each iteration. The condition number of the actual objective function, which would control the convergence of a steepest descent algorithm and influence the rate of convergence of algorithms as those explored in Maher (1998), remains large throughout the optimisation process. The condition number of the preconditioned objective function, for the sample of cases tested, is small compared to the other. This may explain the ability of the traditional search direction to solve efficiently SUE problems.

Itn.	c.no. of Hessian	c.no. of local precondition. Hessian	Itn.	c.no. of Hessian	c.no. of local precondition. Hessian
1	*	*	16	$5.860 \cdot 10^4$	8.273
2	$5.620 \cdot 10^5$	196.81	17	$5.876 \cdot 10^4$	8.258
3	$5.727 \cdot 10^5$	38.526	18	$5.875 \cdot 10^4$	8.259
4	$1.428 \cdot 10^5$	14.181	19	$5.875 \cdot 10^4$	8.257
5	$1.108 \cdot 10^5$	11.631	20	$5.892 \cdot 10^4$	8.257
6	$8.023 \cdot 10^4$	9.859	21	$5.892 \cdot 10^4$	8.258
7	$7.185 \cdot 10^4$	9.243	22	$5.873 \cdot 10^4$	8.256
8	$6.543 \cdot 10^4$	8.757	23	$5.890 \cdot 10^4$	8.254
9	$6.275 \cdot 10^4$	8.541	24	$5.890 \cdot 10^4$	8.260
10	$6.060 \cdot 10^4$	8.408	25	$5.890 \cdot 10^4$	8.259
11	$5.960 \cdot 10^4$	8.367	26	$5.890 \cdot 10^4$	8.259
12	$5.906 \cdot 10^4$	8.344	27	$5.890 \cdot 10^4$	8.259
13	$5.870 \cdot 10^4$	8.301	28	$5.890 \cdot 10^4$	8.259
14	$5.862 \cdot 10^4$	8.305	29	$5.890 \cdot 10^4$	8.259
15	$5.860 \cdot 10^4$	8.273			

*Table 4.5 – Condition number (c.no.) of the Hessian of the SUE problem and of the locally preconditioned Hessian at each iteration (itn.) of an assignment carried out with the traditional search direction on the Sioux Falls network, path set S2, ME approximation. The * indicate the cases in which the Hessian is not definite.*

The data used to obtain tables 4.4. and 4.5 also show that the Hessian matrix of the problem has, at the beginning of the solution, some negative eigenvalues and is therefore not definite. This could be expected from the analysis of the Hessian carried out by Sheffi and Powell (1982) who pointed out that, although the Hessian is positive definite at the optimum, its sign is not generally known since the sign of one component of the Hessian matrix depends on the difference between the current and the auxiliary flows.

The full significance of this result and its effect on the algorithms should be investigated further.

However, the effectiveness of the traditional search direction for SUE and its interpretation as a preconditioned steepest descent direction suggest the use of algorithms known to improve on the steepest descent direction and working in the same local change of coordinates.

Different sorts of search directions may be considered as *e.g.* conjugate gradient search directions, quasi-Newton search directions and limited memory quasi-Newton search directions (see Scales, 1985; Gill *et al.*, 1981). All these methods, on a quadratic function, build conjugate search directions (the latter method in an approximate way) using different mechanisms.

In this thesis preconditioned conjugate gradient search directions are considered as they allow us to work with vectors of link flows rather than matrices which is generally simpler and make them suitable also for large problems. Moreover they allow us to follow the evolution of the current solution both in terms of path and link flows as, for a fixed set of paths, the search direction can be calculated equivalently in either ways and the same optimal step applies.

The next section gives an introduction to conjugate gradient algorithms in general and to the actual algorithms used.

4.3.5.4 Preconditioned Conjugate Gradient Search Directions

On a general function, a steepest descent algorithm works along successive directions that are, by definition, orthogonal to each other and therefore may explore directions along which it has already worked at previous iterations. This results in the characteristic zigzagging behaviour it shows at any point of the solution and in the fact that also for quadratic functions the number of iterations necessary to reach the solution cannot be guaranteed unless the algorithm works along eigenvectors of the quadratic function being minimised, in which case it should be expected to converge in a number of iterations equal to the dimension of the problem. Moreover the steepest descent algorithm does not use information on the shape of the function obtained in previous iterations.

A conjugate gradient algorithm is known to improve on a steepest descent one (see e.g. Scales, 1985). In the case of a quadratic function, a conjugate gradient algorithm works iteratively along directions that are conjugate to the Hessian of the function. The conjugate gradient directions are obtained at each iteration simply as a combination of the present gradient and of the previous search directions (the Hessian is not used directly). Such previous directions are linearly independent, therefore once the minimisation along a direction is performed, the gradient during the following iterations will remain orthogonal to that direction. Since a point in the n dimensional space can be reached by a suitable linear combination of at most n linearly independent vectors, in theory and with precise line searches, the optimum of a quadratic problem of dimension n can be found in n iterations.

The general iteration of a conjugate gradient algorithm can be written as that of any other iterative algorithm:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \lambda_n \mathbf{d}_n \quad (4.44)$$

where \mathbf{x}_{n+1} is the solution obtained after the present iteration by moving from the current solution \mathbf{x}_n along the conjugate gradient direction \mathbf{d}_n of a step length λ_n .

The conjugate gradient search direction is:

$$\mathbf{d}_n = -\mathbf{g}_n + \gamma_n \mathbf{d}_{n-1} \quad (4.45)$$

where \mathbf{d}_{n-1} is the previous search direction that is combined with the present gradient \mathbf{g}_n according to the coefficient γ_n . The search direction at the first iteration (or when the algorithm is restarted) is simply the steepest descent direction:

$$\mathbf{d}_0 = -\mathbf{g}_0 \quad (4.46)$$

The coefficient γ_n can be obtained according to different formulae that characterise the different conjugate gradient methods. Two of the possible formulae generally proposed in the literature are the formula of Fletcher and Reeves (1964):

$$\gamma_n = \frac{\mathbf{g}_n^T \mathbf{g}_n}{\mathbf{g}_{n-1}^T \mathbf{g}_{n-1}} \quad (4.47)$$

and the formula of Polak and Ribiere (1969):

$$\gamma_n = \frac{\mathbf{g}_n^T (\mathbf{g}_n - \mathbf{g}_{n-1})}{\mathbf{g}_{n-1}^T \mathbf{g}_{n-1}} \quad (4.48)$$

The two formulae are equivalent in the case of a quadratic function but are not so when the conjugate gradient algorithm is applied to a non-quadratic function.

In fact, with a non-quadratic function the conjugacy of the search directions is referred to the variable Hessian of the local quadratic approximation to the objective function. This, potentially, reduces the efficiency of the algorithms (the more variable is the Hessian, the less the algorithms have the local properties they show with quadratic functions) and may introduce problems. For instance, as mentioned by Scales (1985), the search direction resulting from (4.47) might become orthogonal to the local gradient in which case the γ_n of Fletcher and Reeves would combine such search direction to the local gradient whilst the method of Polak and Ribiere would yield a γ_n equal to zero thus restarting the algorithm and preventing it from possibly getting stuck.

The Fletcher-Reeves method is tested here knowing that it might not be entirely suitable for non-quadratic functions. The Polak-Ribiere formula should give better results and is used with a modification suggested in the relevant literature to ensure its convergence for non-quadratic functions: if the coefficient γ_n is not positive it is taken as zero, that is the algorithm is restarted. Other restarting techniques such as restarting the algorithm periodically (for instance, each time a number of iterations multiple of the dimension of the problem is reached) are not considered here.

Since the aim is to improve on the traditional search direction for solving the SUE problem that has been interpreted as a steepest descent direction in a modified space, conjugate gradient algorithms in a similarly preconditioned space are considered.

As in the previous section, the preconditioning matrix is \mathbf{J}_n , the Jacobian of the increasing and separable link costs at iteration n , and the matrix defining the change of coordinates is $\mathbf{J}_n^{-1/2}$, the diagonal matrix with the i th diagonal entry equal to $\sqrt{\frac{dc_i(x_i)}{dx_i}}$ from the flows \mathbf{x} at iteration n . The transformation of coordinates defined by this matrix has been given in formula (4.29) and the inverse transformation in formula (4.32). The resulting relationships between the gradient in the original and in the transformed space are given by (4.30) and (4.33) and the corresponding Hessian transformations are given by (4.31) and (4.34).

The formulae of the preconditioned conjugate gradient algorithms in the transformed space can be written as:

$$\mathbf{x}'_{n+1} = \mathbf{x}'_n + \lambda_n \mathbf{d}'_n \quad (4.49)$$

$$\mathbf{d}'_n = -\mathbf{g}'_n + \gamma'_n \mathbf{d}'_{n-1} \quad (4.50)$$

with γ'_n given by:

$$\gamma'_n = \frac{\mathbf{g}'_n{}^T \mathbf{g}'_n}{\mathbf{g}'_{n-1}{}^T \mathbf{g}'_{n-1}} \quad (4.51)$$

in the Fletcher-Reeves case, and by:

$$\gamma'_n = \frac{\mathbf{g}'_n{}^T (\mathbf{g}'_n - \mathbf{g}'_{n-1})}{\mathbf{g}'_{n-1}{}^T \mathbf{g}'_{n-1}} \quad (4.52)$$

in the Polak-Ribiere case.

Substituting the transformations (4.29) and (4.30) in the above formulae allows us to write the preconditioned conjugate gradient algorithm as a function of entities of the original space and, ultimately, to write an algorithm that works in the original space. These transformations have been suggested by the discussion on preconditioning in Gill *et al.* (1981) and by the work of Tronrud (1992), who used the Fletcher-Reeves algorithm and analogous transformations for an optimisation problem with a large number of variables in macromolecular crystallography.

The (4.49) can be readily rewritten in the original space assuming that, consistent with (4.32):

$$\mathbf{d}'_n = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{d}_n \quad (4.53)$$

which, substituted along with (4.32) in (4.49), gives:

$$\mathbf{J}_n^{-\frac{1}{2}} \mathbf{x}_{n+1} = \mathbf{J}_n^{-\frac{1}{2}} \mathbf{x}_n + \lambda_n \mathbf{J}_n^{-\frac{1}{2}} \mathbf{d}_n \quad (4.54)$$

and, multiplying by $\mathbf{J}_n^{-\frac{1}{2}}$, the iteration of the algorithm is rewritten in the original space:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \lambda_n \mathbf{d}_n \quad (4.55)$$

Considering (4.30), the formula (4.50), giving the search direction in the transformed space, can also be rewritten as:

$$\mathbf{d}'_n = -\mathbf{J}_n^{-\frac{1}{2}} \mathbf{g}_n + \gamma'_n \mathbf{d}'_{n-1} \quad (4.56)$$

which, multiplying by $\mathbf{J}_n^{-\frac{1}{2}}$, yields:

$$\mathbf{J}_n^{-\frac{1}{2}} \mathbf{d}'_n = -\mathbf{J}_n^{-\frac{1}{2}} \mathbf{J}_n^{-\frac{1}{2}} \mathbf{g}_n + \gamma'_n \mathbf{J}_n^{-\frac{1}{2}} \mathbf{d}'_{n-1} \quad (4.57)$$

Recalling (4.53), this can be rewritten as:

$$\mathbf{d}_n = -\mathbf{J}_n^{-1} \mathbf{g}_n + \gamma'_n \mathbf{d}_{n-1} \quad (4.58)$$

which gives the search direction expressed as a function of vectors in the original space except for the coefficient γ'_{n+1} that can be also written in the original space applying the same transformations. For the Fletcher and Reeves formula this results in:

$$\gamma'_n = \frac{\mathbf{g}_n^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right)^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right) \mathbf{g}_n}{\mathbf{g}_{n-1}^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right)^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right) \mathbf{g}_{n-1}} \quad (4.59)$$

whilst for the Polak-Ribiere method it results:

$$\gamma'_n = \frac{\mathbf{g}_n^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right)^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right) (\mathbf{g}_n - \mathbf{g}_{n-1})}{\mathbf{g}_{n-1}^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right)^T \left(\mathbf{J}_n^{-\frac{1}{2}} \right) \mathbf{g}_{n-1}} \quad (4.60)$$

Thus formulae (4.55), (4.58) and one of the (4.59), (4.60), allow us to use a preconditioned conjugate gradient algorithm working in the original space of the link flows on the network.

It is interesting to notice that although the γ_n must be calculated using quantities referred to the links, the search directions can be expressed either in terms of link flows or path flows (for a predefined and fixed link-path incidence matrix), thus the algorithm can produce updates of the solution both in terms of path and link flows. However, the algorithms used here to account for path and link flows can be used also for pure link-based models.

To ensure the conjugacy of the search directions, conjugate gradient algorithms should be implemented with precise line search. This is valid also when non-quadratic functions are optimised. However, the two preconditioned conjugate gradient algorithms described above have been tested not only with precise line searches carried out with the quadratic interpolation method but also using line searches refined until an improved gradient or objective function is found and algorithms performing one line search only at each iteration similarly to the cases discussed for the traditional search direction. Using unrefined line searches may introduce a further element of approximation to the conjugacy of the search direction, adding to the effect of solving non-quadratic functions with a method ideally for quadratic functions, but it is included to save on overall computational effort. Moreover, as a safety device, the dot product of the gradient and of the new

direction is calculated at each iteration and, if it is not negative, the preconditioned conjugate gradient direction is discarded and restarted.

From the point of view of the practical application of the algorithms it should be noted that links with possible zero flows will give a zero contribution to the Jacobian of the link costs thus giving problems with the application of the method. Using a fixed set of paths, however, allows us to exclude from the optimisation the links not traversed by any paths (but this is not necessary in the cases examined). Moreover, using a stochastic choice model, the flows on all paths should be greater than zero and thus should be the flows on all links traversed by at least one path. It should be recalled that the application of these algorithms to the Headingley network is heuristic as this network contains connectors that have been excluded from the calculations of γ'_n .

4.3.5.5 Application of the Precise Preconditioned Fletcher-Reeves and Polak-Ribiere Search Directions

Numerical tests applying the precise Fletcher-Reeves (ppFR) and Polak-Ribiere (ppPR) search directions have been carried out using the quadratic line search introduced in section 4.3.3.2. The results confirm that algorithms using such search directions explore the solution space of the problem examined better than the corresponding ones using the traditional search direction as a smaller number of iterations is required to reach the same convergence threshold. Figures 4.31, 4.32 and 4.33 report examples of the convergence trends obtained and show that, either the convergence trends of the ppFR and ppPR practically coincide or there is little to choose between the two especially in comparison with the traditional search direction.

Looking at the number of equivalent loadings necessary to reach the same convergence threshold, which give a more correct indication of the computational effort, the Fletcher-Reeves and the Polak-Ribiere search directions with precise steps calculation are still more efficient than the corresponding algorithms working along the traditional search direction and refined to the same precision.

The actual number of loadings employed by the algorithms refining the steps to a given precision depends on such precision, but looking at steps refined to 10^{-3} , these algorithms result in some cases as efficient or even more efficient than algorithms that do not refine the step determination along the traditional search direction.

The results are different on different networks, as shown in the examples reported in figures 4.34, 4.35 and 4.36.

On the simpler Chen and Alpha network not refining the step along the traditional search direction is clearly the best strategy amongst those compared, both overall and at the start of the calculations. On the Sioux Falls network the new algorithms improve on some of those examined in the previous parts of this chapter but often cannot outperform hybrid algorithms starting with a number of MSA iterations. Moreover the precise step ppFR and ppPR algorithms do not improve on the others at the beginning of the assignment, as shown in figure 4.35.

On the Headingley network in many cases the ppFR and ppPR algorithms improve on both the algorithms working along the traditional search direction with non-refined line search used from the beginning and hybrid algorithms. Moreover, the ppFR and ppPR algorithms typically improve on the convergence trend of the others from the beginning.

The results on the efficiency in terms of loadings, although giving indications not consistent across the networks, suggest that the precise preconditioned conjugate gradient algorithms could be, at least in some cases, an efficient method to solve the SUE problem and encourage an attempt to look into alternative ways of implementing them such as those explored in the next sections.

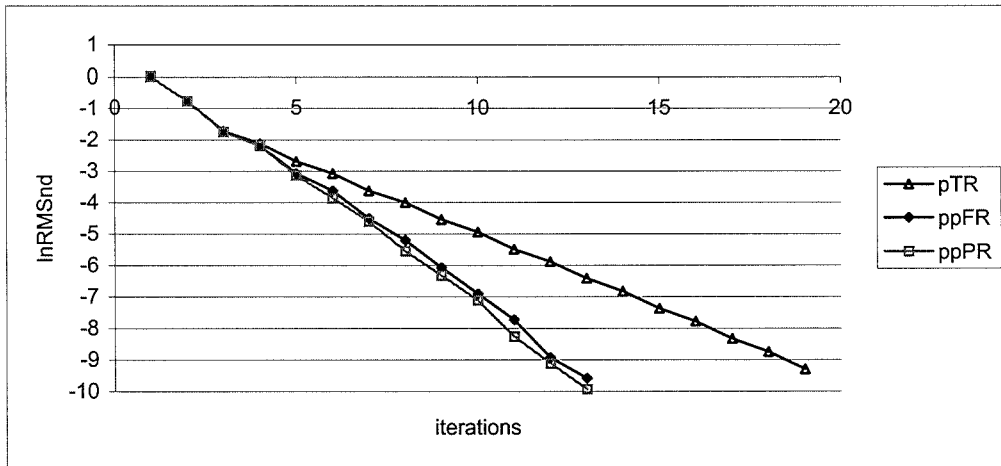


Fig. 4.31 – Comparison of the convergence trend for algorithms using different search directions (TR: traditional; ppFR: precise preconditioned Fletcher Reeves, ppPR: precise preconditioned Polak Ribiere) and step determined to a precision of 10^{-3} . Chen and Alpha network, Mendell-Elston approximation.

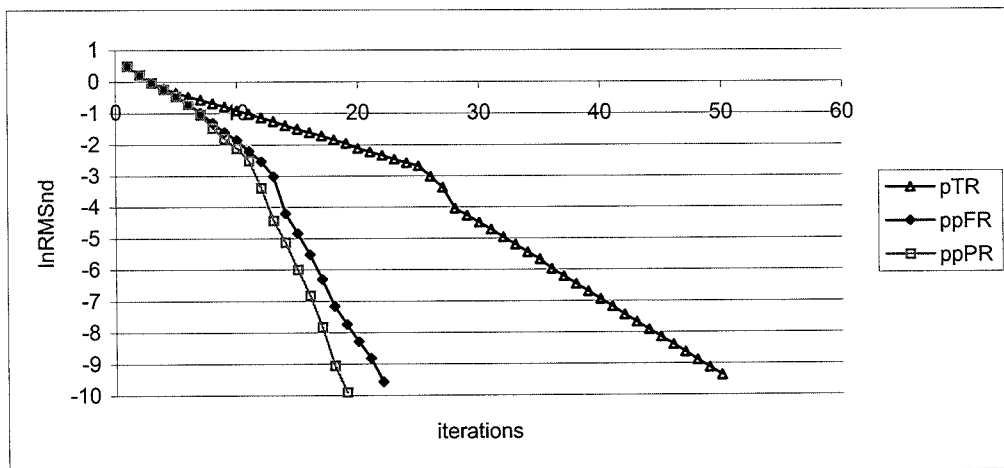


Fig. 4.32 – Comparison of the convergence trend for algorithms using different search directions (TR: traditional; ppFR: precise preconditioned Fletcher Reeves, ppPR: precise preconditioned Polak Ribiere) and step determined to a precision of 10^{-3} on the Sioux Falls network, path set S2, Mendell-Elston approximation.

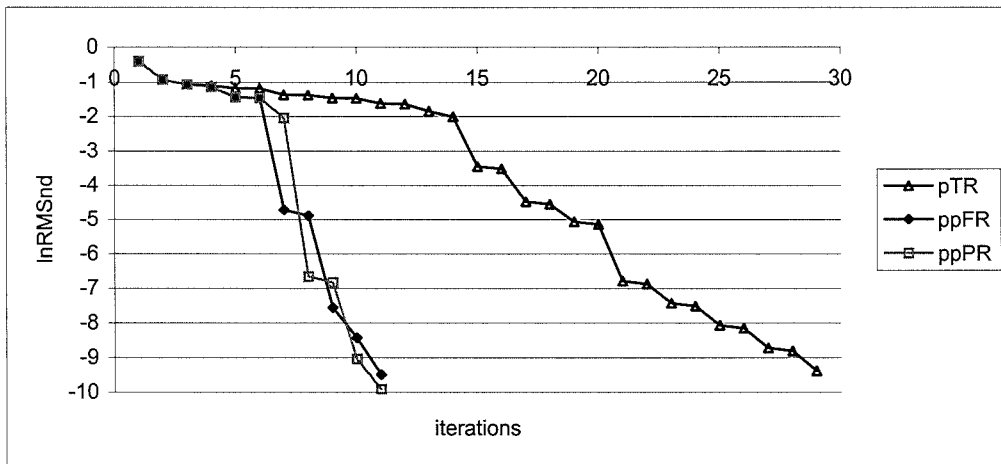


Fig. 4.33 – Comparison of the convergence trend for algorithms using different search directions (TR: traditional; ppFR: precise preconditioned Fletcher Reeves, ppPR: precise preconditioned Polak Ribiere) and step determined to a precision of 10^{-3} on the Headingley network, path set H1, Clark approximation.

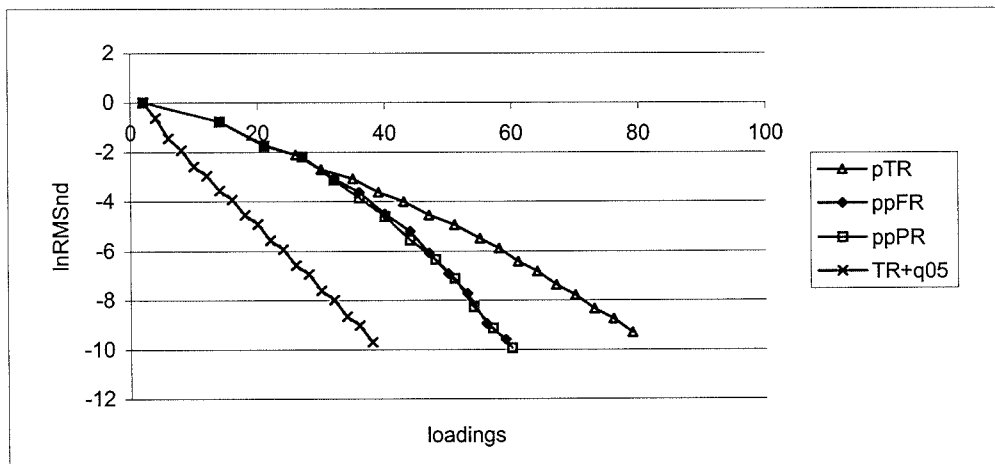


Fig. 4.34 – Comparison of the convergence trend of algorithms using different search directions (TR: traditional; ppFR: precise preconditioned Fletcher Reeves, ppPR: precise preconditioned Polak Ribiere) and step determined to a precision of 10^{-3} and of an algorithm working along the traditional search direction and calculating the step with the partial step quadratic interpolation method (TR q05). Chen and Alpha network, Mendell-Elston approximation.

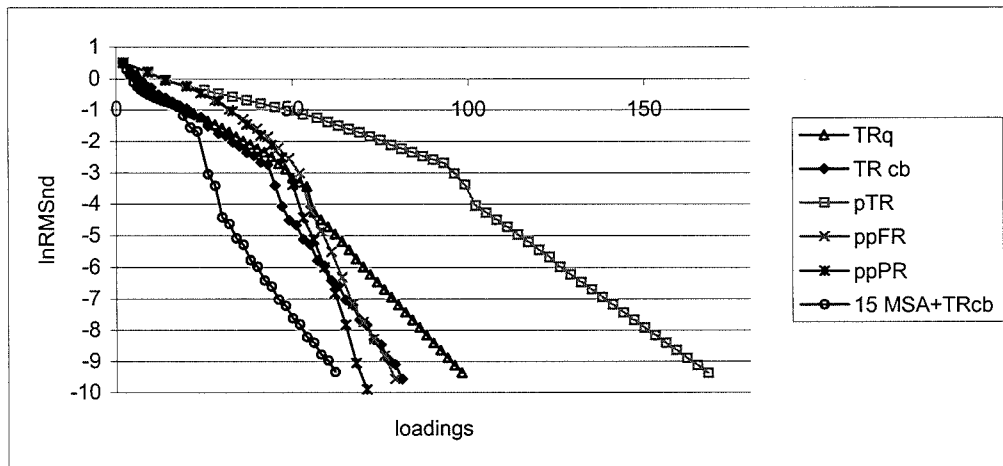


Fig. 4.35 – Comparison of the convergence trend of algorithms using different search directions (TR: traditional; ppFR: precise preconditioned Fletcher Reeves, ppPR: precise preconditioned Polak Ribiere) and step determined to a precision of 10^{-3} (prefix p), algorithms working along the traditional search direction and using the quadratic (TRq) and cubic (TRcb) interpolation methods and a hybrid algorithm based on TRcb. Sioux Falls network, path set S2, Mendell-Elston approximation.

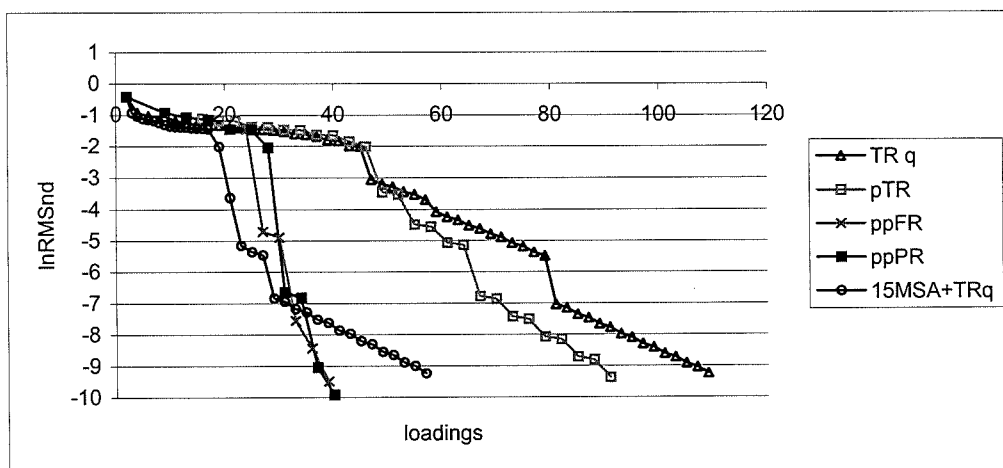


Fig. 4.36 – Comparison of the convergence trend of algorithms using different search directions (TR: traditional; ppFR: precise preconditioned Fletcher Reeves, ppPR: precise preconditioned Polak Ribiere) and step determined to a precision of 10^{-3} (prefix p), an algorithm working along the traditional search direction and using the quadratic (TRq) interpolation method and a hybrid algorithm based on TRq. Headingley network, path set H1, Clark approximation.

4.3.5.6 Application of the Approximate Preconditioned Fletcher-Reeves Search Direction

As mentioned above conjugate gradient search direction algorithms should be applied with a precise line search to guarantee the conjugacy of the successive search directions. However, such conjugacy in the cases examined here is approximate as the preconditioner changes at each iteration, at least at the beginning of the assignment and until the flows on the links become more stable.

The algorithms tested in this section, and referred to as approximate preconditioned Fletcher-Reeves (apFR), use line searches with step calculation carried out once or, alternatively, until a gradient or objective function improvement is ensured along the search direction given by the preconditioned Fletcher-Reeves formula. Using such line searches may introduce a further element of approximation, although checking that an improved gradient or objective function has been found should be expected to be beneficial to the algorithm behaviour.

All the function interpolation methods tested for the traditional search direction have been tested and compared (quadratic, cubic, *regula falsi*, rational, partial step quadratic).

The convergence trends resulting from algorithms finding the step length interpolating once only or returning it only after an improved gradient or objective function is found, coincide on the Chen and Alpha and on the Sioux Falls network, with both MNP approximations employed. However, there are cases for which the trends do not coincide on the Headingley network but, from the results obtained, there seem to be little difference between the trends of the two types of algorithm. Thus, the algorithms checking for an improvement are considered in the following comparisons on the grounds that they are more consistent with the requirements of the line search to be implemented in a conjugate gradient algorithm.

On the Chen and Alpha network, the apFR algorithms with alternative line search methods give similar trends as shown in fig. 4.37, with the exception of the algorithm with quadratic line search which gives an oscillating behaviour that delays the eventual convergence. Fig. 4.37, also suggests that the rational interpolation

method is less efficient than the others, due to the additional loading it requires at each iteration. A comparison between the apFR algorithms and the corresponding ones using the traditional search direction shows that the latter ones are outperformed from the start of the optimisation. A small selection of these comparisons is reported in figs. 4.42 and 4.43.

Also on the Sioux Falls network and on the Headingley network the apFR algorithm used with any of the optimised step lengths investigated above (quadratic, cubic, *regula falsi*, rational, partial step quadratic) performs consistently better than the corresponding algorithms employing the traditional search direction as can be noticed in the examples in figs. 4.38 and 4.39.

Some convergence plots, such as the quadratic one reported in fig. 4.38 and the cubic in fig. 4.41, show that in some cases the algorithms using the cubic and the quadratic line search show an oscillating trend before eventually converging, especially on the Headingley network. The cause of these oscillations has not been further investigated in this study, but it is interesting to note that line search methods using information taken closer to the present solution are not subject to such problems. Moreover these oscillations can be easily avoided by restarting the algorithm, that is anyway approximated, each time one such oscillation of the convergence trend is detected. This device has not been applied to the algorithms whose results are reported.

The *regula falsi* and partial step quadratic are the most efficient line search methods in most cases on both the Sioux Falls and the Headingley network, although sometimes only marginally. A selection of the comparisons with the other methods is reported in figs. 4.39, 4.40 and 4.41. However, the fact that they have not been subject to the possible oscillation mentioned above suggests them as the best performing algorithms with this search direction. It is interesting that the use of information gathered close to the current solution to perform the approximation leads both to safer and more efficient algorithms.

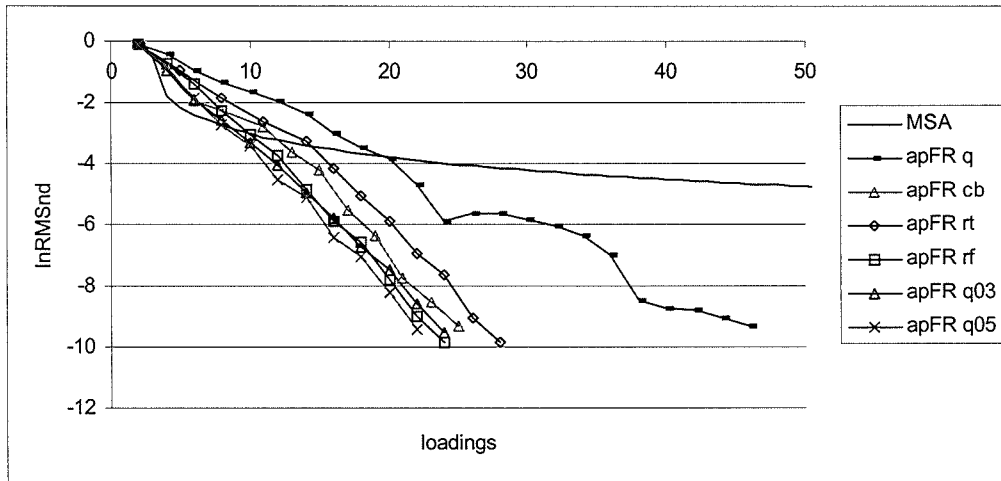


Fig. 4.37 – Comparison of the MSA and algorithms using the apFR search direction (*q*: quadratic, *cb* cubic, *rt*: rational, *rf* regula falsi, *q03* quadratic partial step 03, *q05* quadratic partial step 05). Chen and Alpha network, Clark approximation.

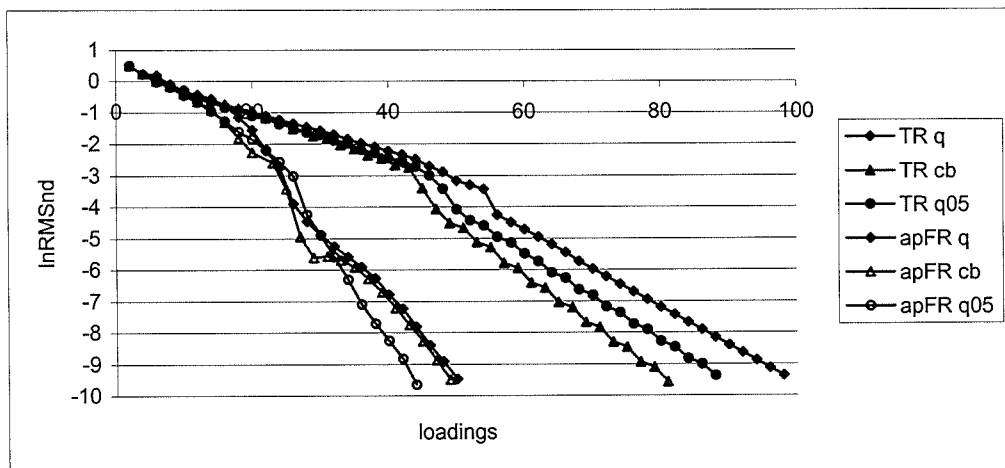


Fig. 4.38 – Comparison of algorithms using the traditional (TR) and the apFR search direction (*q*: quadratic, *cb* cubic, *q05* quadratic partial step 0.5). Sioux Falls network, path set S2, Mendell-Elston approximation.

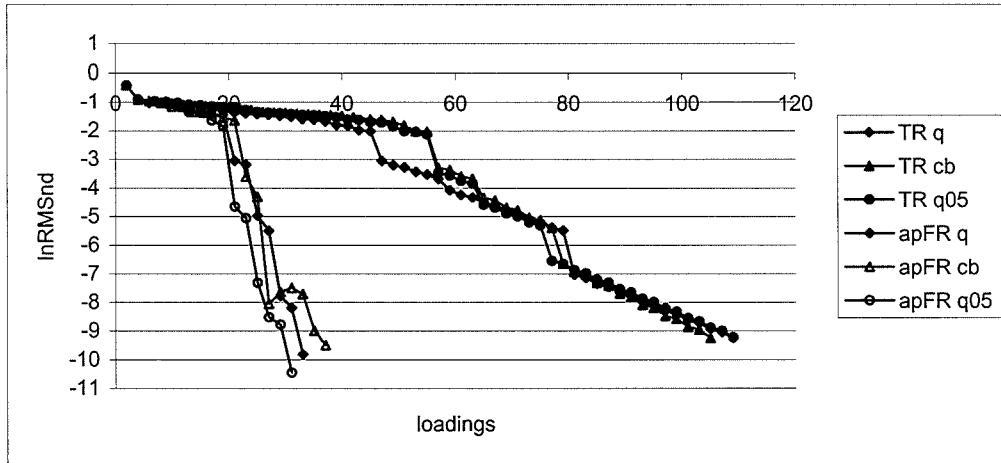


Fig. 4.39 – Comparison of algorithms using the traditional and the apFR search direction (*q*: quadratic, *cb* cubic, *q05* quadratic partial step 05). Headingley network, path set S1, Clark approximation.

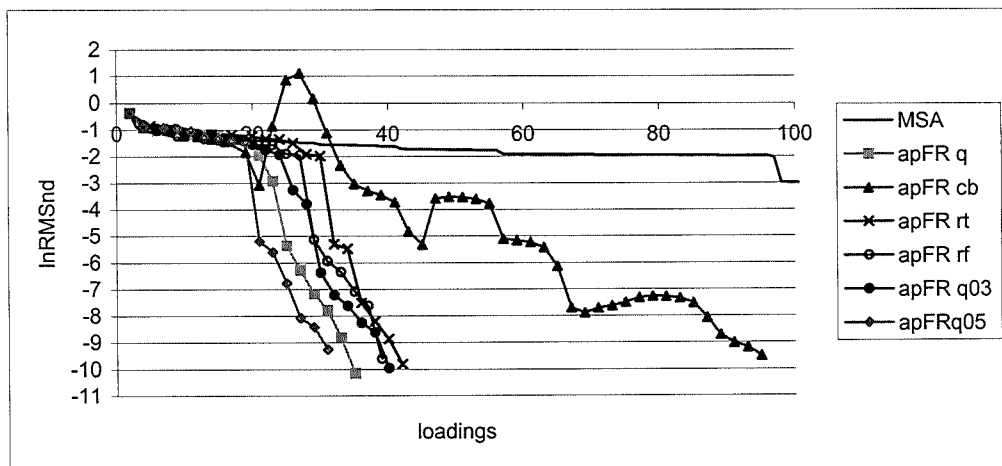


Fig. 4.40 – Comparison of the MSA and algorithms using the apFR search direction (*q*: quadratic, *cb* cubic, *rt*: rational, *rf* regula falsi, *q03* quadratic partial step 03, *q05* quadratic partial step 05). Headingley network, path set H1, Mendell-Elston approximation.

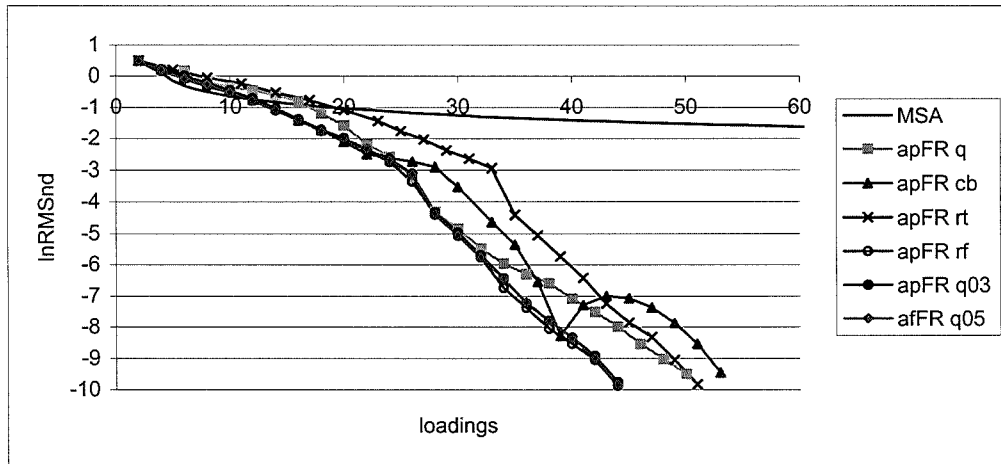


Fig. 4.41 – Comparison of the MSA and algorithms using the apFR search direction (*q*: quadratic, *cb* cubic, *rt*: rational, *rf* regula falsi, *q03* quadratic partial step 03, *q05* quadratic partial step 05). Sioux Falls network, path set *S2*, Clark approximation.

As with the traditional search direction, the initial good efficiency of the MSA suggests the development of hybrid algorithms, started with a number of MSA iterations, then performing an iteration along the traditional search direction to start the apFR algorithm and continuing with the latter. Algorithms started with 5, 10, 15 and 20 MSA iterations have been tested.

On the Chen and Alpha network a limited number of MSA iterations to start (five, and in some cases ten) improve on the convergence behaviour of the apFR algorithms employed from the start as shown in the examples in figs. 4.42 and 4.43. The initial oscillatory behaviour recorded with the quadratic search on the apFR used from the outset is obviated by starting the algorithm with some MSA iterations (compare figs. 4.37 and 4.44).

A comparison of the different line search methods shows that, on the Chen and Alpha network, those based on the linear approximation to the gradient behave very similarly to each other, and are marginally more efficient than those based on the cubic and rational interpolation. In particular the rational interpolation method gives less efficient convergence trends although the difference is of very few loadings, consistent with the results obtained using the traditional search direction.

Also on the Sioux Falls and on the Headingley network starting with some MSA iterations is beneficial to the efficiency of the algorithms with all line search methods. Figures 4.45, 4.46 report a selection of the results supporting these considerations. Moreover hybrid algorithms reduce or eliminate the oscillations recorded in some cases using the quadratic or cubic line search methods as can be seen from the same figures. Once more it is difficult to indicate a general number of MSA iterations to start the algorithms with. However, the experience gathered from these investigations suggests that starting with 5 or 10 MSA iterations might not give the best possible performance but generally improves on the use of line search from the outset.

It is also interesting to note that, using hybrid algorithms on the Sioux Falls and on the Headingley networks, the best performing line search methods are those interpolating the gradient as linear but using data taken close to the present solution: the *regula falsi* and the partial step quadratic methods typically give the most efficient convergence trends, as shown in the examples in figs. 4.45, 4.46, 4.47 and 4.48. However, when a large number of MSA (15 or 20) is used to start the algorithms there is little to choose between the different line search methods, although the rational interpolation method, requiring one more loading at each iteration performs slightly worse than the others.

Also when started with a number of MSA iterations the apFR algorithms perform better than the corresponding ones using the traditional search direction as reported in figs. 4.47 and 4.48. The improvement is clear with any number of MSA to start on the Chen and Alpha and the Sioux Falls network whilst, on the Headingley network, when more than 10 MSA are used, the methods using the different search directions have practically the same efficiency.

Before closing the discussion on the apFR algorithms, it should be noted that the apFR without refined step do better than those with the refined step on the Chen and Alpha and the Sioux Falls networks and do similarly to the refined step algorithms on the Headingley network.

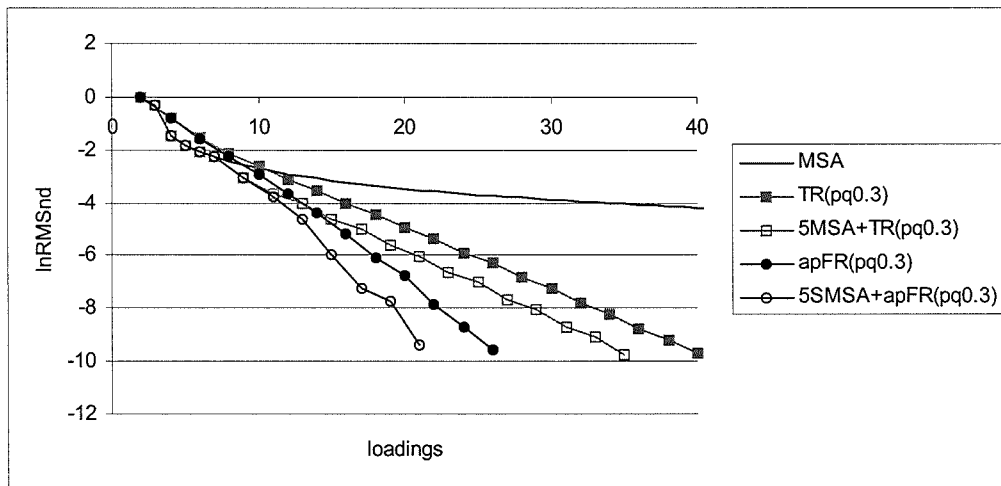


Fig. 4.42 – Comparison of the MSA and corresponding algorithms using the traditional (TR) and the apFR search direction (q03 quadratic interpolation partial step 03), and hybrid algorithms started with 5 MSA iterations. Chen and Alpha network, Mendell-Elston approximation.

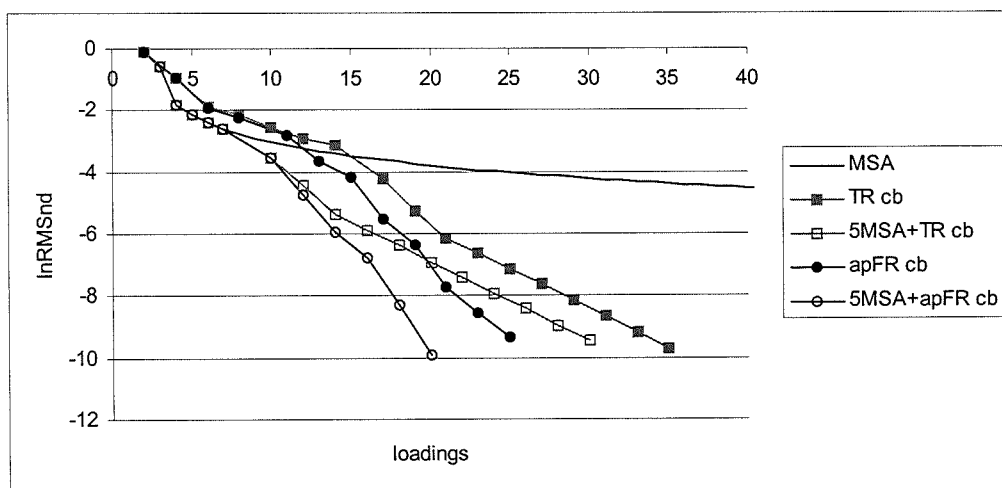


Fig. 4.43 – Comparison of the MSA, corresponding algorithms using the traditional (TR) and the apFR search direction (cb cubic interpolation) and hybrid algorithms started with 5 MSA iterations. Chen and Alpha network, Clark approximation.

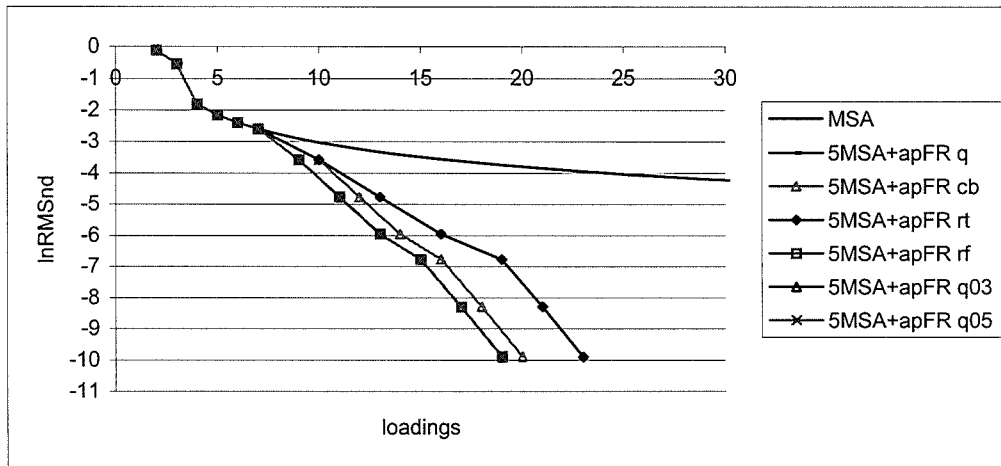


Fig. 4.44 – Comparison of the MSA and hybrid algorithms started with 5 MSA iterations before using the apFR search direction (q: quadratic, cb cubic, rt: rational, rf regula falsi, q03 quadratic partial step 03, q05 quadratic partial step 05). Chen and Alpha network, Clark approximation.

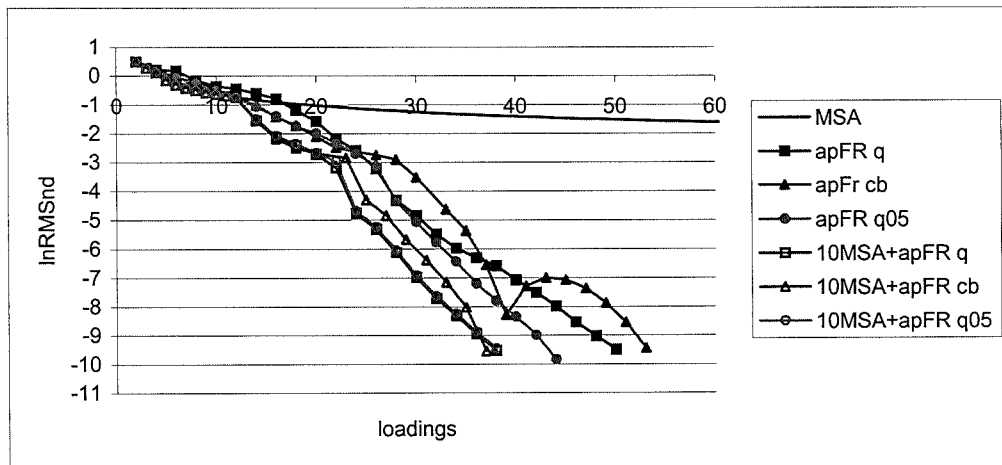


Fig. 4.45 – Comparison of the MSA, algorithms using the apFR search direction (q: quadratic, cb cubic, q05 quadratic partial step 0.5) and corresponding hybrid algorithms started with 10 MSA iterations. Sioux Falls network, path set S5, Clark approximation.

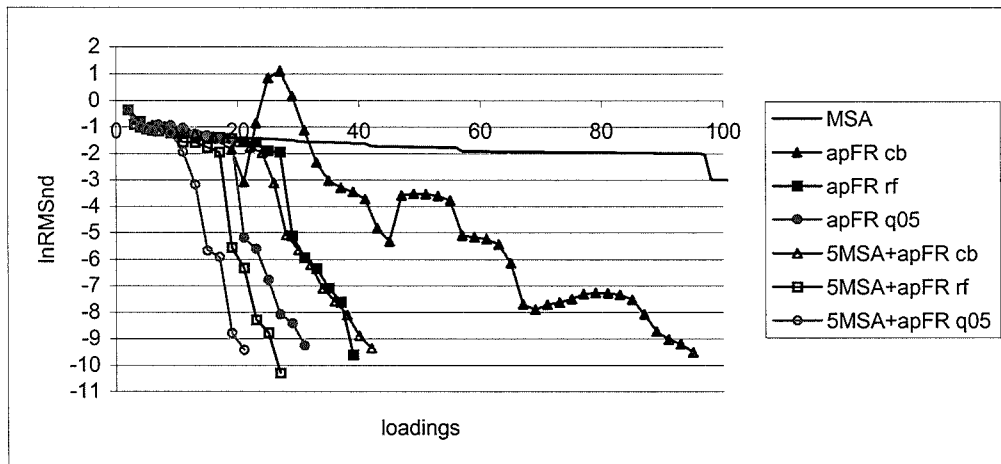


Fig. 4.46 – Comparison of the MSA, algorithms using the apFR search direction (*q*: quadratic, *rf* regula falsi, *q05* quadratic partial step 0.5) and corresponding hybrid algorithms started with 5 MSA iterations. Headingley network, path set H1, Mendell-Elston approximation.

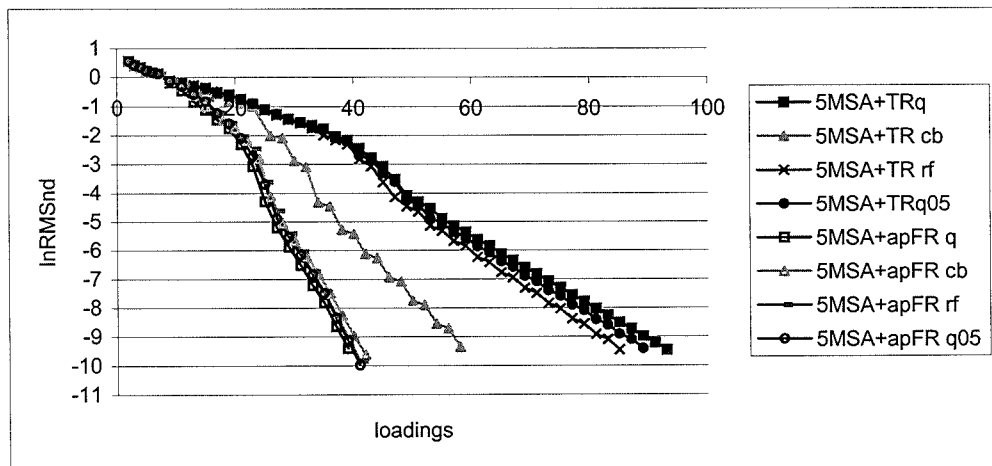


Fig. 4.47 – Comparison of hybrid algorithms using the traditional (TR) and the apFR search direction (*q*: quadratic, *cb* cubic, *rf* regula falsi, *q05* quadratic partial step 0.5) started with 5 MSA iterations. Sioux Falls network, path set S1, Mendell-Elston approximation.

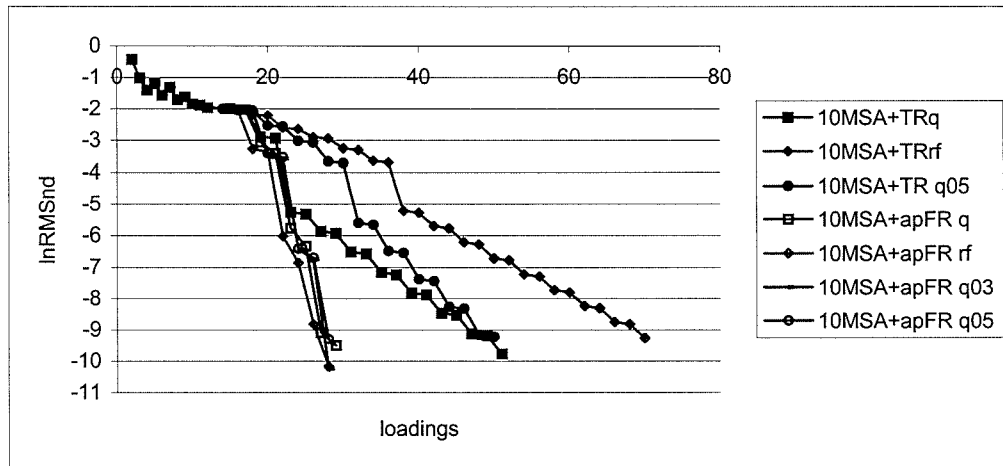


Fig. 4.48 – Comparison of hybrid algorithms using the traditional (TR) and the apFR search direction (*q*: quadratic, *cb* cubic, *rf* regula falsi, *q03* quadratic partial step 0.33, *q05* quadratic partial step 0.5) started with 10 MSA iterations. Headingley network, path set H2, Mendell-Elston approximation.

4.3.5.7 Application of the Approximate Preconditioned Polak-Ribiere Search Direction

The algorithms tested in this section combine the search direction given by the preconditioned Polak-Ribiere algorithm with line search methods interpolating once only or checking that an improved gradient or objective function has been found before returning the step. The algorithms thus obtained are referred to as approximate preconditioned Polak-Ribiere algorithms (apPR). All the function interpolation methods tested for the traditional search direction have been tested also with the apPR search direction (quadratic, cubic, *regula falsi*, rational, partial step quadratic).

Notwithstanding the possible approximation introduced by not refining the line searches it has been checked that, on the test networks employed, the apPR algorithms typically perform better than refined step methods (the verification has been limited to the quadratic and cubic interpolation cases).

Also in the apPR case, on the Chen and Alpha and on the Sioux Falls networks, the methods returning the step after a single interpolation and those checking that an improved gradient or objective function is found, give coinciding convergence

trends. They are different in some cases on the Headingley network, as noted already with other search directions. There seem to be typically minor differences between the behaviour of the two sorts of algorithms and, as in the apFR case, those checking for the improvement are considered in the following comparison on the grounds that they are more consistent with the requirements of conjugate gradient algorithms.

Observing the convergence trends on the Chen and Alpha network, the methods based on the partial step quadratic interpolation perform slightly better than the others from the outset, though the differences are small, as can be seen from the data reported in figure 4.49. The same figure shows how the quadratic interpolation method is outperformed by all the others but does not show the oscillating trend seen in the apFR case.

Moreover, on the Chen and Alpha network, the apPR algorithms are seen to be more efficient than the corresponding algorithms working along the traditional search direction. Two examples of these different convergence trends are reported in figure 4.50 and 4.51.

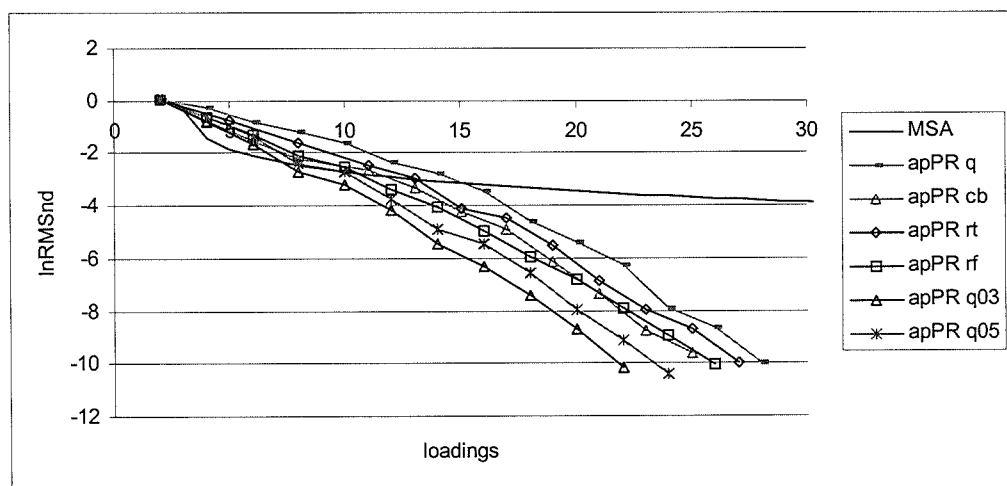


Fig. 4.49 – Comparison of the MSA and algorithms using the apPR search direction (q: quadratic, cb cubic, rt: rational, rf regula falsi, q03 quadratic partial step 0.33, q05 quadratic partial step 0.5). Chen and Alpha network, Mendell-Elston approximation.

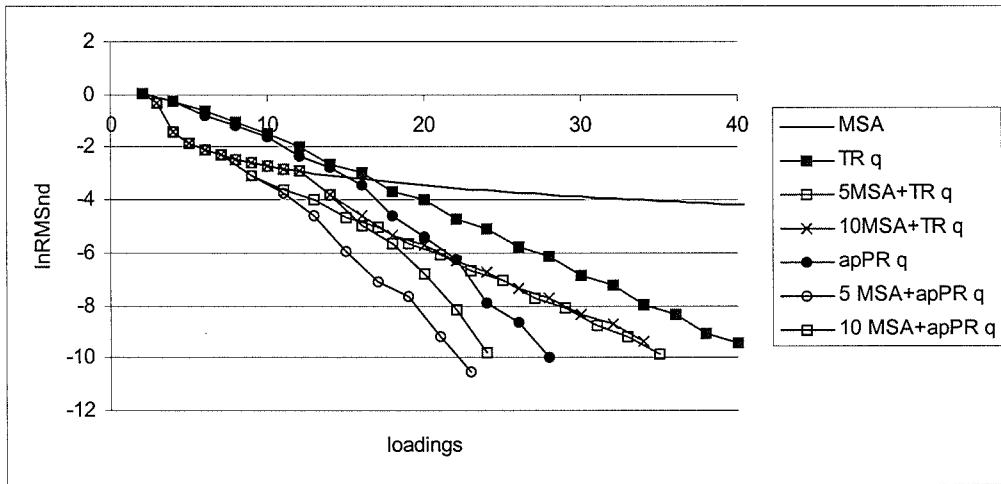


Fig. 4.50 – Comparison of the MSA, corresponding algorithms using the traditional (TR) and the apPR search direction (q quadratic interpolation) and hybrid algorithms. Chen and Alpha network, Mendell-Elston approximation.

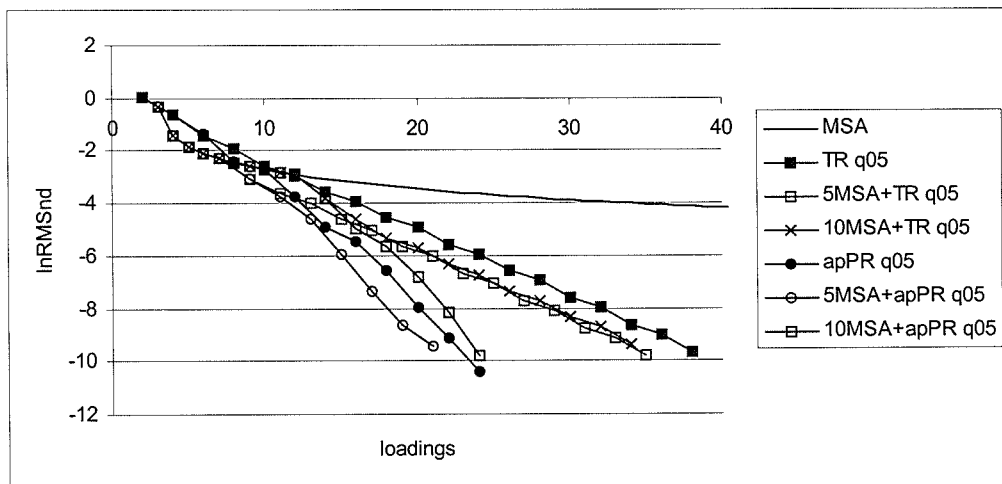


Fig. 4.51 – Comparison of the MSA, corresponding algorithms using the traditional (TR) and the apPR search direction ($q05$ quadratic partial step interpolation) and hybrid algorithms. Chen and Alpha network, Mendell-Elston approximation.

Looking at the data for the two larger networks (Sioux Falls and Headingley), a comparison of the different step determination methods suggests that in all cases those based on the quadratic interpolation perform best (see fig. 4.52). The rational interpolation consistently performs worse than the other algorithms.

Also on the Sioux Falls and on the Headingley network the algorithms using the apPR search direction perform consistently better than those using the traditional search direction from the start of the calculations. Examples of convergence trends showing this are given in figures 4.53, 4.54 and 4.55. There are cases of minor oscillations of the convergence trends, that are of lesser importance than in the apFR case, but they do not hamper the convergence. In fact fig. 4.54 shows one such case where the apPR *regula falsi* algorithm, although showing some oscillations, does better than the corresponding method using the traditional search direction. The absence of the oscillations that had been recorded for the apFR convergence trends, can probably be explained by the ability of the PR algorithm to restart itself when the gradient is orthogonal to the search direction or when γ' is negative.

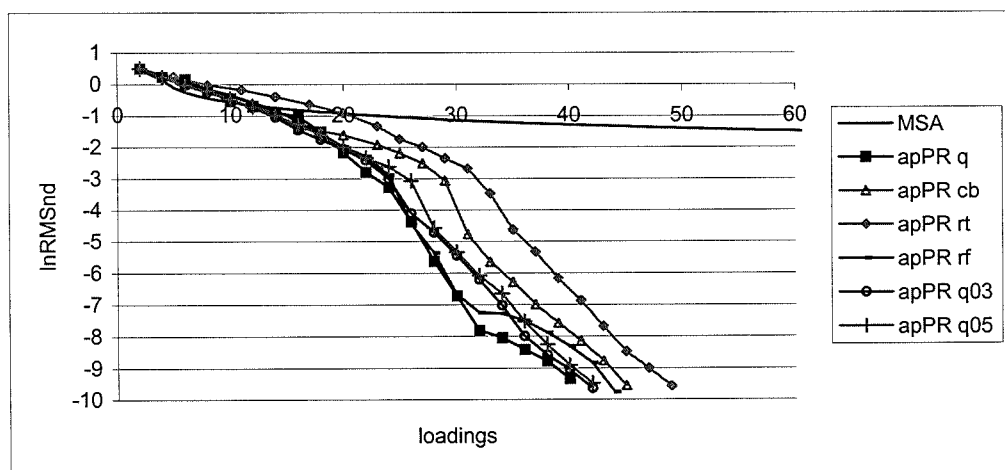


Fig. 4.52 – Comparison of the MSA and algorithms using the apPR search direction (q: quadratic, cb cubic, rt: rational, rf regula falsi, q03 quadratic partial step 0.33, q05 quadratic partial step 0.5). Sioux Falls network, path set S2, Mendell-Elston approximation.

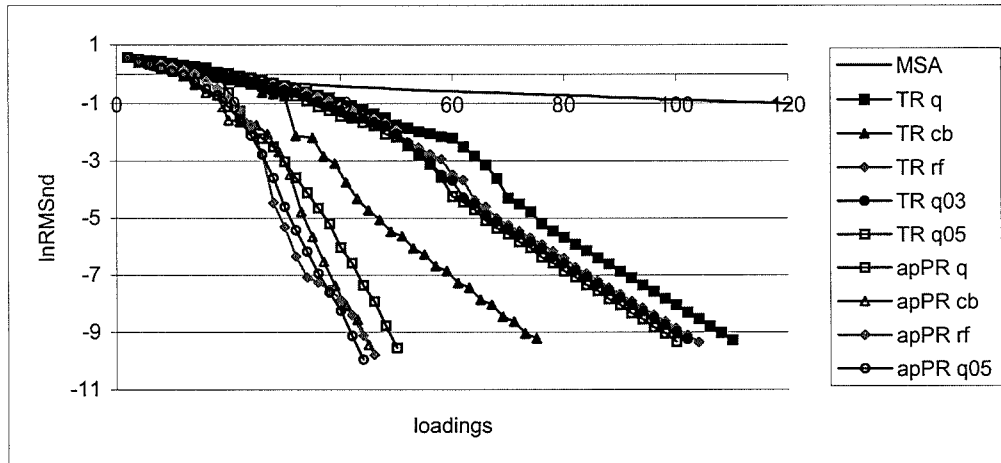


Fig. 4.53 – Comparison of the MSA and algorithms using traditional and the apPR search direction (*q*: quadratic, *cb* cubic, *rt*: rational, *rf* regula falsi, *q03* quadratic partial step 0.33, *q05* quadratic partial step 0.5). Sioux Falls network, path set S1, Mendell-Elston approximation.

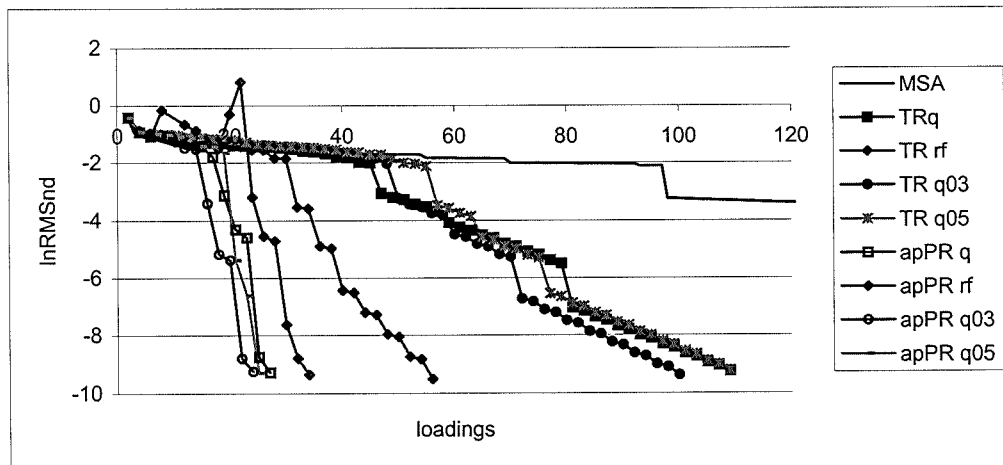


Fig. 4.54 – Comparison of the MSA and corresponding algorithms using the traditional (TR) and the apPR search direction (*q*: quadratic, *rf* regula falsi, *q03* quadratic partial step 0.33, *q05* quadratic partial step 0.5). Headingley network, path set H1, Clark approximation.

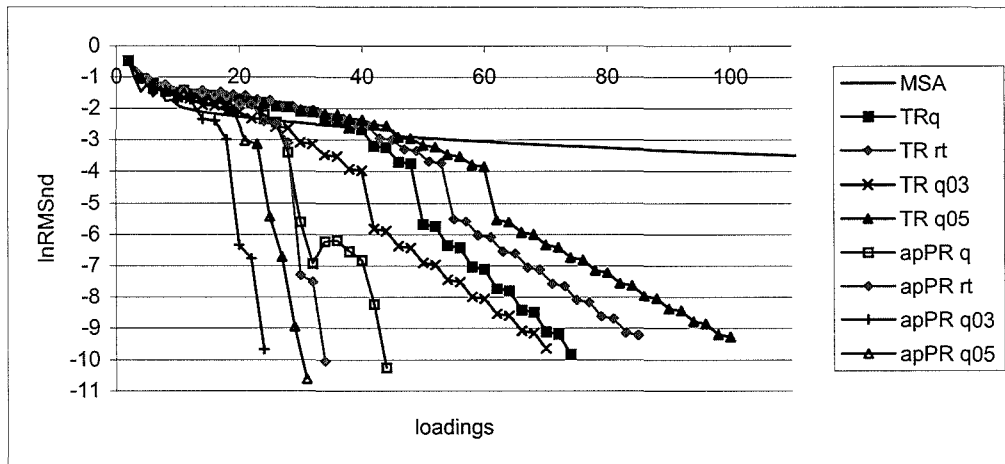


Fig. 4.55 – Comparison of the MSA and corresponding algorithms using the traditional (TR) and the apPR search direction (q : quadratic, rt rational, $q03$ quadratic partial step 0.33, $q05$ quadratic partial step 0.5). Headingley network, Path set H2, Clark approximation.

The figures above and other similar results show that the MSA also does better than the apPR methods during the first iterations. Once more, this suggests the use of hybrid algorithms that begin to solve the problem with a number of MSA iterations, then perform an iteration along the traditional search direction to initialise the actual apPR algorithm used. Algorithms started with 5, 10, 15 and 20 MSA iterations have been tested.

On the Chen and Alpha network, with the cubic and the partial step quadratic methods the improvement given by the use of hybrid algorithms is absent or very limited (this is shown for the partial step quadratic in fig. 4.51 and for the cubic in fig. 4.56), whilst with the other line search methods a limited number of MSA iterations to start is beneficial, as shown in figure 4.52, although using more than 10 initial MSA iterations leads to slower convergence trends than with the original algorithms.

This behaviour is also suggested by the fact that the MSA does better than the other algorithms only for very few iterations, although the effectiveness of starting the algorithms with some MSA iterations does not seem to be related in a simple way to how good a convergence statistic the initial MSA iterations produce.

Still on the Chen and Alpha network, there is little to choose amongst the different apPR methods started with the same number of MSA iterations although the rational interpolation does slightly less well. Moreover, hybrid algorithms including the apPR search direction do better than the corresponding ones based on the traditional search direction (see figs. 4.50 and 4.51).

On the Sioux Falls and Headingley networks, the use of some MSA iterations to start the algorithms eliminates the oscillations recorded in some cases when the apPR direction is used from the beginning (see fig. 4.57). The effect of the MSA start device on the efficiency seems limited or absent in many cases on the Headingley network, as shown in fig. 4.58 but is clearly important from the tests on the Sioux Falls network, as shown by fig. 4.59. The numerical tests also showed that, with some MSA iterations to start, the best performing methods are those based on quadratic interpolation, although there is little to choose except for the rational interpolation performing consistently slightly worse than the others.

The remark on the difficulty to determine a best number of MSA iterations to start the hybrid algorithm is also relevant here, although, as mentioned before, five or ten iterations eliminate the oscillation problems and generally give an improved convergence.

However, it has been found that hybrid apPR algorithms perform consistently better than hybrid algorithms using the traditional search direction as shown in fig. 4.60 and 4.61.

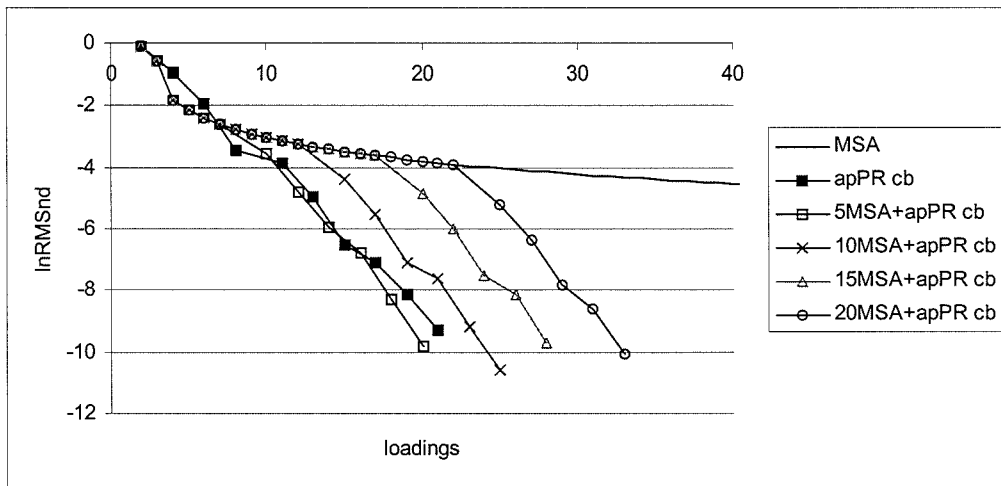


Fig. 4.56 – Comparison of the MSA, the apPR algorithm with cubic line search (apPR cb) and hybrid algorithms. Chen and Alpha network, Clark approximation.

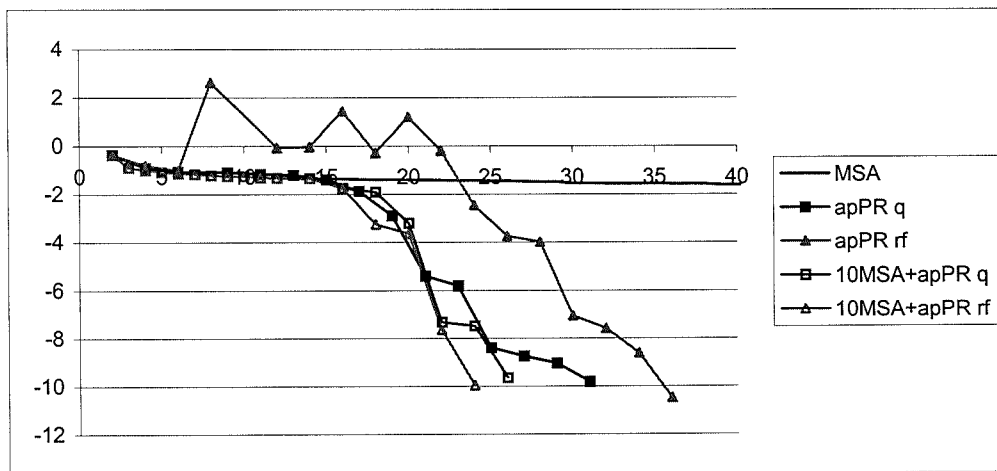


Fig. 4.57 – Comparison of the MSA and corresponding algorithms using the traditional (TR) and the apPR search direction (q: quadratic, rf regula falsi). Headingley network, path set H2, Mendell-Elston approximation.

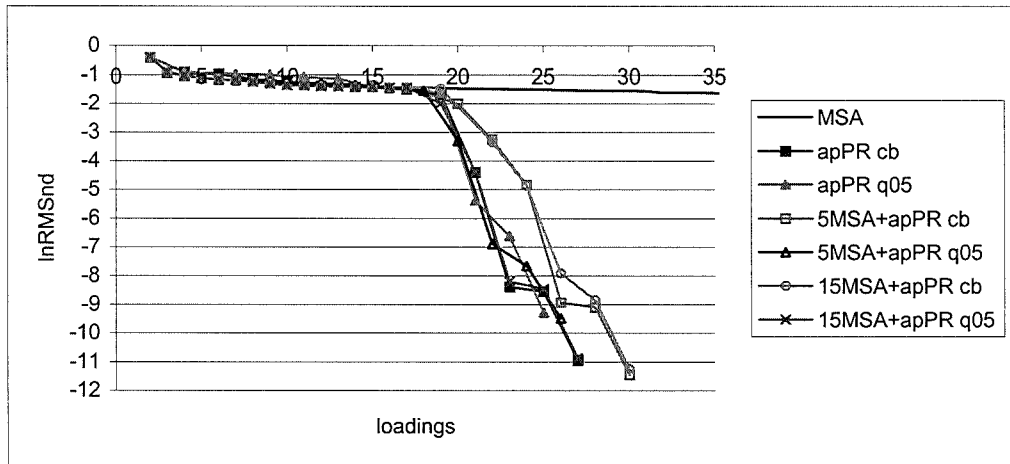


Fig. 4.58 – Comparison of MSA, algorithms using the apPR search direction (cb cubic, q05 quadratic partial step 0.5) and corresponding hybrid algorithms started with 5 and 15 MSA iterations. Headingley network, path set H1, Clark approximation.

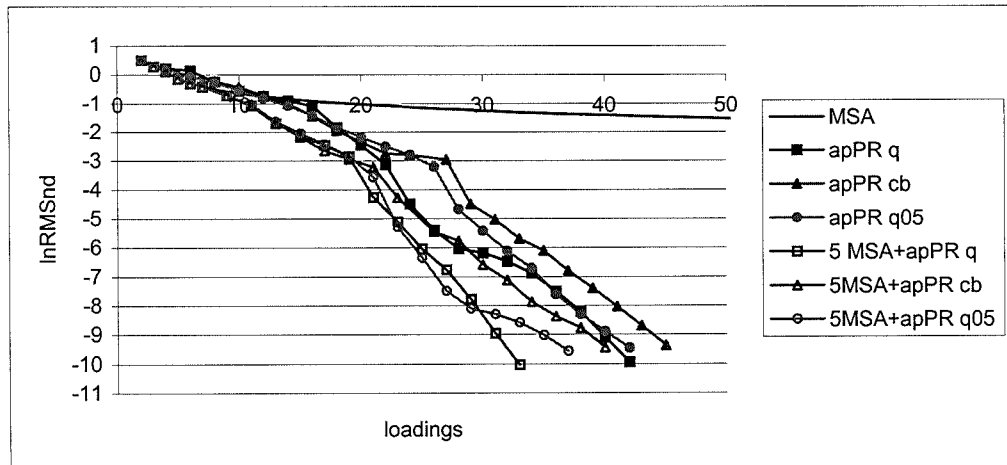


Fig. 4.59 – Comparison of MSA, algorithms using the apPR search direction (q: quadratic, cb cubic, q05 quadratic partial step 0.5) and corresponding hybrid algorithms started with 5 MSA iterations. Sioux Falls network, path set S2, Clark approximation.

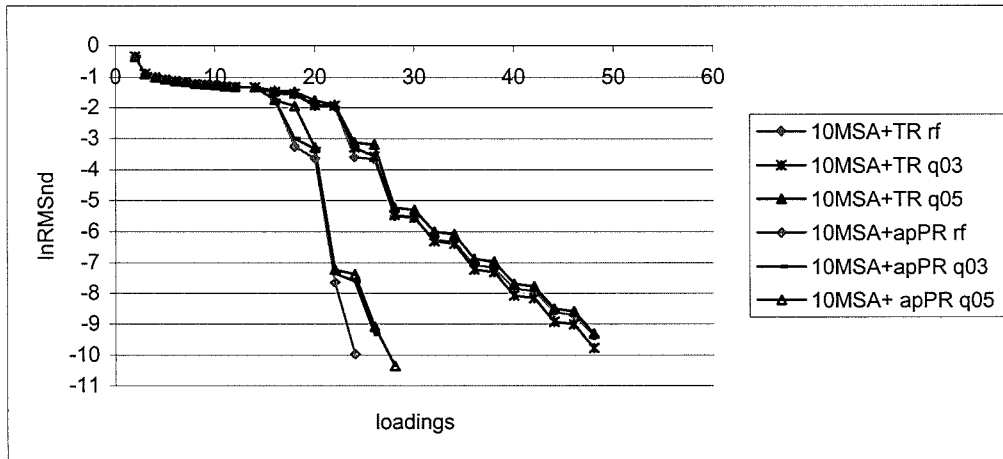


Fig. 4.60 - Comparison of corresponding hybrid algorithms started with 10 MSA iterations before using the traditional (TR) and the apPR search direction (rf regula falsi, q03 quadratic partial step 0.33, q05 quadratic partial step 0.5). Headingley network, path set H1, Mendell-Elston approximation.

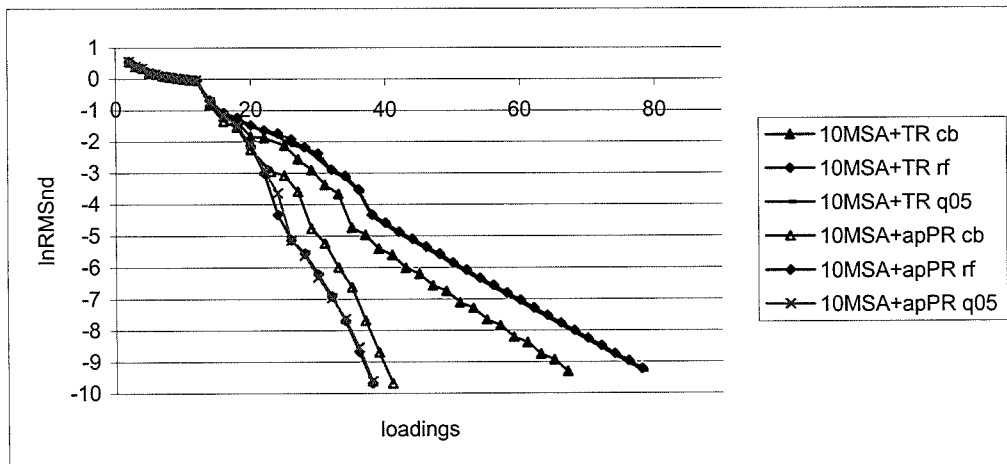


Fig. 4.61 - Comparison of the MSA and corresponding algorithms using the traditional (TR) and the apPR search direction (cb: cubic, rf regula falsi, q05 quadratic partial step 0.5). Sioux Falls network, path set S1, Clark approximation.

4.3.5.8 Comparison of Approximate Preconditioned Conjugate Gradient Algorithms and Conclusions

The figures reported above and the other results collected showed that using either approximate preconditioned conjugate gradient search directions improves significantly on the convergence trends given by the traditional search direction.

Looking at the overall convergence efficiency of the different methods with the different search directions on the Chen and Alpha network, there is little to choose between the apFR and the apPR search directions, although the latter performs slightly better especially when the partial step quadratic interpolation method or the cubic interpolation methods are used. It is also interesting to note that, when the apFR and apPR algorithms are started with a number of MSA iterations their trends practically coincide, which shows that the initial MSA steps, on this network, take the algorithms through the part of the calculation where the function is less easily approximated by a quadratic.

Comparing the convergence trends of the algorithms using the traditional and the apFR and apPR search directions for the Sioux Falls and the Headingley network, show that both the preconditioned conjugate gradient methods improve significantly on the traditional search direction. There would be little difference between the two preconditioned conjugate gradient methods if the apFR were not prone to oscillations in some cases. This suggests that the apPR should be preferred. When a number of MSA iterations is used to start the algorithms, and the oscillation problem is eliminated, the difference in performance between the preconditioned conjugate gradient methods is much smaller than that between them and the traditional search direction.

It is also interesting to note that the line search methods based on the quadratic interpolation perform consistently well for all the cases investigated. In some cases on the Headingley network good performances are also obtained using cubic interpolation. However, the good convergence efficiency obtainable with the quadratic and the *regula falsi* methods and, particularly, with the partial step quadratic methods can be verified in all cases.

Since the advantage of using the apFR and, especially, the apPR search direction is consistently present also when the algorithms are started with a number of MSA iterations (although in this case it is reduced in general and lost for the rational interpolation method) it is possible to suggest that, in general it is of advantage over the other methods investigated in this thesis to implement an algorithm using an unrefined line search based on the linear interpolation of the gradient along directions obtained with the apPR algorithm.

4.4 Conclusions

This chapter has looked at methods for the solution of the stochastic user equilibrium (SUE) multinomial probit traffic assignment problem in a path based implementation, when the paths are fixed at the outset of the assignment.

An introduction to the traffic assignment problem has described the main differences between different models for traffic assignment, also defining SUE. A review of the models in the literature for UE, SNL and SUE both in the link based and the path based framework has followed a review of the actual path sets considered in such models, which has been included as, often, one criticism to path-based models concerns the way paths through networks are chosen.

Several algorithms for traffic assignment have been tested on three test networks.

First, the Method of Successive Averages (MSA) and the algorithms proposed by Maher and Hughes (1997a) have been tested finding, as in the results given by Maher and Hughes, that the MSA is outperformed by methods using the same search direction but optimising the step length along it by interpolating the objective function with a quadratic or a cubic function either once or until a better gradient or objective function is found. The results obtained with the Mendell-Elston and the improved Clark approximation have been found to be rather similar. In general it seems that algorithms started with a number of MSA iterations before using optimised step lengths are the best performing ones, again consistent with the results of Maher and Hughes.

To explore the possibility of improving on the efficiency of the methods of Maher and Hughes, a number of alternative step calculation methods have been investigated. The MWA, Method of Weighted Averages, which is a variation of the MSA, gave interesting results, outperforming the MSA and in some cases also the optimised step algorithms. However, which one of the different MWAs investigated performs best seems to depend on the particular problem.

Three alternative step interpolation methods have been proposed, two of them, the *regula falsi* and the partial step quadratic method, approximate the gradient of the objective function by a linear function along the search direction, whilst the other approximates it with a rational function. All of these methods could be used as alternatives to those proposed by Maher and Hughes, but the tests carried out do not give any indication that they are able to do consistently better. Thus they cannot be proposed as an improvement on the algorithms in the literature.

To try and obtain more efficient algorithms it has been proposed to read the traditional search direction, used in the MSA and in the other algorithms just mentioned and given by the vector linking the current and the auxiliary solution, as a steepest descent direction in a preconditioned space, that is after the problem is considered with a change of coordinates defined by the square root of the Jacobian of the link costs. Starting from this interpretation of the traditional search direction, two preconditioned conjugate gradient algorithms, using respectively the formula of Fletcher and Reeves and the formula of Polak and Ribiere, have been proposed and tested. Used with line search determining the step to a given precision they resulted more efficient than the corresponding algorithms using the traditional search direction, but not always more efficient than the algorithms using the traditional search direction and unrefined line search and related hybrid algorithms. Approximate preconditioned Fletcher-Reeves and Polak-Ribiere algorithms using unrefined line search have then been tested. The efficiency results obtained are interesting for both types of algorithms but the ones obtained using the Fletcher and Reeves formula present some instabilities of the convergence trends, due to the unrefined line searches and possibly to the fact that such an algorithm is less suitable for not exactly quadratic functions. The instabilities are obviated by starting the algorithm with some MSA iterations but, more importantly, they are less important

or absent when the Polak and Ribiere formula is used, which has a built-in ability to restart when it runs into problems. The results obtained with the Polak and Ribiere formula are consistently better than those using the same optimised line search methods with the traditional search direction, also when the algorithms are started with a number of MSA iterations. Thus this sort of algorithm, which performs well especially when used along with line search methods that interpolate the gradient as linear, can be proposed as a possible efficient alternative to the optimised step algorithms in the literature.

In closing this chapter it should be noted that, although the investigations have been carried out with multinomial probit approximations and in a path-based case with paths fixed from the outset of the assignment, they could be used also with other choice models solved analytically (and the MWA also when they are solved by simulation) and in a link-based framework. Their performance in such cases, however, should be further verified.

5. EXTENSIONS OF SUE: MULTIPLE USER CLASSES AND ELASTIC DEMAND

5.1 Introduction

The Stochastic User Equilibrium (SUE) models considered in the previous chapter rest on the assumption that all the drivers have the same characteristics and perceive the network costs in the same way, except for the variations accommodated by the choice model error terms, and model only the choice of route through a network. The present chapter considers models with a more detailed representation of the drivers by including multiple user classes and models employing elastic demand to account in an aggregate way for transport related choices other than that of route.

The chapter is organised as follows. First, the inclusion of Multiple User Classes (MUC) in SUE models is discussed, with a review of the literature, the description of the objective function and of the algorithms that can be used to solve the MUC SUE problem, followed by the results of the tests of the algorithms proposed. Then, SUE with elastic demand (SUE ED) is introduced by a review of the literature, a new objective function is proposed and discussed and algorithms are suggested. The latter are not tested in the SUE ED case but similar algorithms are tested in the Multiple User Classes Stochastic User Equilibrium (MUC SUE ED) case, introduced next, for which objective functions are derived and solution algorithms, that are extensions of those for SUE ED, are proposed and tested. In all cases, paths are accounted explicitly and enumerated at the outset of the assignment. However, the same objective functions and similar algorithms can be applied to the link-based case. Moreover, the results reported have been obtained using the algorithms with the Mendell-Elston and the Clark approximation for MNP but they could be applied also using other choice models solved analytically.

5.2 Stochastic User Equilibrium Models with Multiple User Classes (MUC SUE)

5.2.1 Introduction

The SUE models considered in chapter 4 deal with drivers that are assumed to perceive network costs according to the same distribution. Thus, the drivers modelled constitute a homogeneous group with respect to their choice behaviour or, in other words, a single user class. This is a fairly common assumption in modelling practice.

However, in some applications it is important and possible (in terms of data availability) to distinguish amongst several classes of users (that are groups of users), each with different cost functions and/or different cost variability perceptions.

For instance, models with multiple user classes have been used to assess the effect of having drivers making decisions on the basis of information of different accuracy as when only a proportion of them has vehicles equipped with route guidance devices (see *e.g.* Van Vuren and Watling, 1991; Maher and Hughes, 1996a). Furthermore, using different cost functions allows us to consider not only different sorts of drivers (*e.g.* drivers attaching a different weight to elements of the generalised cost) but also limits on network usage imposed on some users or on some sorts of vehicles (as *e.g.* HGVs).

Multiple User Classes (MUC) models include two or more groups of users (or market segments, according to the choice model terminology introduced in section 2.4.5), each representing a homogeneous group with respect to the definition of the perceived generalised costs and each separately in equilibrium. The MUC-SUE conditions can be simply expressed as the SUE ones (given by Daganzo and Sheffi, 1977):

at (MUC)SUE no user can improve his *perceived* travel cost by unilaterally changing route

since this definition accommodates also different groups of perceived costs.

5.2.2 MUC SUE Models in the Literature

Several studies in the literature are concerned with MUC within the SUE framework.

Van Vuren and Watling (1991) considered, at the same time, classes in User Equilibrium, User Optimum and logit SUE in a link-based model backed theoretically by the results of Daganzo (1982). They solved the model using an adaptation of SATURN (Van Vliet and Hall, 1993) to carry out the stochastic network loadings and the MSA algorithm to find the equilibrium point.

Maher and Hughes (1996a) proposed an analytical link-based probit model accounting for any number of user classes each with a different perception error of the costs or with the same perception error but costs that are multiples of each other (the two settings are equivalent because of the scalability of random utility choice models). They used it to assess the effect of having drivers with different perceptions of the variability of the costs because of the introduction of route guidance devices. Their model is solved using an extension of the Sheffi and Powell (1982) objective function for SUE that is minimised using optimised line search techniques similar to those implemented for SAM with a single user class (Maher and Hughes, 1997a). Although the model has been implemented with the Markovian link based SAM probit model, it can be used with any choice model and also in a path-based setting. In fact, Maher (1998) suggested applying the same ideas to link-based logit MUC SUE.

A more general MUC SUE model has been proposed by Daganzo (1982) and is reviewed in detail in the next section since it provides the framework on which the algorithms proposed here and the elastic demand case treated in section 5.4 are based.

5.2.3 The MUC SUE Model of Daganzo (1982)

Daganzo (1982) considered a rather general MUC SUE setting with users subject to different costs, not necessarily multiples of each other, and different perceptions. He proposed an objective function assuming separable link costs and suggested solving the probit MUC SUE problem using simulation and the MSA. In a later paper

Daganzo (1983) extended further this framework to include non-separable link costs and elastic demand using a fixed-point formulation rather than an equivalent mathematical program.

The MUC framework put forward by Daganzo (1982) considers K user classes each with its OD matrix and cost distribution. In addition to the link flows by user class, Daganzo considered also standardised link flows, similar to those used in traffic engineering to account for vehicles of different characteristics. The standardised flow v_i on link i is:

$$v_i = \sum_K \alpha^{(k)} x_i^{(k)} \quad (5.1)$$

where $\alpha^{(k)}$ is the positive coefficient to standardise the flow of vehicles of the user class k and $x_i^{(k)}$ is the flow of the vehicles of class k on link i .

The cost $c_i^{(k)}$ on a link i and for the user class k is defined as:

$$c_i^{(k)} = c_{0i}^{(k)} + \beta^{(k)} b_i(v_i) \quad (5.2)$$

that is as the sum of a fixed part $c_{0i}^{(k)}$, if need be different by class, and a flow dependent part obtained as the product of a positive coefficient $\beta^{(k)}$, typical of each user class, and a strictly increasing function $b_i(v_i)$ common to all user classes and depending on the standardised flow v_i . The link costs are assumed separable and such an assumption is maintained here. Moreover, path costs are assumed to be additive.

Using the above definitions of flows and costs Daganzo (1982) suggested to solve for MUC SUE by minimising the objective function:

$$z_{MUCSUE} = - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} S_{rs}^{(k)} + Z(\mathbf{b}) \quad (5.3)$$

where k is one of the K user classes, rs is one of the RS OD pairs, $q_{rs}^{(k)}$ is the flow of class k between the OD pair rs , $S_{rs}^{(k)}$ is the satisfaction (the expectation of the minimum perceived cost) for a user of class k travelling between r and s , and $\alpha^{(k)}$,

$\beta^{(k)}$, are as defined for (5.1) and (5.2). $Z(\mathbf{b})$ is a scalar function whose gradient is the vector of the inverse of the common link costs \mathbf{b} .

The objective function (5.3) is independent of the choice model used and the related optimisation program can be solved as unconstrained as the flow conservation constraints are naturally satisfied at equilibrium. It can be explicitly written in two alternative forms, depending on whether a formulation based on common link costs or on standard link flows is preferred.

When the formulation in terms of common link costs is used, which is also suggested by the definition of $Z(\mathbf{b})$ given by Daganzo, (5.3) becomes:

$$z(\mathbf{b})_{MUCSUE} = -\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} S_{rs}^{(k)}(\mathbf{b}) + \sum_I \int_0^{\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}}} b_i^{-1}(\omega) d\omega \quad (5.4)$$

where i is one of the I links, $b_i^{-1}(\omega)$ is the inverse of the link cost function $b_i(z_i)$ appearing in the link cost expression, $c_i^{(k)}$ and $c_{0i}^{(k)}$ are as defined for (5.1) and (5.2) whilst the other symbols are as for (5.3). The upper integration limits for the integrals of the inverse of the link cost function are the same for all the user classes on a link, so in the objective function they are expressed using the values relative to the first user class, which is always present.

Alternatively, with a change of variable, (5.3) can be written in terms of standard link flows:

$$z(\mathbf{v})_{MUCSUE} = -\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} S_{rs}^{(k)}(\mathbf{v}) + \sum_I v_i b_i(v_i) - \sum_I \int_0^{v_i} b_i(\omega) d\omega \quad (5.5)$$

where the symbols are as for (5.4).

As can be easily verified, when only one user class with link costs as in (5.2) is considered, the MUC SUE objective function becomes equivalent to the SUE objective function of Sheffi and Powell (1982) in the corresponding formulation (link flows or link costs).

The equivalence of the minimum of the MUC SUE objective function to the MUC SUE conditions for a network and its uniqueness are discussed and proved in Daganzo (1982).

However, the proof of equivalence is also given in the next section, as it is useful to introduce the derivation of the search directions considered in this chapter. Also the proof of the uniqueness of the MUC SUE solution is given in the following sections analysing the Hessian of the two versions of the objective function, similarly to the analysis carried out in the SUE case by Sheffi and Powell (1982), as it is important to establish the uniqueness of the minimum of (5.5) that here is used to develop solution algorithms. Daganzo (1982) deduced the convexity of (5.4) simply from the convexity of its components but did not extend the discussion to a programme in terms of standardised flows.

5.2.3.1 The Gradient of the MUC SUE Objective Function

Working along the lines of the derivation of the gradient of (4.4) reported in Sheffi and Powell (1982) it is possible to derive the gradient of the MUC SUE objective function (5.4) with respect to \mathbf{b} the vector of common parts of the link costs.

To organise the discussion, the objective function (5.3) can be rewritten as the sum of two terms:

$$z_{MUCSUE} = z' + z'' \quad (5.6)$$

where, consistent with (5.3),

$$z' = - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} S_{rs}^{(k)} \quad (5.7)$$

(the independent variable for the satisfaction is omitted to simplify the notation)

$$z'' = \sum_I \frac{c_i^{(i)} - c_{0i}^{(i)}}{\beta^{(i)}} \int_0^1 b_i^{-1}(\omega) d\omega \quad (5.8)$$

Starting by considering z' the derivative with respect to a link common cost b_i is:

$$\frac{\partial z'}{\partial b_i} = - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} \frac{\partial S_{rs}^{(k)}}{\partial b_i} \quad (5.9)$$

and the derivative of the satisfaction of travelling between the OD pair rs with respect to b_i can be obtained using the chain rule as:

$$\frac{\partial S_{rs}^{(k)}}{\partial b_i} = \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial b_i} \quad (5.10)$$

where $e_p^{(k)}$ is the cost of path p between the OD pair rs for user class k .

The derivative of the satisfaction of travelling between the OD pair rs with respect to the costs of travelling on any of the paths between them is simply:

$$\frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} = P_{rsp}^{(k)} \quad (5.11)$$

where $P_{rsp}^{(k)}$ is the choice probability of path p between r and s for the user class k , as results from the properties of the satisfaction function (see *e.g.* Sheffi, 1985, pp. 269-270).

The costs of travelling along path p for the users of the class k (considering additive path costs) can be written explicitly as:

$$e_p^{(k)} = \sum_I (c_{0i}^{(k)} + \beta^{(k)} b_i) \delta_{pi} \quad (5.12)$$

therefore its derivative with respect to b_i is:

$$\frac{\partial e_p^{(k)}}{\partial b_i} = \frac{\partial}{\partial b_i} \left(\sum_I (c_{0i}^{(k)} + \beta^{(k)} b_i) \delta_{pi} \right) = \beta^{(k)} \delta_{pi} \quad (5.13)$$

Thus, considering (5.11) and (5.13), (5.10) can be rewritten as:

$$\frac{\partial S_{rs}^{(k)}}{\partial b_i} = \sum_P P_{rsp}^{(k)} \beta^{(k)} \delta_{pi} \quad (5.14)$$

which, substituted in (5.9), yields:

$$\frac{\partial z'}{\partial b_i} = -\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \beta^{(k)} \delta_{pi} = -\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} \quad (5.15)$$

Considering the second term z'' , its derivative w.r.t. one of the common components of the link costs is:

$$\frac{\partial z''}{\partial b_i} = \frac{\partial}{\partial b_i} \left(\sum_I \frac{c_i^{(i)} - c_{0i}^{(i)}}{\beta^{(i)}} \int_0^{b_i^{-1}(x)} b_i^{-1}(x) dx \right) = b_i^{-1} \left(\frac{c_i^{(i)} - c_{0i}^{(i)}}{\beta^{(i)}} \right) \quad (5.16)$$

therefore the i th component of the gradient of $Z_{MUCSUE}(\mathbf{b})$ with respect to the vector of common costs on links, is:

$$\frac{\partial Z_{MUCSUE}}{\partial b_i} = -\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + b_i^{-1} \left(\frac{c_i^{(i)} - c_{0i}^{(i)}}{\beta^{(i)}} \right) \quad (5.17)$$

where the first term is the weighted summation of the auxiliary flows of class k on link i , *i.e.* the auxiliary standard flow on link i , and the second term is the current standard flow on link i .

The i th component of the gradient of the MUC SUE objective function in terms of standard link flows (5.5) can be obtained by considering (5.17) and applying the chain rule or, equivalently, by differentiating (5.5) directly (as reported in appendix A1) to obtain:

$$\frac{\partial Z_{MUCSUE}}{\partial v_i} = \left(-\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + v_i \right) \frac{db_i(v_i)}{dv_i} \quad (5.18)$$

(5.17) or (5.18) are zero only when the auxiliary standardised link flows are consistent with the current common part of the costs and standardised flows, that is for the MUC SUE conditions on the network. In fact, the consistency of the standardised link flows implies the consistency of the common part of the link costs and of the link costs by user class and, given a set of paths between each OD for

each class, the consistency of the path costs. The latter implies also the consistency of the pattern of choice and, therefore, of the path and link flows by user class.

5.2.3.2 Shape of the MUC SUE Objective Function and Uniqueness of Flows and Costs

Similarly to the SUE case, the MUC SUE objective function (5.5) expressed in terms of standard link flows is not convex everywhere, although it is so at equilibrium. In fact, the Hessian of such version of the objective function (derived in appendix A1) is given by:

$$\begin{aligned} \nabla^2 z_{MUCSUE}(\mathbf{v}) = & \nabla_{\mathbf{v}} \mathbf{b} + \mathbf{R} \nabla_{\mathbf{v}}^2 \mathbf{b} + \\ & + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs} \left(\nabla_{\mathbf{v}} b \Delta_{rs}^{(k)} \right) \left(-\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} b \Delta_{rs}^{(k)} \right)^T \end{aligned} \quad (5.19)$$

where the first term is the Jacobian of the common part of the link costs, which is diagonal and positive definite by the definition of the link costs. The second term is the product of the diagonal matrix \mathbf{R} of the differences between the current and auxiliary standardised flows by the Hessian of the common part of the link costs. As the matrix \mathbf{R} is made up of elements that can take any sign, the sign of this term is not generally known. The third term is a summation of quadratic forms similar to those appearing in the Hessian of the SUE objective function (see Sheffi, 1985). $\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)}$ is the Jacobian of the choice probabilities for class k , which is negative definite by definition (the strict definiteness is valid with choice models, like the MNP, that assign a positive probability to each option for finite values of the costs, otherwise it is only semidefinite (see Sheffi, 1985) but this does not alter the conclusion of the present discussion). Its opposite is therefore positive definite and the quadratic form that includes it is positive semidefinite (because of the definition of the link-path incidence matrix).

Since, in general, the term including \mathbf{R} can take any sign, so may $\nabla^2 z_{MUCSUE}(\mathbf{v})$. However, at equilibrium \mathbf{R} is $\mathbf{0}$ and $\nabla^2 z_{MUCSUE}(\mathbf{v})$ reduces to the sum of a positive definite matrix and of a number of positive semidefinite matrices, thus resulting

positive definite and establishing the local convexity of the function. This is similar to what is known about the SUE objective function (see Sheffi, 1985).

The objective function rewritten in terms of common link costs (5.4), however, is always convex. In appendix A2 it is shown that its Hessian is:

$$\nabla^2 z_{MUCSUE}(\mathbf{b}) = \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \Delta_{rs}^{(k)} \left(-\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \Delta_{rs}^{(k)T} + \nabla_{\mathbf{b}} \mathbf{b}^{-1} \quad (5.20)$$

where the first term is the summation, as above, of a number of positive semidefinite matrices as it includes quadratic forms of positive definite matrices. The second and last term, is the Jacobian of the inverse of the common part of the link costs and by definition it is diagonal and positive definite. Thus, this Hessian is the sum of a number of positive semidefinite matrices and of a positive definite matrix and it is always positive definite. The function (5.4) is then always convex and its minimum is a global minimum.

Because of the one to one correspondence between standard link flows and costs, implied by the strict convexity of the common part of the link costs, the uniqueness of the minimum of (5.4) implies also the uniqueness of the minimum of (5.5).

The (5.19) and (5.20) show that the solution is unique in terms of standard link flows and common part of the costs. Daganzo (1982) states that there is no uniqueness in terms of class flows and costs. However, as discussed above, a unique pattern of common link costs corresponds to a unique pattern of class link costs, as class link costs are simply the sum of a constant and of the common cost multiplied by a coefficient. If a set of paths is given, the additive class path costs are uniquely defined, as are the resulting choice probabilities by user class, when a stochastic choice model is given. In fact, if the choice probabilities were not unique it would mean that a stochastic choice model could give different results for the same set of utilities, which is not true. Since choice probabilities are unique and the OD flows by user class are fixed, the path flows, standard and by user class, are unique and so are the resulting link flows by user class.

Therefore, the MUC SUE conditions on a network correspond to unique link costs, for the common part and by user class, and standardised link flows, and, given a set

of paths, to unique path flows, standardised and by user class, and unique path costs and link flows by user class.

5.2.4 Algorithms for the Probit Path-Based MUC SUE Problem

5.2.4.1 Introduction

This section describes briefly a number of algorithms that can be used to solve the MUC SUE problem by minimising (5.5). The algorithms are extensions of those proposed and examined in the SUE case in chapter 4, adapted to include multiple user classes and are tested in the next section.

As in the SUE case the algorithms have been used to obtain the solution in terms of both path and link flows, considering a set of paths given and fixed from the outset of the assignment calculations. The algorithms used are therefore path-based and the formulation (5.5) of the problem, although link based, is employed because it is convenient to use and has a proven unique solution.

5.2.4.2 The MSA Algorithm for MUC-SUE

The MSA algorithm for MUC SUE is a simple extension of the one for a single user class reported in section 4.3.3.1.

The step is again the fixed MSA step $(l+n)^{-1}$ where n is the iteration number. The function optimised is different from the SUE one and the search direction is changed accordingly: in the MUC SUE case it is given by the vector linking the auxiliary standard link flows and the current ones. So it is a vector with a general term that can be written as:

$$\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} - v_i = w_i - v_i \quad (5.21)$$

The auxiliary standard link flows w_i , analogously to the SUE case, are obtained loading the network on the costs consistent with the current standardised flows. Given a set of paths between each OD, the search direction expressed in terms of paths corresponds uniquely to the search direction expressed in terms of link flows

as in (5.21). This allows us to work with algorithms accounting in parallel for the solution in terms of links and of path flows.

Similarly to the SUE case, the MSA search direction for MUC SUE is referred to here as the traditional search direction and is, numerically, the opposite of the gradient (5.17). However, as mentioned in section 4.7.2 for the SUE case, the traditional search direction cannot be associated with that gradient as it is used to solve the problem in the flow space and is simply a vector that defines always a descent direction for the problem. This can be easily verified by taking its dot product with the gradient of the objective function (5.5). This product will always be negative as the link performance functions are assumed to be strictly increasing.

The extension of the MSA to multiple user classes also requires the accommodation of the MUC data structure, more computer memory demanding than the SUE one as the path sets and covariance matrices may differ amongst user classes.

5.2.4.3 MUC SUE Algorithms with Traditional Search Direction and Optimised Line Search

Algorithms using the same search direction as the MSA but moving with steps obtained at each iteration from information on the local shape of the objective function (5.5) can be obtained similarly to the SUE case using the value of the objective function and its gradient.

All the interpolation methods suggested in chapter 4 can be applied to the MUC SUE problem (quadratic, cubic, rational, *regula falsi*, partial step quadratic) using the values of the gradient of the objective function along the search direction at the points indicated in the description of each method in chapter 4, and the values of the objective function at the two extremes of the search direction for the cubic interpolation method.

In this chapter, rather than comparing different step calculation methods, the assignments are performed using the quadratic interpolation method devised by Maher and Hughes (1997a) and described in section 4.3.3.2. This method has proved

to be robust and using only one method allows us to focus on comparing the merits of different search directions.

As in the SUE case, the quadratic interpolation method for the determination of the step corresponds to a linear interpolation of the gradient of the objective function between its two extremes. Thus it requires the values of the gradient along the search direction at those points, which can be obtained from loading data.

The calculation of the gradient along the search direction can be explained in more detail by writing the components of a point $v_i^{(\lambda)}$ along the search direction at iteration n as a function of the step λ taken along the search direction as:

$$v_i^{(\lambda)} = v_i^{(n)} + \lambda(w_i^{(n)} - v_i^{(n)}) \quad (5.22)$$

where $v_i^{(n)}$ is the current standard flow on link i at iteration n , $w_i^{(n)}$ is the standard auxiliary flow on link i at iteration n .

Then, exploiting the chain rule of derivation, the derivative of the MUC SUE objective function (5.5) along the search direction can be written as:

$$\frac{dz_{MUCSUE}}{d\lambda} = \sum_I \frac{\partial z_{MUCSUE}}{\partial v_i} \frac{dv_i}{d\lambda} \quad (5.23)$$

From (5.18) the first factor in the summation in (5.23) is:

$$\frac{\partial z_{MUCSUE}}{\partial v_i^{(\lambda)}} = (-w_i^{(\lambda)} + v_i^{(\lambda)}) \frac{db_i}{dv_i^{(\lambda)}} \quad (5.24)$$

whilst it is simple to see from (5.22) that the second factor is:

$$\frac{dv_i}{d\lambda} = w_i^{(n)} - v_i^{(n)} \quad (5.25)$$

thus (5.23) can be rewritten as:

$$\frac{dz_{MUCSUE}}{d\lambda} = \sum_I (-w_i^{(\lambda)} + v_i^{(\lambda)}) \frac{db_i}{dv_i^{(\lambda)}} (w_i^{(n)} - v_i^{(n)}) \quad (5.26)$$

The gradient at the current point can be obtained by substituting the current and auxiliary solution in standardised flows for $\mathbf{v}^{(\lambda)}$ and $\mathbf{w}^{(\lambda)}$ whilst the gradient at the extreme of the search direction is obtained by substituting the auxiliary flows for $\mathbf{v}^{(\lambda)}$ and the auxiliary flows resulting from a further stochastic loading on the costs consistent with $\mathbf{v}^{(\lambda)}$, for $\mathbf{w}^{(\lambda)}$.

The actual form of $\frac{db_i}{dv_i}$ depends on the sort of function used to describe the standard cost of travelling along a link. For instance, in the case of power or BPR performance functions (US Bureau of Public Roads, 1964) the b_i can be written as:

$$b_i = \omega_i \left(\frac{v_i}{c_i} \right)^{\gamma_i} \quad (5.27)$$

and therefore results in:

$$\frac{db_i}{dv_i} = \frac{\omega_i \gamma_i}{(c_i)^{\gamma_i}} (v_i)^{\gamma_i-1} \quad (5.28)$$

The quadratic interpolation can be carried out once or refined in subintervals containing the zero of the gradient until an improved gradient is found or to a given precision. The latter method is referred to, in the following sections, as precise or refined whilst the other methods are referred to as unrefined.

As in the SUE case, the quadratic interpolation method has been included as the second part of hybrid algorithms started with a number of MSA iterations.

5.2.4.4 MUC SUE Algorithms with Alternative Search Directions

The MUC SUE search direction used in the algorithms described above can be interpreted in conjunction with the MUC SUE objective function (5.5) as the transformation into the space of the flows of a steepest descent search direction defined in a space related to the flow space by a non-linear change of coordinates given by the square root of the Jacobian of the common part of the link costs. In fact, the discussion on the interpretation of the traditional MUC SUE search direction

parallels the one given in 4.3.5.2 for the traditional SUE search direction with the difference that the preconditioning matrix \mathbf{J}_n , in the MUC SUE case is the Jacobian of the common part of the link costs and the standardised flows should be substituted for the flows appearing the SUE case, as can be seen by comparing the respective expressions of the gradient of the objective function.

Continuing the parallel with the interpretation of the traditional search direction presented in chapter 4, it is possible to write algorithms working along preconditioned conjugate gradient directions. The final formulae can be written in the space of the standardised flows and are identical to (4.55), (4.58), (4.59) and (4.60) except for the fact that the link flows should now be intended as standardised link flows, the preconditioner \mathbf{J}_n is the Jacobian of the common part of the link costs and the gradient of the objective function refers now to the MUC SUE function (5.5).

As in the SUE case, although the coefficient combining the previous search direction with the current traditional search direction must be calculated using quantities referred to the links, the search directions can be expressed either in terms of link flows or path flows (for a predefined and fixed link-path incidence matrix), thus the algorithm can produce updates of the solution both in terms of path and link flows. Moreover, similar algorithms can be used for pure link-based models.

The tests reported in the following sections have been carried out with the preconditioned Polak-Ribiere and the preconditioned Fletcher-Reeves search directions. From the SUE tests the latter is expected to perform less well. In fact, the MUC SUE problem is solved simply as a SUE problem in the standardised flows and a similar behaviour of the algorithms should be expected (allowing for the differences in the corresponding problems due to the different congestion induced by the flows).

The extremes of the search direction described for the MSA define always a vector in the space of the feasible flows. If the preconditioned gradient search vector extends into the semispace of the negative flows it is shortened so that no flow is less than zero.

Also with the preconditioned conjugate search directions the tests are carried out only using the quadratic interpolation method described in the previous section. Formulae (5.22)-(5.26) can be adapted to the present case by simply recalling that the extreme of the current search direction is now given by $u_i^{(n)}$, the point returned by the preconditioned conjugate gradient algorithm, rather than by the auxiliary solution, thus (5.22) should be modified as:

$$v_i^{(\lambda)} = v_i^{(n)} + \lambda(u_i^{(n)} - v_i^{(n)}) \quad (5.29)$$

and it is immediately seen that (5.26) results in:

$$\frac{\partial z_{MUCSUE}}{\partial \lambda} = \sum_I \left(-w_i^{(\lambda)} + v_i^{(\lambda)} \right) \frac{db_i}{dv_i^{(\lambda)}} (u_i^{(n)} - v_i^{(n)}) \quad (5.30)$$

Once more, the gradient at the current point can be obtained by substituting the current and auxiliary solutions in standardised flows for $v^{(\lambda)}$ and $w^{(\lambda)}$ whilst the gradient at the extreme of the search direction is obtained by substituting $u^{(n)}$ for $v^{(\lambda)}$ in the first bracket and the auxiliary flows resulting from a further stochastic loading on the costs consistent with $u^{(n)}$, for $w^{(\lambda)}$.

The interpolation is carried out once, or until an improved gradient is found or to a given precision. The algorithms not implementing precise line search are used to try and save on computational effort although the approximate line search may introduce a further element of approximation (beside the changing preconditioner and the fact that the function is not exactly quadratic) to the conjugacy of the search direction. In the sections reporting the results, the preconditioned Fletcher-Reeves algorithms are marked ppFR when they implement a line search carried out to a given precision and apFR when they implement a line search carried out once or until an improved gradient is found. The corresponding preconditioned Polak-Ribiere algorithms are coded respectively ppPR and apPR.

As a safety device, the dot product of the gradient of the objective function and of the new direction in terms of link flows is calculated at each iteration and, if it is not negative, the preconditioned conjugate gradient direction is discarded and restarted.

As in the SUE case, the preconditioned conjugate gradient search directions have been included as the second part of hybrid algorithms started with a number of MSA iterations.

5.2.5 Performance of the Algorithms for the MUC SUE Problem

5.2.5.1 Introduction

This section presents the results of the numerical tests carried out by solving for MUC SUE with the algorithms described above. The results are described separately for each of the test networks used after introducing the test methodology and the test bed. The section closes with a summary of the test results.

5.2.5.2 MUC SUE Algorithm Test Methodology

The algorithms examined have been tested for efficiency with a methodology similar to the one used for SUE and discussed in chapter 4.

Solving for MUC SUE on test networks the number of loadings and iterations and the computing time necessary to reach a certain level of convergence of the solution have been recorded.

The stochastic network loading (here intended as the loading of all the user classes being assigned) has been taken as the elemental operation performed by the traffic assignment algorithms. The reason for choosing the number of loadings is the same as explained in chapter 4 for the SUE case: iterations cannot be used as a cost unit as an iteration with different algorithms requires different operations and number of loadings and, thus, a different computational effort. The number of loadings seems to reflect the largest part of the computational time involved in the computation also when a search direction different from that used in the MSA, described above, is employed. Moreover, it allows us to avoid referring directly to the computational time that is machine specific.

The level of convergence reached by an algorithm is measured as a function of the distance between the current and the auxiliary solution at an iteration since, as the equilibrium is approached, the current and auxiliary solutions get close to each other until they coincide, as has been remarked above on the discussion on the gradient of the objective function, written in either possible ways. The plots in the following sections display the natural logarithm of a non-dimensional measure of the distance between the current and the auxiliary solution, which can be written as:

$$\ln RMSnd = \ln \sqrt{\frac{1}{P} \sum_p \frac{(x_p - y_p)^2}{(0.5(x_p + y_p))^2}} \quad (5.31)$$

where the current and the auxiliary solution, respectively x_p and y_p , are in terms of standardised path flows and P is the number of paths included in the statistic (those for which either the current or the auxiliary solution is at least 0.1% of the standardised flow between the relevant OD).

As the equilibrium is approached the current and the auxiliary flows tend to coincide and the statistic (5.31) tends to $-\infty$.

The statistic (5.31) is considered here to compare the algorithms because it provides an aggregate measure of convergence for all user classes that is convenient to use for comparing different algorithms and because the corresponding measures by user class, that can be used as an alternative, have a similar trend as shown in the example in fig. 5.1. In practical applications the statistic for the user class that converges last should probably be considered before closing the calculations.

The smaller the number of loadings required by an algorithm to reach a pre-set threshold of the statistic (5.31), the more efficient is the algorithm. In the test results reported here the solution is considered reached for a target level of the RMSnd statistics of 10^{-4} , corresponding to a value of $\ln RMSnd$ of -9.21 . This is the same criterion to compare the algorithms used in the SUE case. As when comparing SUE algorithms in chapter 4, the plots employed, similar to fig. 5.1, allow us also to appreciate the relative performance of the algorithms during the calculation, rather than simply the final computational effort.

The tests have been carried out using the same two approximation methods used in chapter 4, the Mendell-Elston approximation (Mendell-Elston, 1974) with the optimised calculation order of Kamakura (1989) and the improved Clark approximation (Clark, 1961), that have been described in chapter 3.

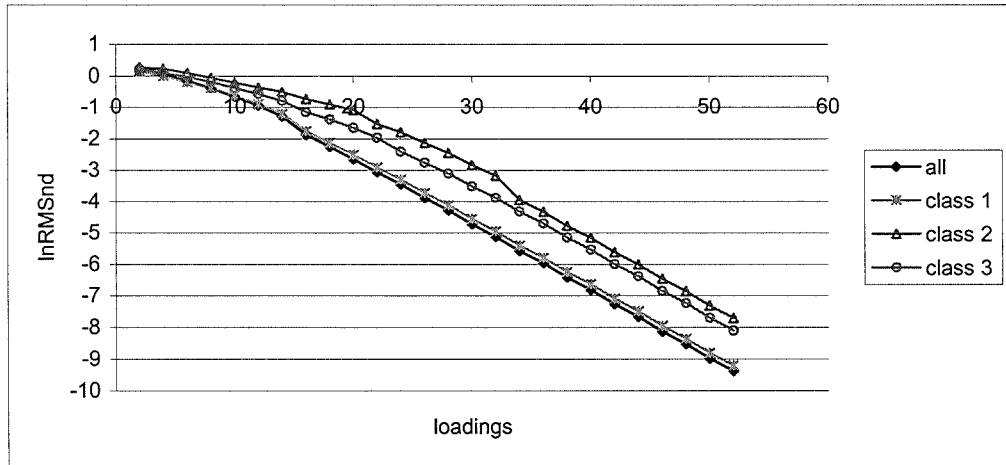


Fig. 5.1 – Example of convergence trend obtained plotting the statistic (5.31) and similar statistics disaggregate by user class against the number of MUC stochastic network loadings for a case with 3 user classes.

5.2.5.3 MUC SUE Algorithm Test Bed

The algorithms for MUC SUE have been tested on modifications (to accommodate 2 and 3 user classes) of the networks of Chen and Alfa (1991a), on the Sioux Falls networks (used *e.g.* in LeBlanc, 1975; Vythoukcas, 1990) and on the Headingley network (used *e.g.* in Maher *et al.*, 1999) that have been used already in chapter 4. Their main characteristics are summarised in section 4.3.2.2.

On each network the tests have been carried out considering two and three user classes. The OD matrix for each user class has been obtained as a portion of the original OD matrix used for the experiments in chapter 4. Tables 5.1, 5.2 and 5.3 report the portion P of the OD flows considered in the SUE case assigned to each user class for each network.

The same sets of paths used in chapter 4 and discussed in section 4.3.2.2 have been used for all user classes. This assumption has been made for simplicity but the

algorithms can be used also when different sets of paths are specified for each user class.

For each user class and on each network, different free flow costs, coefficients α and β (appearing in eq. (5.1) and (5.2.)) and, in some cases, ratio of the variance to the free flow cost of each link have been considered.

The free flow costs of the first user class on each network are the original ones whilst those for the remaining classes have been obtained by multiplying the cost for each link by a factor drawn at random from a uniform distribution [0.50;1.00] or [0.75;1.25].

The coefficients α and β and ratio of the variance to the free flow cost used in each test are detailed in the following tables 5.1, 5.2, 5.3. They have not been chosen to replicate a particular situation but simply to test the algorithms.

C+A	UC1	UC2
P	0.85	0.15
α	1.00	1.50
β	0.80	1.20
$\sigma^2/\text{ff cost}$	0.50	0.40

C+A	UC1	UC2	UC3
P	0.80	0.10	0.10
α	1.00	1.00	2.00
β	0.80	1.20	1.00
$\sigma^2/\text{ff cost}$	0.50	0.40	0.50

Table 5.1 – Proportion P, coefficients α and β and ratio of the variance to the free flow costs of the links used in the cases of 2 and 3 user classes (UC) on the Chen and Alpha network.

Sioux F	UC1	UC2
P	0.25	0.75
α	1.20	0.90
β	0.70	1.25
$\sigma^2/\text{ff cost}$	0.50	0.40

Sioux F	UC1	UC2	UC3
P	0.75	0.10	0.15
α	0.90	1.50	0.90
β	0.70	1.20	0.70
$\sigma^2/\text{ff cost}$	0.50	0.50	0.10

Table 5.2 – Proportion P, coefficients α and β and ratio of the variance to the free flow costs of the links used in the cases of 2 and 3 user classes (UC) on the Sioux Falls network.

Hdngly	UC1	UC2
P	0.75	0.25
α	1.00	0.80
β	0.80	1.20
$\sigma^2/\text{ff cost}$	0.70	0.70

Hdngly	UC1	UC2	UC3
P	0.60	0.30	0.10
α	1.00	1.00	1.00
β	1.00	1.10	0.80
$\sigma^2/\text{ff cost}$	0.50	0.40	0.50

Table 5.3 – Proportion P , coefficients α and β and ratio of the variance to the free flow costs of the links used in the cases of 2 and 3 user classes (UC) on the Headingley network.

It should be recalled again that the Headingley network contains connectors, which are links with fixed costs leading in and out of centroids, which are not accounted for in the formulation of the problem being used. Thus the application of the algorithms proposed to the Headingley network is heuristic.

5.2.5.4 Test Results for the Chen and Alpha Network

Before discussing the results it is important to note that the selection of figures reported for this and for the other test networks refer alternatively to the Mendell-Elston or to the Clark approximation because typically no substantial differences were found using the same algorithm on the same test case with either approximation. For the same reason the comments reported refer to the algorithms applied with either approximation.

The tests on the Chen and Alpha network show that the MSA is outperformed by the algorithm using the traditional search direction and optimising the step by quadratic interpolation, as in fig. 5.2 and 5.3, respectively for cases with 2 and 3 user classes. However, as seen already in the SUE case, the MSA outperforms the optimised step algorithm at the beginning of the optimisation.

The convergence trends for the algorithms with step calculated with one interpolation and those returning it after checking that an improved gradient has been found, coincide and, as in the SUE case, these algorithms outperform those with the step refined to a given precision.

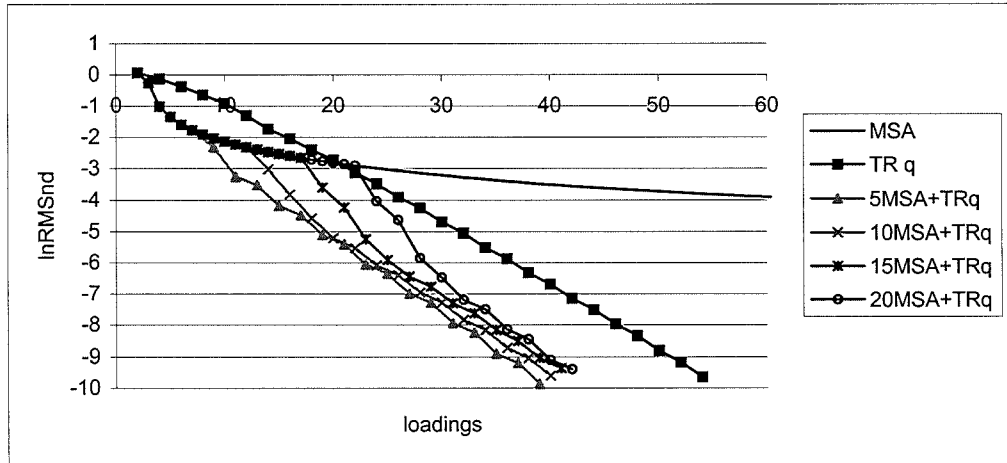


Fig. 5.2 – Comparison of the MSA, the algorithm using the traditional search direction (TR) and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Mendell-Elston approximation, 2 user classes.

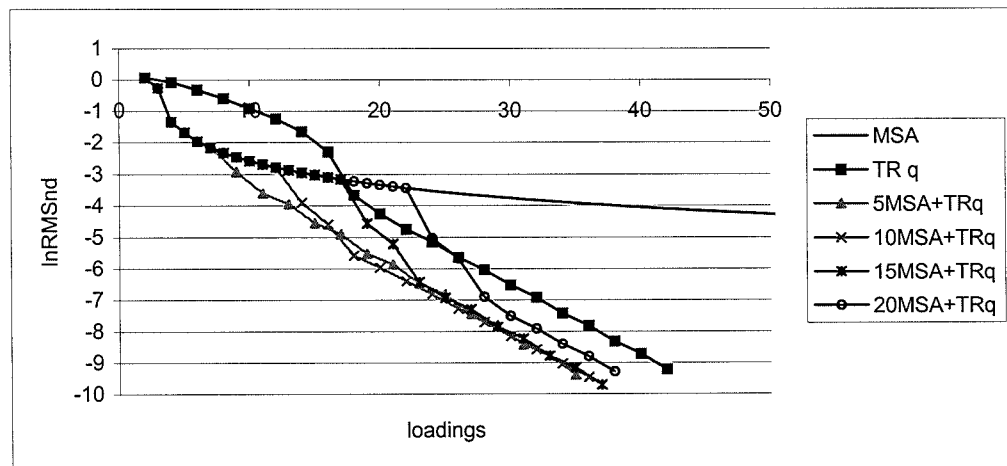


Fig. 5.3 – Comparison of the MSA, the algorithm using the traditional search direction (TR) and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Clark approximation, 3 user classes.

The initial good performance of the MSA suggests the use, again as in the SUE case, of hybrid algorithms starting with a number of MSA iterations before calculating an optimised step along the traditional search direction. The tests with hybrid algorithms started with 5, 10, 15 and 20 MSA iterations gave results as those reported again in fig. 5.2 and 5.3: the hybrid algorithm outperforms the MSA and the optimised step algorithm and there is rather little to choose between the different number of starting MSA when the final performance is considered.

Using the preconditioned Fletcher-Reeves search direction with precise determination of the step (ppFR) improves on the corresponding algorithms employing the traditional search direction both in terms of iterations and in terms of loadings (see the example in fig. 5.4). If the step is only refined to 10^{-2} the ppFR performs as well (but not from the start) as the algorithms using unrefined line search along the traditional search direction. In the other cases they require more loadings than the algorithms without refined line search.

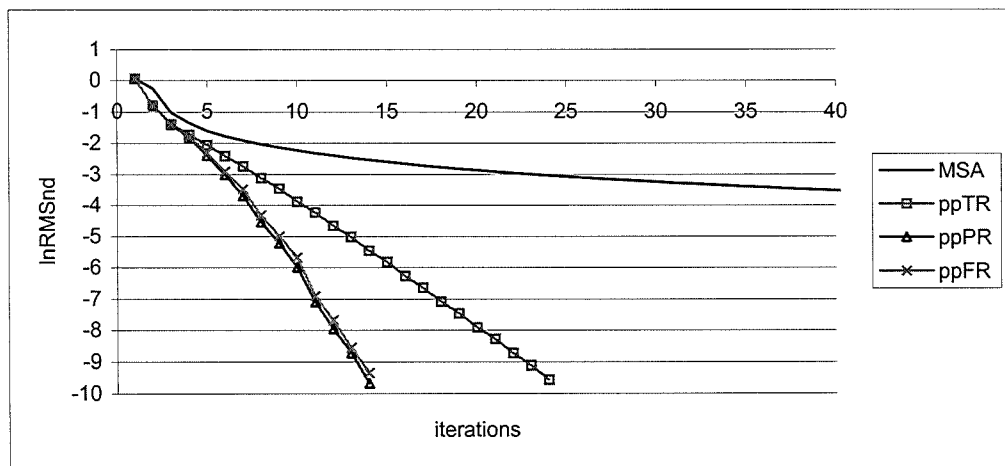


Fig. 5.4 – Comparison of the MSA and algorithms refining the step to a precision of 10^{-3} along the traditional search direction (ppTR), the preconditioned Polak-Ribiere search direction (ppPR) and the preconditioned Fletcher-Reeves search direction (ppFR). Chen and Alpha network, Mendell-Elston approximation, 2 user classes.

The algorithms using the approximate preconditioned Fletcher-Reeves (apFR) search direction along with a quadratic interpolation to determine the step carried out once or until an improved gradient is found, coincide in all examples and with both approximations employed.

The algorithm using the unrefined optimised step from the start shows oscillations of the convergence trend similar to those reported for the SUE case, that delay the convergence (an example of this is reported in fig. 5.5). This unsatisfactory behaviour is obviated if the apFR is implemented as part of hybrid algorithms started with a number of MSA iterations. Such hybrid algorithms with the apFR search vector show no such oscillations and improve on the convergence trend given by the

corresponding hybrid algorithms using the traditional search direction as shown in fig. 5.6 and 5.7. They also do better than ppFR algorithms.

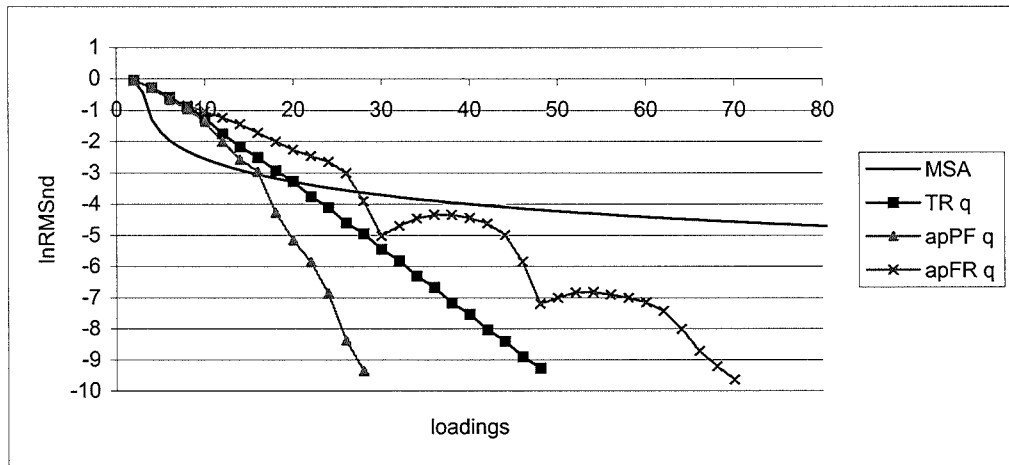


Fig. 5.5 – Comparison of the MSA, the algorithm using the traditional search direction with step obtained to give an improved gradient (TRq) and corresponding algorithms using the preconditioned Fletcher-Reeves (apFR) and the preconditioned Polak-Ribiere (apPR) search direction. Chen and Alpha network, Clark approximation, 2 user classes.

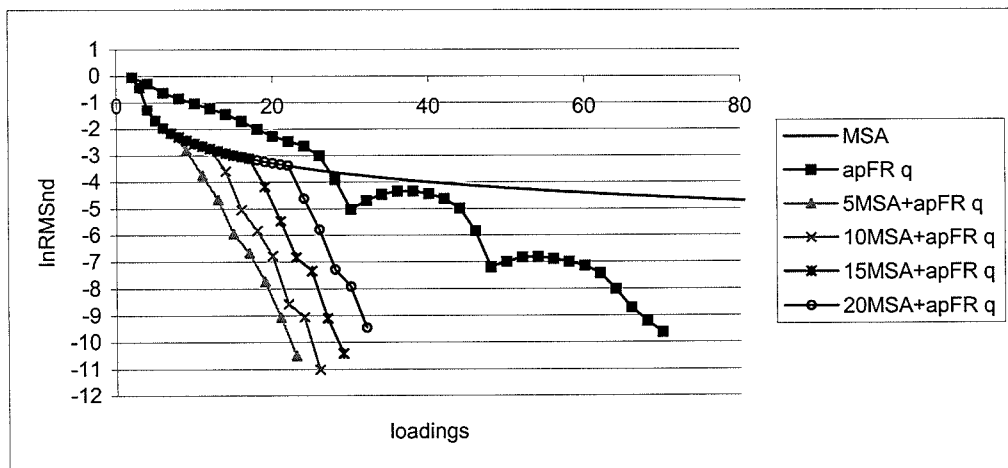


Fig. 5.6 – Comparison of the MSA, the algorithm using preconditioned Fletcher-Reeves (apFR) search direction and hybrid apFR algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Clark approximation, 2 user classes.

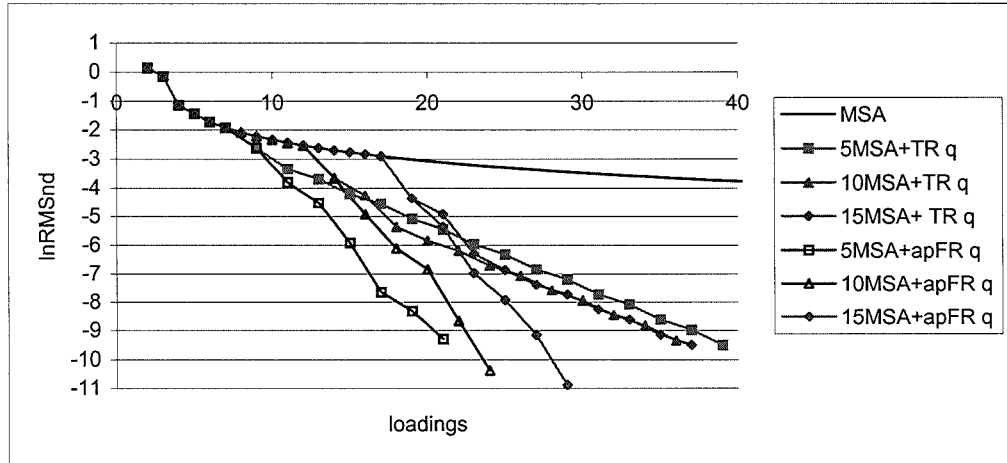


Fig. 5.7 – Comparison of the MSA, and hybrid algorithms using the traditional (TRq) and the approximate preconditioned Fletcher-Reeves (apFR) search direction started with the number of MSA iterations indicated. Chen and Alpha network, Mendell-Elston approximation, 3 user classes.

Using the Polak-Ribiere search direction with precise determination of the step gives trends similar to those of the ppFR both when the costs are considered in terms of iterations and in terms of loadings. An example of the comparison between the two precise preconditioned conjugate gradient directions and the traditional search direction in terms of iterations is given in fig. 5.4, which shows, as found in the SUE case, that preconditioned conjugate gradient directions are more effective than the traditional one in exploring the solution space.

The algorithm using the apPR search direction with step obtained by checking that a point corresponding to an improved gradient has been found does not show the oscillations of the apFR and improves noticeably soon after the start on the corresponding algorithm using the traditional search direction, as can be seen from the examples reported in figs. 5.8 and 5.9. Moreover, the apPR algorithm also does better than the corresponding algorithms using precise line search.

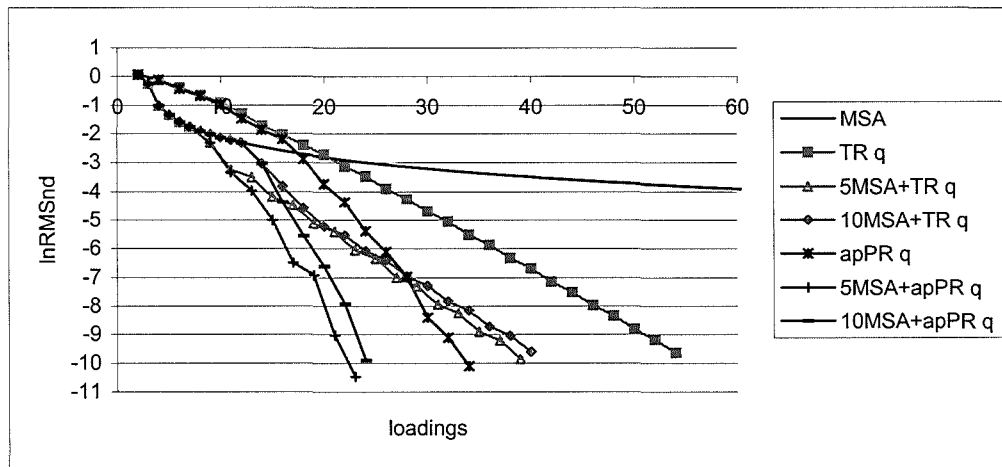


Fig. 5.8 – Comparison of the MSA, the algorithm using the traditional search direction (TR q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Mendell-Elston approximation, 2 user classes.

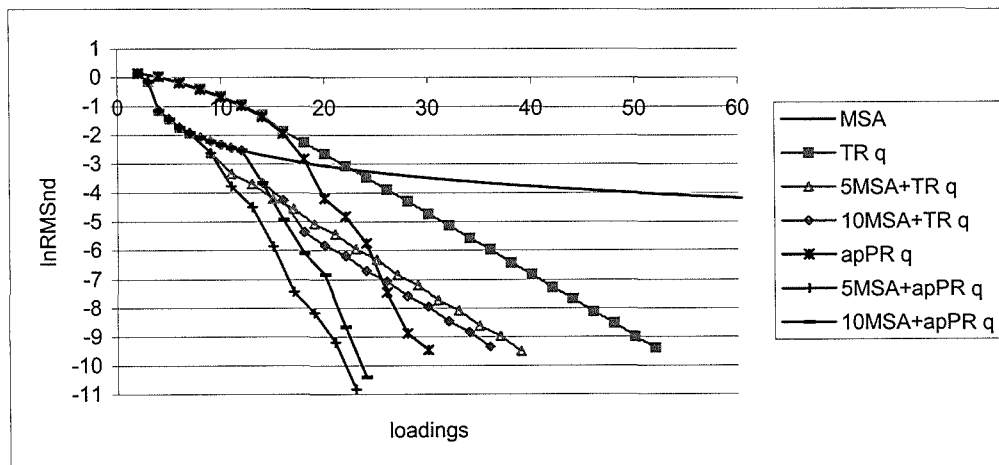


Fig. 5.9 – Comparison of the MSA, the algorithm using the traditional search direction (TR q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Clark approximation, 3 user classes.

The MSA, at the beginning of the calculations, outperforms also the apPR. Hybrid algorithms with the apPR (tested, as before, starting with 5, 10, 15 and 20 MSA iterations) improve on the corresponding hybrid algorithms employing the traditional search direction but there is no practical difference between the trend of the hybrid apFR algorithms and those of the hybrid apPR.

With this search direction not all the numbers of starting MSA iterations improve on the apFR used from the outset: 5 or 10 iteration, however, ensure a higher efficiency, as shown in figs. 5.8 and 5.9.

5.2.5.5 Test Results for the Sioux Falls Network

On the Sioux Falls network the convergence trend for steps along the traditional search direction obtained with one interpolation or checking that an improved gradient is obtained coincide in all cases.

Using the traditional search direction refining the step to a given precision results in convergence trends less efficient than when the step is not refined, as shown in the example in fig. 5.10. The MSA does, on the whole, worse than using an optimised step but, as seen in many other cases, performs well at the beginning (see again the example in fig. 5.10). Using hybrid algorithms, started with 5, 10, 15, 20 MSA iterations before turning to seeking the step by interpolation, results in some cases in noticeable efficiency improvements as in fig. 5.11, although in other cases the improvement is minor or there is little to choose between the original algorithm and the hybrid ones, as in the example in fig. 5.12. In that case, starting with 5 MSA iterations improves on using the optimised step at the beginning of the calculations but does not as well on the whole.

Using the preconditioned Fletcher-Reeves search direction with precise line search (algorithm ppFR) improves on the corresponding algorithms using the traditional search direction both in terms of loadings and iterations, showing once more the effectiveness of such search direction. Moreover, using precise line search does not bring about the instabilities that are recorded in several cases when the Fletcher-Reeves search direction is used with step determined by simple interpolation or making sure that an improved gradient has been found (algorithm apFR).

This oscillating convergence trend is not present in any case when the apFR is used in hybrid algorithms after a number of MSA iterations, as shown in fig. 5.13. All the

hybrid algorithms, besides preventing the oscillations, give a better performance than the algorithm with line search from the start. Fig. 5.14 depicts a typical result obtained with apFR hybrid algorithms: they improve in all cases on the corresponding algorithms using the traditional search direction.

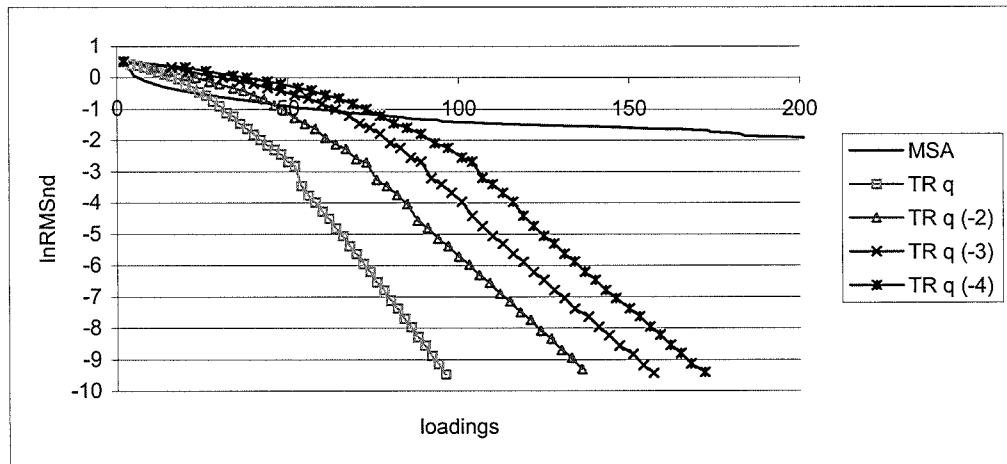


Fig. 5.10 – Comparison of the MSA and algorithms refining the step along the traditional search direction (TRq). The number in parenthesis gives is power of ten of the precision required. Sioux Falls network, path set S1, Mendell-Elston approximation, 3 user classes.

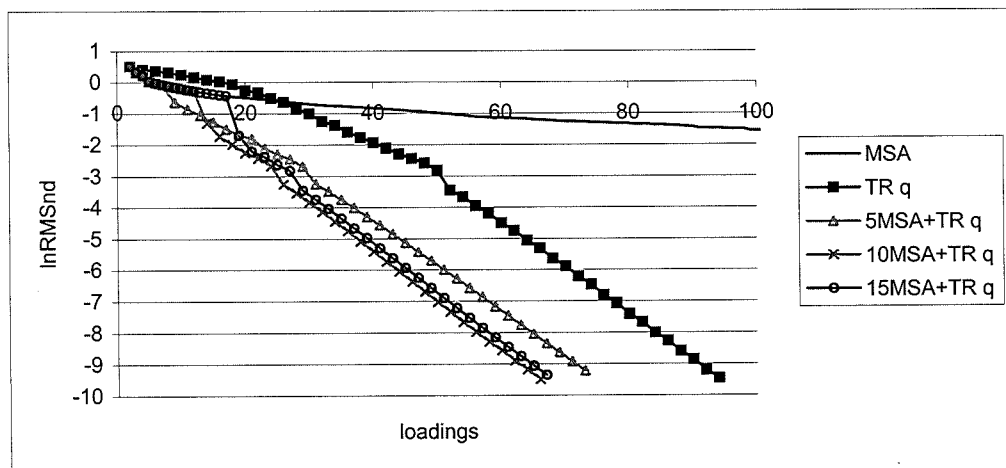


Fig. 5.11 – Comparison of the MSA, the algorithm using the traditional search direction (TR) and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network, path set S1, Clark approximation, 3 user classes.

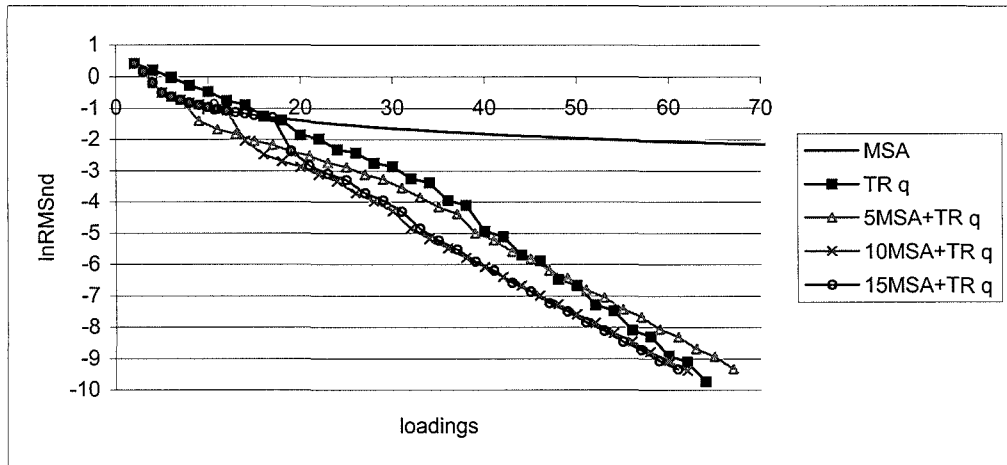


Fig. 5.12 – Comparison of the MSA, the algorithm using the traditional search direction (TR) and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network, path set S2, Mendell-Elston approximation, 3 user classes.

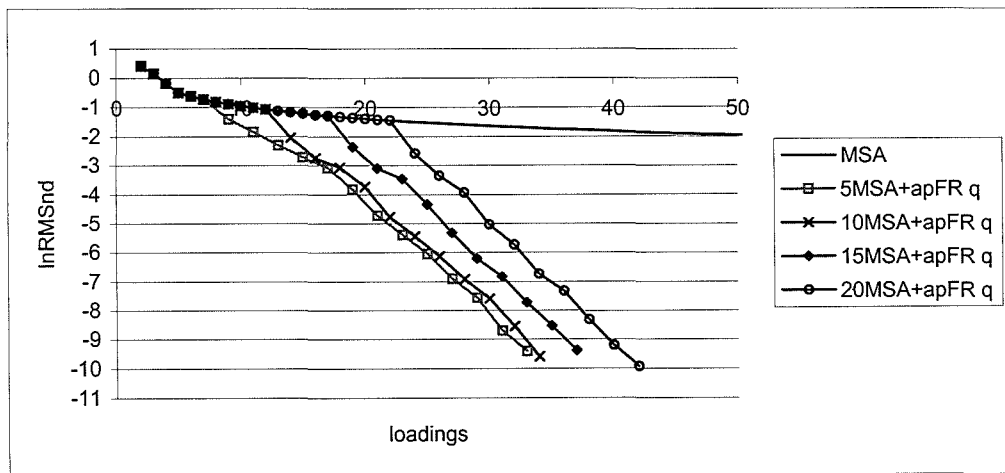


Fig. 5.13 – Comparison of the MSA and hybrid algorithms using the approximate Fletcher-Reeves search direction (apFR) started with the number of MSA iterations indicated. Sioux Falls network, path set S2, Mendell-Elston approximation, 3 user classes.

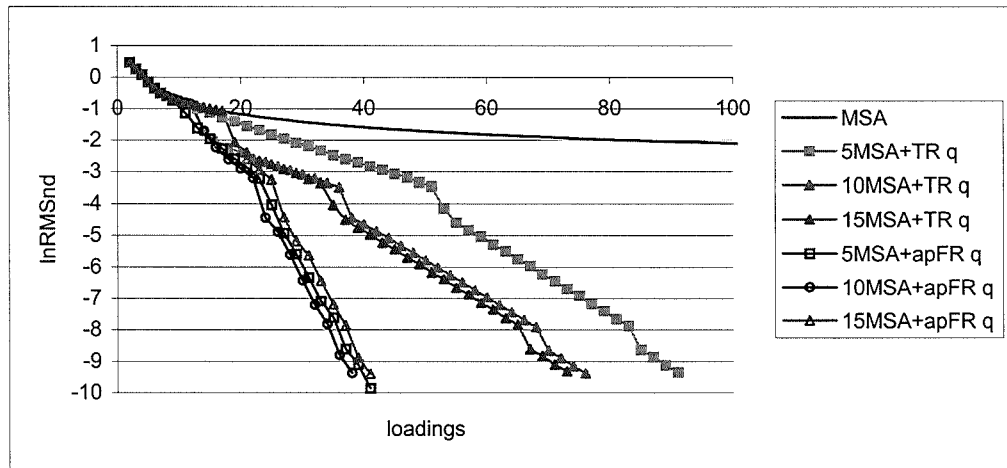


Fig. 5.14 – Comparison of the MSA, and hybrid algorithms using the traditional (TRq) and the approximate preconditioned Fletcher-Reeves (apFR) search direction started with the number of MSA iterations indicated. Sioux Falls network, path set S2, Clark approximation, 3 user classes.

The preconditioned Polak-Ribiere search direction, when coupled with a line search refined to a given precision (algorithm ppPR), gives a convergence trend that is similar to that of the preconditioned Fletcher-Reeves search direction (on which it slightly improves in some cases) but improves noticeably on the convergence behaviour of the corresponding algorithms using the traditional search direction. This can be noticed both when analysing the convergence trends in terms of loadings and in terms of iterations.

Using the preconditioned Polak-Ribiere search direction with optimised steps obtained making sure that an improved gradient has been obtained (algorithm apPR) is more efficient than refining the step and is more efficient than using the corresponding algorithm working along the traditional search vector or the preconditioned Fletcher-Reeves search direction (when the latter does not give the previously mentioned oscillating trends), though the improvement on the traditional one is more noticeable, as shown in fig. 5.15. Moreover the apPR algorithm never shows instabilities.

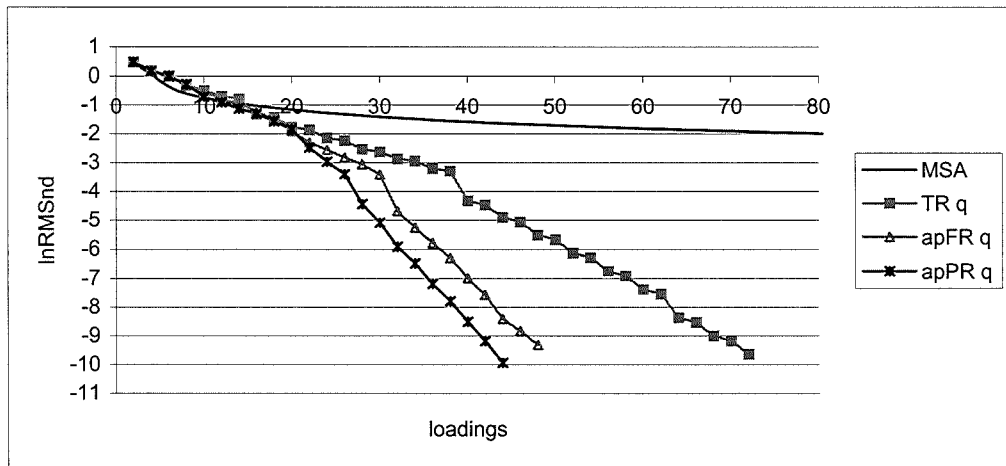


Fig. 5.15 – Comparison of the MSA and algorithms refining the step to obtain an improved gradient value along the traditional search direction (TRq), the preconditioned Fletcher-Reeves search direction (apFR q) and the preconditioned Polak-Ribiere search direction (apPR q). Sioux Falls network, path set S2, Clark approximation, 2 user classes.

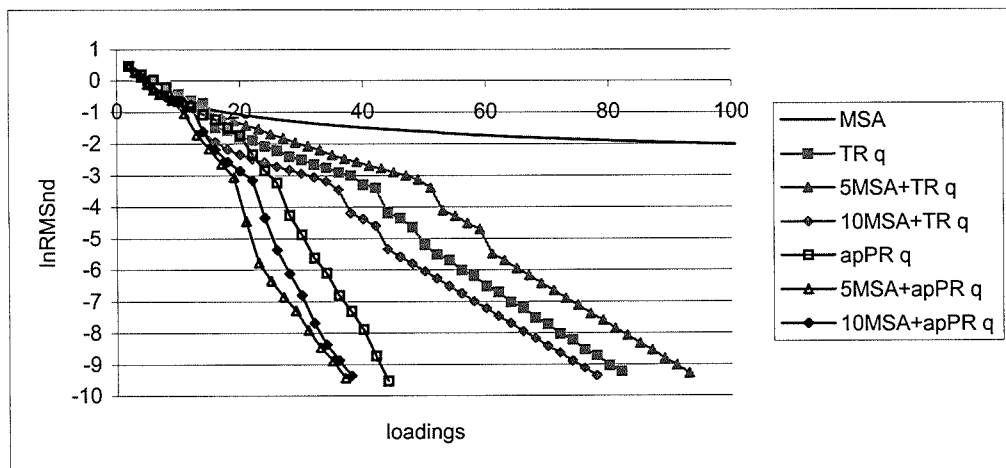


Fig. 5.16 – Comparison of the MSA, algorithms using the traditional (TRq) and the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network, path set S2, Mendell-Elston approximation, 2 user classes.

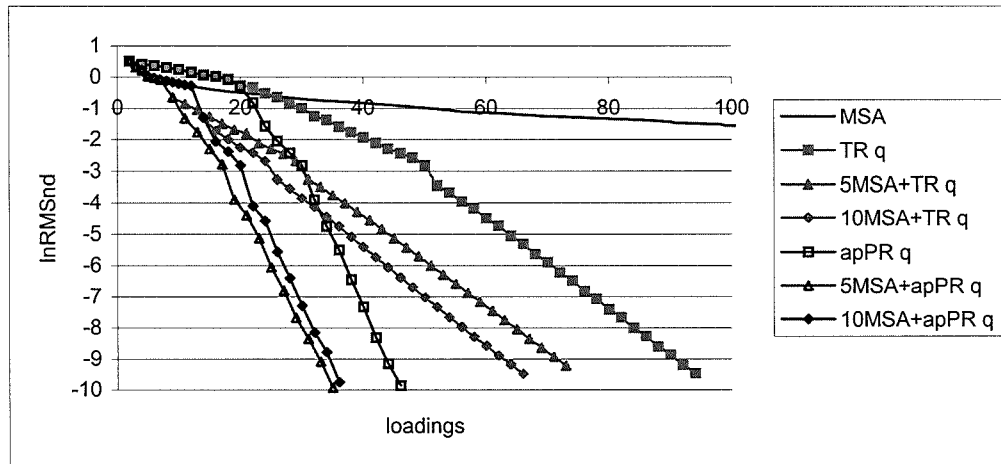


Fig. 5.17 – Comparison of the MSA, algorithms using the traditional (TRq) and the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network, path set S1, Mendell-Elston approximation, 3 user classes.

The results obtained by starting the apPR algorithm after 5, 10, 15 and 20 MSA iterations showed that, in most cases, this device improves on the performance of the apPR used from the start. Moreover such algorithms are always more efficient than the corresponding ones using the traditional search direction, and for low number of starting MSA iterations, also slightly more efficient than the hybrid apFR ones (though the difference disappears when more than 10 starting MSA iterations are used). This can be seen observing again the examples in fig. 5.16 and 5.17.

5.2.5.6 Test Results for the Headingley Network

On the Headingley network the algorithms using the MUC SUE version of the traditional search direction and returning the step after a single interpolation or checking that an improved gradient has been found do not give identical convergence trends in all cases and in some cases with 3 user classes the algorithms not performing the improvement check show instabilities that can be explained by the higher congestion forced by the presence of the three user classes. In fact these instabilities are absent when the same network and user classes are considered with elastic demand. However, it is important to note that algorithms checking for an improved gradient as well as those refining the step to a given precision are robust also in this case.

As shown in figs. 5.18 and 5.19, also in this case the MSA does well at the beginning of the calculations before being outperformed by the methods using the traditional search direction and an optimised step. Using hybrid algorithms improves on the convergence performance of the optimised step algorithm used from the start, although in some cases more than 10 starting MSA iterations are necessary to bring about an improvement, as shown again in figs. 5.18 and 5.19. The use of hybrid algorithms also eliminates the instabilities recorded for algorithms that do not check for gradient improvement.

Using steps refined to a given precision results, in some cases, in algorithms that are less efficient than those checking only that an improved gradient has been found, but in other cases refined step algorithms perform as well or slightly better. However they do not perform better than hybrid algorithms.

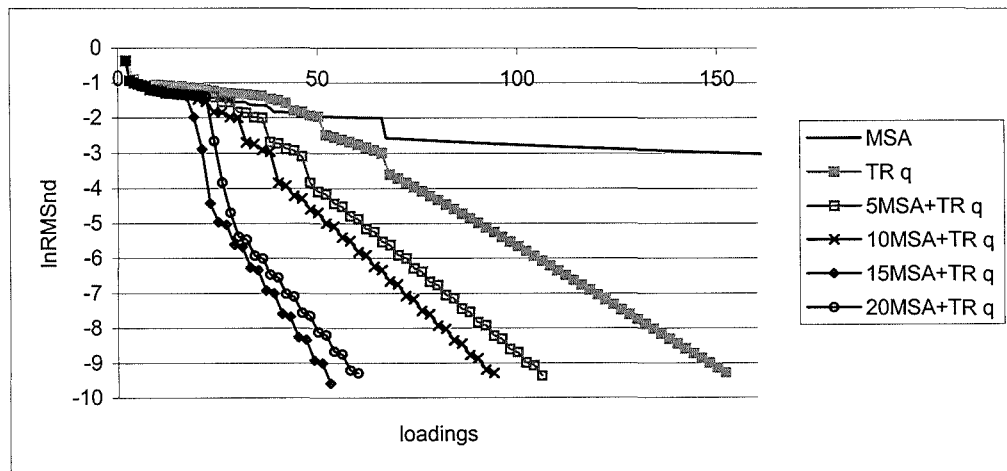


Fig. 5.18 – Comparison of the MSA, the algorithm using the traditional (TRq) search direction with step returned after finding an improved gradient and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, path set H1, Mendell-Elston approximation, 3 user classes.

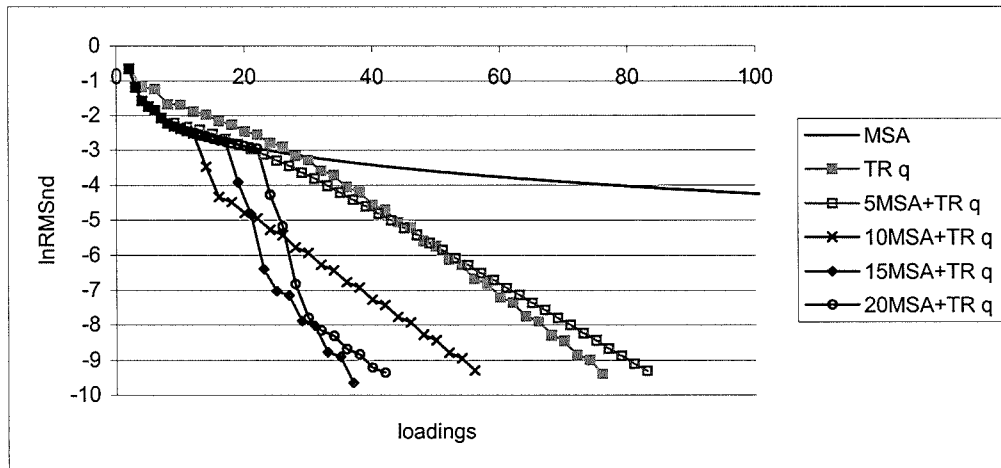


Fig. 5.19 – Comparison of the MSA, the algorithm using the traditional (TRq) search direction with step returned after finding an improved gradient and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, path set H2, Clark approximation, 2 user classes.

Algorithms working along the preconditioned Fletcher-Reeves search direction and refining the step to a given precision (algorithms ppFR) do better than the corresponding algorithms working along the traditional search direction as depicted in the example in fig. 5.20.

Moreover, using refined steps does not result in instabilities of the convergence trend which are instead recorded in a few cases and especially when the gradient improvement is not checked. The device of using hybrid algorithms to eliminate the instabilities works in all cases although, for some, at least ten MSA iterations to start are necessary. This proves again the lack of robustness of the algorithms which do not check for gradient improvement and also suggest that the preconditioned Fletcher-Reeves search direction applied approximately to the solution of the MUC SUE problem may be prone to instabilities, just as it was in the SUE case.

Using the apFR algorithm after a number of MSA iterations gives convergence trends that improve on those of the corresponding algorithms using the traditional search direction as shown in fig. 5.21.

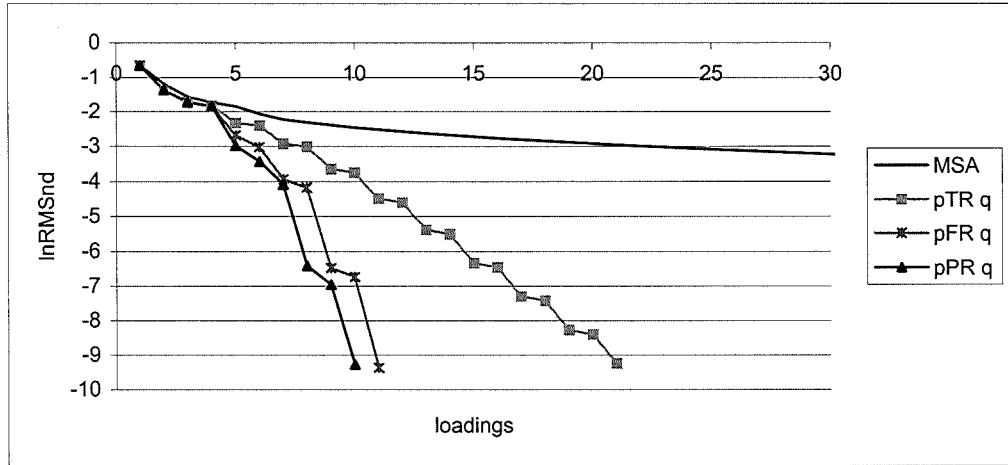


Fig. 5.20 – Comparison of the MSA and algorithms refining the step to a precision of 10^{-3} along the traditional search direction (pTR), the preconditioned Fletcher-Reeves search direction (pFR) and the precondition Polak-Ribiere search direction (pPR). Headingley network, path set H2, Clark approximation, 2 user classes.

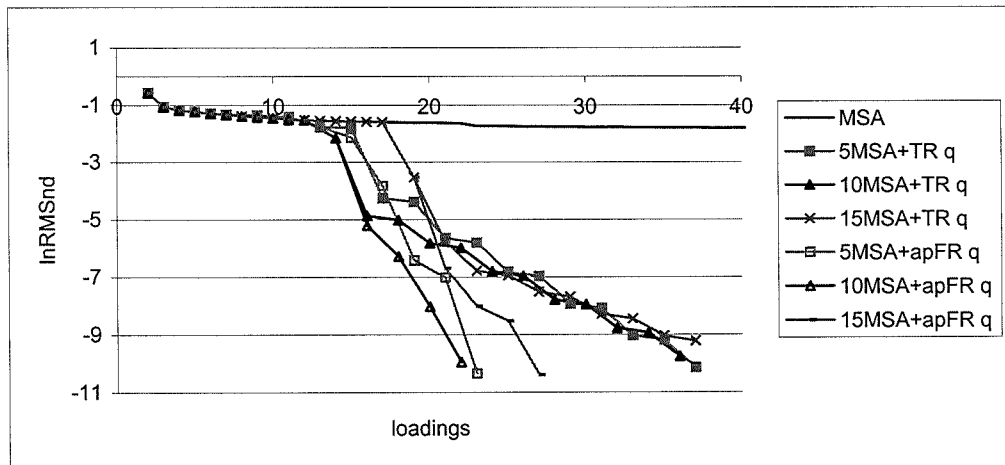


Fig. 5.21 – Comparison of the MSA, and hybrid algorithms using the traditional (TRq) and the approximate preconditioned Fletcher-Reeves (apFR) search direction started with the number of MSA iterations indicated. Headingley network, path set H1, Clark approximation, 2 user classes.

There is little to choose between the convergence trend of the preconditioned Polak-Ribiere algorithm with step refined to a given precision (ppPR) and the corresponding preconditioned Fletcher-Reeves algorithms as shown in fig. 5.20, which plots the measure of convergence reached by the algorithm against the number of loadings.

Not refining the step and simply checking that an improved gradient has actually been found is, however, the best strategy to obtain the best performance from this search direction. This algorithm does at least slightly better than the corresponding Fletcher-Reeves and noticeably better than the one using the traditional search direction (see the examples in figs. 5.22 and 5.23).

Also using a number of MSA iterations before using the optimised line search is generally of advantage over using an optimised line search from the start, as can be seen again in figs. 5.22 and 5.23. Different numbers of MSA to start may have different degrees of effectiveness in accelerating the rate of convergence of the algorithm but the improvement over the corresponding algorithms using the traditional search direction is always noticeable, whilst there is little to choose between hybrid apFR and apPR algorithms, especially with more than 10 MSA iterations to start.

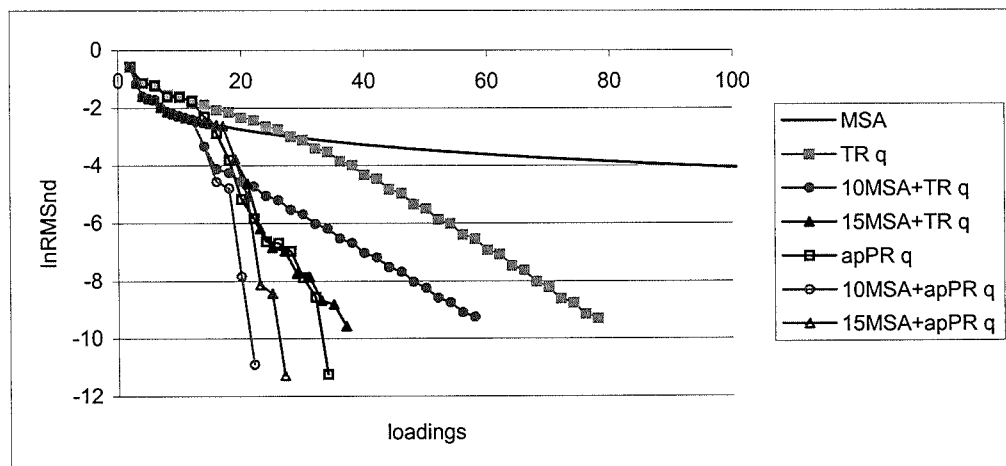


Fig. 5.22 – Comparison of the MSA, the algorithm using the traditional search direction (TR q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, paths set H2, Mendell-Elston approximation, 2 user classes.

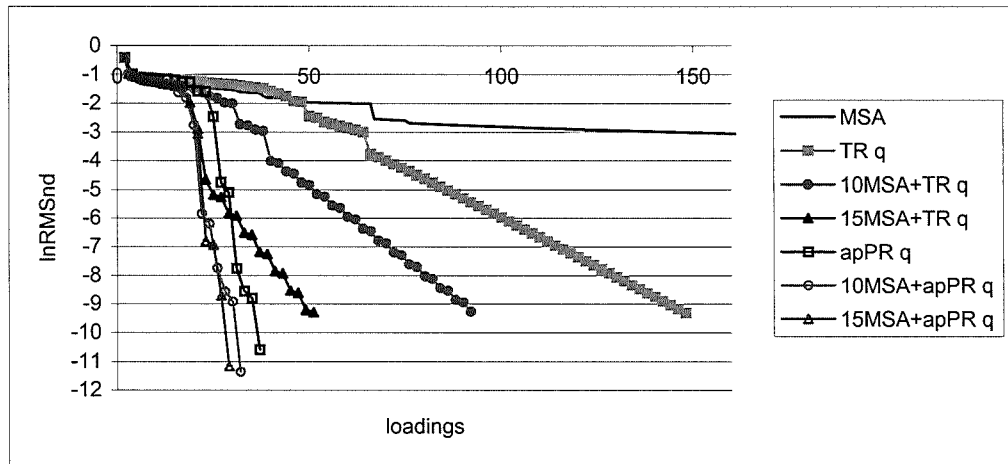


Fig. 5.23 – Comparison of the MSA, the algorithm using the traditional search direction (TR q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, paths set H1, Clark approximation, 3 user classes.

5.2.6 Conclusions

The multiple user classes framework proposed by Daganzo (1982) provides a flexible way to include users perceiving different costs in traffic assignment models. The objective function suggested by Daganzo (1982) can be elaborated to obtain the equivalent programme for MUC SUE (5.5) in terms of standardised costs that, although defined by an objective function that is not convex everywhere, has a unique solution.

Solution algorithms that are extensions of those for SUE studied in chapter 4 have been proposed to solve the MUC SUE problem using the objective function (5.5). The search direction linking the current and the auxiliary standardised link and path flow solutions, that is the equivalent of the traditional search direction in the MUC SUE problem, can be used in algorithms with fixed step and with step length obtained by interpolation. An MSA algorithm based on this search direction and an algorithm finding the step by quadratic interpolation of the objective function have been proposed and tested. Interpreting the previous search direction as a preconditioned steepest descent search direction, the idea of using preconditioned conjugate gradient algorithms has been put forward and in particular the

preconditioned Polak-Ribiere and Fletcher-Reeves algorithms, finding the step by quadratic interpolation of the objective function, have been suggested.

The numerical tests suggested no substantial differences between the cases when the algorithms were used with the improved Clark and with the Mendell-Elston approximation. The results showed that the MSA is outperformed by the other algorithms although its good initial efficiency supports the development of a hybrid algorithm starting with a number of MSA iterations before employing one of the other methods.

Refining the step along a search direction was found to be not efficient and the best strategy seemed to interpolate until an improved gradient is found, which in most cases coincided with carrying out a single interpolation. The algorithms using the traditional search direction with this step determination strategy were outperformed by those using the preconditioned Polak-Ribiere search direction, whilst those using the preconditioned Fletcher-Reeves showed some instabilities.

Starting these algorithms after a number of MSA iterations, that is using them in hybrid methods, gave the best overall performances. In particular, the algorithms using the traditional search direction are again outperformed by those using the preconditioned Polak-Ribiere one, which have efficiency similar to those using the preconditioned Fletcher-Reeves search direction.

5.3 Stochastic User Equilibrium Models with Elastic Demand (SUE ED)

5.3.1 Introduction

Elastic demand (ED) models refine the SUE framework by considering, in an aggregate way, drivers' responses to changes in network costs different from re-routing. The SUE and MUC SUE models considered so far, in fact, are known as fixed demand models since the OD matrix assigned is fixed and independent of the network conditions and, when the costs on the network change, the modelled drivers may only change the route they travel along.

However, two reports in the recent literature, one of which was issued by SACTRA (1994) and the other produced by MVA (1997), respectively concerned with “Trunk roads and the generation of traffic” and the “Impact of highway capacity reduction”, remarked on the need to model variable demand for obtaining more realistic results.

Although it is rather common practice to assume the demand fixed and independent of the network conditions, several possible, and sometimes dramatic, changes in transportation demand may be brought about by variations of transportation facility provision and availability and may take place in different temporal horizons. The SACTRA (1994) and the MVA (1997) reports characterised the following possible changes: route change (re-assignment), change in travel frequency, release of completely new journeys or suppression of existing journeys, journey re-timing, travel to new destinations (trip re-distribution), change in modal split, change in vehicle occupancy, and change in land use pattern.

The SACTRA suggested that future models should be extended to account for the complex and interrelated sets of land use and travel decisions that encompass, as the final stage, the choice of route through a network. While these more comprehensive models are being developed, the SACTRA report suggested using elastic demand modelling to account for some of those responses.

In Stochastic User Equilibrium assignment with Elastic Demand (SUE ED) the OD matrix entries are related to the network conditions between the relevant OD pair through an elastic demand function such as, for instance, the power function:

$$D_{rs} = D_{0rs} \left(\frac{S_{0rs}}{S_{rs}} \right)^e \quad (5.32)$$

where D_{rs} is the demand between the OD pair rs , D_{0rs} the relevant base value of the demand, S_{rs} the present expectation of the minimum perceived cost of travelling between r and s (the satisfaction), S_{0rs} its base value and e the elasticity. In (5.32) the current satisfaction S_{rs} is used as a summary measure of the cost of travelling between the relevant OD pair and determines the level of demand travelling. Assuming a positive elasticity (5.32) gives a decreasing level of demand with increasing current costs.

The power elastic demand function (5.32) is used in the present work as it is widely employed in the economic literature and uses a constant elasticity. However the models discussed are independent of the functional form of the demand functions. Other forms of elastic demand function can be used, as long as they are invertible. The commercial software SATURN (Van Vliet and Hall, 1993), for instance, allows also the use of logit or closed exponential function, of the exponential and of the elastic exponential or semi-log function (Hall *et al.*, 1992). Different sorts of elastic demand functions and their effect on trip suppression or release of long and short trips are discussed in Emmerson (1992).

Including elastic demand in an equilibrium traffic assignment model entails modelling an equilibrium situation where the traffic pattern is consistent with the network costs (as in the models in chapter 4) and also the level of traffic between each OD pair is consistent with the relevant cost of travelling (the deterministic cost between the OD pairs in the UE case and the expectation of the minimum perceived cost, the satisfaction, in the SUE case).

5.3.2 Elastic Demand Models in the Literature

Before looking at the Elastic Demand models in the literature it should be mentioned that they are not the only possible way to complement traffic assignment with aggregate models that vary the number of trips between each OD pair depending on the network conditions. In fact, also a number of heuristics used for this aim has been reviewed by the European Commission (1996) which, however, underlined that they account only for trip suppression due to congestion and lack a behavioural basis. The same report remarked that elastic demand models should be preferred as, beside being behaviourally sound, they account both for suppressed trips (in the case of worsening of network conditions) and for released trips (when network conditions improve).

In the deterministic User Equilibrium case the Elastic Demand problem is well studied and efficient algorithms exist in commercial software (see *e.g.* Hall *et al.* 1992, for a study on using the SATEASY module of SATURN for elastic demand). In fact, it had been studied already by Beckmann *et al.* (1956) and can be solved in an elegant way by network representation: it is reduced to a UE traffic assignment problem (and therefore can be solved with an UE algorithm) on an extended network where an added dummy link between each OD pair, with a cost function obtained as the inverse of the ED function, carries the demand not assigned to the real network (see *e.g.* Sheffi, 1985).

Within the SUE framework, the ED problem has not been reduced to an assignment problem on an extended network. However, as shown recently in the research work on the extension to ED of the SAM-SUE model (Maher and Hughes 1997b, 1998b; Hughes, 1998; Maher *et al.*, 1999 and Maher and Zhang, 2000), it can be solved with algorithms closely related to the ones for SUE traffic assignment with fixed demand. The methods proposed in that research work are described in the remainder of this section and have been the starting point for the algorithms and formulation discussed in the following sections.

Maher and Hughes (1998b) suggested solving the problem, in either the logit or the probit case, by solving separately but simultaneously for the equilibrium of the flows and of the demand. They posed the problem as one of minimising simultaneously the objective function for SUE:

$$z_{SUE}(\mathbf{x}) = -\sum_I \int_0^{x_i} c_i(u) du + \sum_I x_i c_i(x_i) - \sum_{RS} q_{rs} S_{rs}(\mathbf{x}) \quad (5.33)$$

proposed by Sheffi and Powell (1982) and introduced already in chapter 4, and an objective function for elastic demand:

$$z_{ED}(\mathbf{x}, \mathbf{q}) = \sum_{RS} q_{rs} S_{rs}(\mathbf{x}) - \sum_{RS} \int_{q_{0rs}}^{q_{rs}} D_{rs}^{-1}(q) dq \quad (5.34)$$

where rs is one of the RS OD pairs on the network, q_{rs} is the current flow between rs , S_{rs} is the satisfaction for the drivers travelling between rs , and $D_{rs}^{-1}(\cdot)$ is the inverse demand function for the OD pair rs . Calculating the gradient of Z_{ED} (5.34) w.r.t. the OD demands q_{rs} , it is possible to verify that a stationary point of this function is the point at which $S_{rs}(\mathbf{x}) = D_{rs}^{-1}(q_{rs})$, that is the ED equilibrium point, since the network costs and the demand are consistent with each other. Thus, the point at which both (5.33) and (5.34) are minimised is the SUEED equilibrium point for a network.

Simultaneous minimisation of (5.33) and (5.34) can be carried out as unconstrained as the solution satisfies the non-negativity of link and path flows and the consistency of the flows on the paths between an OD pair and those resulting from the elastic demand functions.

Maher and Hughes (1997b) proposed two algorithms working along the traditional search direction for solving (5.33) (as in SUE, see chapter 4) and along the vector linking the demands obtained from the current costs and the current demand for function (5.34).

The algorithms find the solution by moving iteratively of a step along each of the search directions:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda_1^{(n)}(\mathbf{y}^{(n)} - \mathbf{x}^{(n)}) \quad (5.35)$$

$$\mathbf{q}^{(n+1)} = \mathbf{q}^{(n)} + \lambda_2^{(n)}(\mathbf{t}^{(n)} - \mathbf{q}^{(n)}) \quad (5.36)$$

where $\mathbf{x}^{(n)}$ and $\mathbf{q}^{(n)}$ are respectively the current link flows and the current OD demands and $\mathbf{y}^{(n)}$ and $\mathbf{t}^{(n)}$ are their auxiliary counterparts, while $\lambda_1^{(n)}$ and $\lambda_2^{(n)}$ are the steps along the different search directions.

The two algorithms proposed by Maher and Hughes (1997b) differ in the way the step lengths are obtained. One algorithm is based on the MSA and both the steps are $1/(1+n)$ at each iteration. The second algorithm optimises the step length at each iteration by approximating the gradients of the two functions as linear in both the step lengths and obtaining the step lengths as the point along each search direction at which the relevant approximated gradient is zero. Numerical tests on a link based

case (Maher and Hughes, 1997b) showed that the second algorithm, named the Twin Step Algorithm (TSA), is more efficient than the MSA and it can be even more efficient if started with some MSA steps.

Maher and Hughes (1998b) proposed for the SUE ED problem the Balanced Demand Algorithm (BDA) which owes its name to the fact that at each iteration it finds the SUE solution along a search direction defined by two points satisfying the demand conditions, therefore with demand and network costs in balance. The algorithm is also based on the assumption that the points along that search direction satisfy approximately the demand conditions. The first of the two points satisfying the demand conditions and defining the search direction is given by the current link flow pattern and by the demand consistent with the satisfactions due to the current link flow pattern. The auxiliary set of link flows is found by loading such demand on the network according to the costs due to the current flow pattern. The auxiliary solution and the demand consistent with the corresponding costs give the point defining the other extreme of the search direction. The SUE objective function (5.33) is interpolated between these two points (by fitting a quadratic function, in the algorithm proposed) and the minimum point of the interpolating function is taken as optimal step length.

Maher and Hughes (1998b) remarked also that an alternative algorithm keeping the assigned flows in balance, rather than the demands, could be used. However, they did not consider it further as the computational costs of keeping the flows in SUE would be higher than the cost of keeping the demand in balance.

Also the BDA has been found to perform well in numerical tests, and its efficiency improves when it is started with a number of MSA iterations.

Maher *et. al.* (1999) devised a single objective function for SUE ED that is independent of the choice model:

$$\begin{aligned}
z_{SUEED}(\mathbf{x}, \mathbf{q}) = & -\sum_I \int_0^{x_i} c_i(u) du + \sum_I x_i c_i(x_i) - \sum_{RS} S_{rs}(\mathbf{x}) D_{rs}(S_{rs}(\mathbf{x})) + \\
& + \sum_{RS} D_{rs}^{-1}(q_{rs}) D_{rs}(S_{rs}(\mathbf{x})) + \sum_{RS} \int_{q_{0rs}}^{q_{rs}} D_{rs}^{-1}(q_{rs}) dq - \sum_{RS} q_{rs} D_{rs}^{-1}(q_{rs})
\end{aligned} \tag{5.37}$$

where i is one of the I links, c_i and x_i are respectively the cost and the traffic flow (of the unique user class considered in this case) on link i . The vector \mathbf{x} includes the flows of all links of the network, whilst q_{rs} and S_{rs} are respectively the total number of trips between the OD pair rs (one of the RS OD pairs of the network) and the expected minimum travel cost for the drivers travelling between that OD pair. The base value of the demand is q_{0rs} . Furthermore $D_{rs}(\cdot)$ is the strictly increasing demand function for the trips between the OD pair rs (the objective function can accommodate different demand functions or different parameters of the demand functions for different ODs, if necessary) and $D_{rs}^{-1}(\cdot)$ is its inverse.

The objective function (5.37) can be optimised as unconstrained, as the solution will satisfy the non negativity of path and link flows and the consistence of the sum of the path flows between an OD pair and the flows given by the elastic demand functions.

The minimum of the function (5.37) is the SUE ED point for a network as can be seen obtaining its gradient as in Maher *et al.* (1999). A term of the gradient with respect to the link flows results:

$$\frac{\partial z}{\partial x_j} = \left(x_j - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{pj} \right) \frac{dc_j}{dx_j} + \sum_{RS} (D_{rs}^{-1}(q_{rs}) - S_{rs}) \frac{\partial D_{rs}}{\partial x_j} \tag{5.38}$$

whilst differentiating with respect to the OD flows yields the following general term:

$$\frac{\partial z}{\partial q_{rs}} = (D_{rs}(S_{rs}) - q_{rs}) \frac{dD_{rs}^{-1}}{dq_{rs}} \tag{5.39}$$

(5.38) and (5.39) are both zero only at the SUE ED point that is when the current flows and demands are consistent with the current costs.

Maher and Zhang (2000) showed that (5.37) is locally convex around the solution, thus proving that the latter is a local optimum.

It is interesting to note that, when the demand is in balance the gradient (5.38) and (5.39) of the SUEED objective function reduces to the gradient (4.5) of the SUE objective function. This gives a theoretical framework for the use of the BDA.

5.3.3 A New Objective Function for SUE ED

This section proposes a new equivalent program for SUE ED that provides an alternative interesting framework for the BDA and can be obtained by modifying the SUE program of Sheffi and Powell (1982) written in terms of link flows or costs.

The equivalent SUE ED program is given by the objective function, which is minimised for the SUE ED link flows:

$$z_{SUEED}(\mathbf{x}) = -\sum_I \int_0^{x_i} c_i(\omega) d\omega + \sum_I x_i c_i(x_i) - \sum_{RS} \int_{S_{0rs}}^{S_{rs}(\mathbf{x})} D_{rs}(S_{rs}) dS_{rs} \quad (5.40)$$

or, by the equivalent objective function, which is minimised by the SUE ED link costs:

$$z_{SUEED}(\mathbf{c}) = \sum_I \int_{c_{0i}}^{c_i} c_a^{-1}(\omega) d\omega - \sum_{RS} \int_{S_{0rs}}^{S_{rs}(\mathbf{c})} D_{rs}(S_{rs}) dS_{rs} \quad (5.41)$$

Both objective functions can be minimised as unconstrained and the equivalence of their solution to the SUE ED point for a network can be shown by writing a typical term of their gradient. For (5.40), written in terms of link flows, assuming separable link costs and additive path costs, a term of the gradient can be obtained as detailed in appendix A3 and results in:

$$\frac{\partial z_{SUEED}(\mathbf{x})}{\partial x_i} = \left(x_i - \sum_{RS} D_{rs}(S_{rs}) \sum_K P_{rsk} \delta_{ik} \right) \frac{dc_i}{dx_i} = (x_i - y_i) \frac{dc_i}{dx_i} \quad (5.42)$$

where x_i is the i th component of the current link flow vector and y_i is the i th component of the balanced auxiliary flow vector, the vector of link flows obtained

by loading the demand consistent with current costs on the choice pattern consistent with the same costs. Assuming strictly increasing link cost functions, (5.42) is zero only when the flows obtained by assigning the result of the demand functions due to the current costs are consistent with the current flows. This corresponds to the SUEED conditions. In fact, the SUE conditions can be written as the consistency of the link flows on the network (Sheffi, 1985):

$$x_i - \sum_{rs} q_{rs} \sum_k P_{rsk} \delta_{ik} = 0 \quad \forall i \quad (5.43)$$

and the demand equilibrium as:

$$q_{rs} - D_{rs}(S_{rs}) = 0 \quad \forall rs \quad (5.44)$$

Assuming that the demand equilibrium condition is satisfied, as it is implied by programme (5.40), the SUE ED condition can be written as:

$$x_i - \sum_{rs} D_{rs}(S_{rs}) \sum_k P_{rsk} \delta_{ik} = 0 \quad \forall i \quad (5.45)$$

which is the term in brackets in (5.42).

A typical term of the gradient of the function (5.41) written in terms of link costs is derived in detail in appendix A4 and is:

$$\frac{\partial Z_{SUEED}(\mathbf{c})}{\partial c_i} = x_i - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{pi} = x_i - y_i \quad (5.46)$$

where the symbols are as for (5.42) and is, again, zero when the flows on the network and the demands (through the costs) are consistent and (5.45) is satisfied, that is at the SUE ED point.

The solution to the program (5.40) or (5.41) is unique in terms of link flows, link costs and demands and, when a set of paths between each OD is given, in terms of path flows and costs.

The uniqueness of the solution to (5.40) and (5.41) in terms of link flows and costs can be seen considering the respective Hessian matrices.

The Hessian of (5.40), obtained in appendix A3, is:

$$\begin{aligned} \nabla^2 z(\mathbf{x})_{SUEED} = & \nabla_x \mathbf{c} + \nabla_x^2 \mathbf{c} \mathbf{R} + \sum_{RS} \left(-\frac{dD_{rs}(S_{rs})}{dS_{rs}} \right) (\nabla_x \mathbf{c} \Delta_{rs} \mathbf{P}_{rs}) (\nabla_x \mathbf{c} \Delta_{rs} \mathbf{P}_{rs})^T + \\ & + \sum_{RS} D_{rs}(S_{rs}) (\nabla_x \mathbf{c} \Delta_{rs}) (-\nabla_e \mathbf{P}_{rs}) (\nabla_x \mathbf{c} \Delta_{rs})^T \end{aligned} \quad (5.47)$$

Similarly to the SUE case, in general it is not possible to characterise the sign of this Hessian, though it is positive definite at equilibrium. The first matrix in the expression of the Hessian is the diagonal Jacobian of the link costs and is positive definite thanks to the assumption of strictly increasing link costs. The second matrix is the product of the Hessian of the link costs by the diagonal matrix \mathbf{R} of differences between the auxiliary and the current link flows. The components of the matrix \mathbf{R} can take any sign but will be zero at equilibrium, where this component vanishes. The third term is the summation of positive semidefinite terms built up by a positive number (the opposite of the derivative of the demand function, which is negative by definition) and a positive semidefinite matrix (it is only semidefinite due to the definition of Δ_{rs}). The fourth and last term is the summation of positive semidefinite components given by the product of the demand (a positive number) and a quadratic form applied to a positive definite matrix (the opposite of the Jacobian of the path choice probabilities). At the solution point the Hessian (5.47) is given by the sum of a positive definite matrix and a number of positive semidefinite matrices and is thus positive definite. The minimum of (5.40) is therefore a local optimum.

The Hessian of (5.41), obtained in appendix A4, is, in matrix form:

$$\begin{aligned} \nabla^2 z_{SUEED}(\mathbf{c}) = & \nabla_c \mathbf{c}^{-1} + \sum_{RS} \left(-\frac{dD_{rs}(S_{rs})}{dS_{rs}} \right) (\Delta_{rs} \mathbf{P}_{rs}) (\Delta_{rs} \mathbf{P}_{rs})^T + \\ & + \sum_{RS} D_{rs}(S_{rs}) (\Delta_{rs}) (-\nabla_e \mathbf{P}_{rs}) (\Delta_{rs})^T \end{aligned} \quad (5.48)$$

The first term is positive definite, according to the definition of the link costs assumed. The second term is the summation of elements given by a positive number, the opposite of the derivative of the demand function and by a positive semidefinite matrix as it can be seen as a quadratic form. The third term is a summation of positive semidefinite matrices, as the Jacobian of the path choice probabilities is negative definite. Therefore the Hessian (5.48) is the sum of a positive definite matrix and of a number of positive semidefinite ones and is thus always positive definite and the function (5.41) is convex in all its domain and its minimum is a global minimum.

The one to one correspondence between link flows and costs means that also the solution to (5.40) is unique although that function is not necessarily convex away from the solution.

The analysis of Hessian (5.48) shows the global uniqueness in terms of link flows and costs. Given a set of paths, the uniqueness of the link costs entails the uniqueness of the satisfaction between each OD pair and therefore also of the demand. It also entails the uniqueness of the path costs, and thus of the path choice pattern between each OD pair and, as a result of the uniqueness of the demands, also the uniqueness of the paths flows.

When the demand is fixed, the objective functions (5.40) and (5.41) reduce to objective functions equivalent to those for SUE. In fact, in that case they differ from the SUE ones by a constant term (irrelevant for the optimisation) that is the summation, over all OD pairs, of the constant demand by the reference satisfaction.

The new objective function (5.40) gives an alternative framework for the Balanced Demand Algorithm proposed by Maher and Hughes (1998b). In fact, the relation between the BDA and (5.40) parallels the one between the traditional SUE search direction and the SUE objective function (4.3). The BDA search direction is the opposite of the term in brackets in (5.42) therefore the dot product of (5.42), the gradient of the new SUE ED objective function (5.40), and of the BDA search

direction is always negative, thus ensuring that the BDA search direction is always a descent direction for the SUE ED problem.

5.3.3.1 Relationship Between SUE ED Objective Functions

The objective function of Maher *et al.* (1999) and the new objective function (5.40) for SUEED proposed in the previous section are related by the demand balancing and by a change of variable.

The first two terms of (5.37) and (5.40), referring to the current flows and costs, are the same in the two expressions.

The demand considered in (5.40) is in balance with the current costs, as the integral with upper integration limits $S_{rs}(\mathbf{x})$ implies. This entails that the following two expressions are satisfied:

$$q_{rs} = D_{rs}(S_{rs}(\mathbf{x})) \quad (5.49)$$

$$S_{rs}(\mathbf{x}) = D_{rs}^{-1}(q_{rs}) \quad (5.50)$$

Considering the demand as being in balance, the third and fourth terms in (5.37) which are:

$$-\sum_{RS} S_{rs}(\mathbf{x})D(S_{rs}(\mathbf{x})) + \sum_{RS} D_{rs}^{-1}(q_{rs})D(S_{rs}(\mathbf{x})) \quad (5.51)$$

can be rewritten substituting (5.50) for S_{rs} in the first term:

$$-\sum_{RS} D_{rs}^{-1}(q_{rs})D(S_{rs}(\mathbf{x})) + \sum_{RS} D_{rs}^{-1}(q_{rs})D(S_{rs}(\mathbf{x})) \quad (5.52)$$

which is equal to zero. An equivalent result is obtained substituting (5.49) for q_{rs} in the second term.

Furthermore, considering (5.50) and writing:

$$dS_{rs} = \frac{dD_{rs}^{-1}(q_{rs})}{dq_{rs}} dq_{rs} \quad (5.53)$$

the third term in (5.40) can be written as a function of $q_{rs}=D_{rs}(S_{rs}(\mathbf{x}))$:

$$-\sum_{RS} \int_{S_{0rs}}^{S_{rs}(\mathbf{x})} D_{rs}(S_{rs}) dS_{rs} = -\sum_{RS} \int_{q_{0rs}}^{q_{rs}(\mathbf{x})} \omega \frac{dD_{rs}^{-1}(\omega)}{d\omega} d\omega \quad (5.54)$$

Integrating this by parts gives:

$$\begin{aligned} &-\sum_{RS} \int_{q_{0rs}}^{q_{rs}(\mathbf{x})} \omega \frac{dD_{rs}^{-1}(\omega)}{d\omega} d\omega = \\ &-\sum_{RS} D_{rs}^{-1}(q_{rs}(\mathbf{x})) q_{rs}(\mathbf{x}) + \sum_{RS} D_{rs}^{-1}(q_{0rs}) q_{0rs} + \sum_{rs} \int_{q_{0rs}}^{q_{rs}(\mathbf{x})} D_{rs}^{-1}(\omega) d\omega \end{aligned} \quad (5.55)$$

The second term of (5.55) is a constant and can be disregarded as it does not change the point at which the objective function is minimised. The other two terms coincide with the fifth and sixth term in (5.37). Therefore (5.40) can be read as a section of (5.37) obtained when the demand is in balance.

5.3.4 Algorithms for the Probit Path-Based SUE ED Problem

5.3.4.1 Introduction

Building on the studies in the literature outlined above and on the new objective function proposed in section 5.3.3, a number of algorithms solving the SUE ED problem by minimising the objective function (5.40) can be put forward for path-based probit, and similar methods could be used also in link-based cases and with other choice models solved analytically.

The algorithms suggested in this section are presented without reporting on numerical tests. Information on the performance of such algorithms can be, however, inferred (at least to an extent) from the efficiency data for similar algorithms adapted for solving the MUC SUE ED problem, studied later in this chapter, as that problem is an extension of the SUE ED one and is solved with algorithms that are extensions of those suggested here.

With all the algorithms, the value of the satisfaction function, or expected minimum cost between an origin-destination pair, must be determined, not only to drive the

algorithms, but also to calculate the variation of the demand as a function of the network conditions, as it is the independent variable of the elastic demand functions. In chapter 3 a number of alternative ways to approximate the value of such function have been delineated.

5.3.4.2 SUE ED Path-Based Algorithms with BDA Search Direction

Building on the discussion on the algorithms for SUE and for MUC SUE, the BDA search direction can be used along with methods for determining the step length according to a predefined rule or by using information on the local shape of the objective function.

The first type of algorithm includes the MSA, that gives the step as $(1+n)^{-1}$ where n is the current iteration number, and similar methods such as the MWA proposed in chapter 4. An MSA working along the BDA search direction has already been proposed by Maher and Hughes (1998b).

The second type of algorithms couples the BDA search direction with any one of the optimised line search method tested in chapter 4. When the objective function is approximated with a quadratic along the search direction using the values at its two extremes, the original version of the BDA algorithm put forward by Maher and Hughes (1998b) is obtained.

The interpolation methods presented in chapter 4 use the values of the gradient and, in the cubic case, of the objective function at different points along the search direction. The objective function (5.40) can be readily evaluated especially as in SUE ED problems the values of the satisfaction between each OD pair are used also to perform the balanced stochastic loadings.

The value of the gradient of (5.40) along a search direction can be calculated working along the same lines followed to obtain the gradient of the SUE and of the MUC SUE objective functions along a search direction. In more detail, the components of a point $x_i^{(\lambda)}$ along the search direction at iteration n can be written as a function of the step λ taken along the search direction as

$$x_i^{(\lambda)} = x_i^{(n)} + \lambda(y_i^{(n)} - x_i^{(n)}) \quad (5.55)$$

where $x_i^{(n)}$ is the current flow on link i at iteration n , $y_i^{(n)}$ is the auxiliary flow on link i at iteration n , obtained performing a balanced loading on the network costs due to the flows $x_i^{(n)}$, consistent with the definition of the balanced demand search direction.

Then, exploiting the chain rule of derivation, the derivative of the MUC SUE objective function (5.40) along the search direction can be written as:

$$\frac{dz_{SUEED}}{d\lambda} = \sum_I \frac{\partial z_{SUEED}}{\partial x_i} \frac{dx_i}{d\lambda} \quad (5.56)$$

The first term is simply the gradient of the SUE ED objective function, already obtained in (5.42), that is rewritten here as taken w.r.t. $x_i^{(\lambda)}$:

$$\frac{\partial z_{SUEED}}{\partial x_i^{(\lambda)}} = \left(-y_i^{(\lambda)} + x_i^{(\lambda)}\right) \frac{dc_i}{dx_i^{(\lambda)}} \quad (5.57)$$

The definition of $x_i^{(\lambda)}$ above gives immediately

$$\frac{dx_i^{(\lambda)}}{d\lambda} = y_i^{(n)} - x_i^{(n)} \quad (5.58)$$

thus (5.56) can be rewritten as:

$$\frac{dz_{SUEED}}{d\lambda} = \sum_I \left(-y_i^{(\lambda)} + x_i^{(\lambda)}\right) \frac{dc_i}{dx_i^{(\lambda)}} \left(y_i^{(n)} - x_i^{(n)}\right) \quad (5.59)$$

The gradient at a point along the search direction is obtained by substituting the flows defining that point and the relevant auxiliary solution for $\mathbf{x}^{(\lambda)}$ and $\mathbf{y}^{(\lambda)}$.

Therefore, whilst the MSA requires simply a balanced demand loading to determine the search direction, using one of the interpolation methods proposed in chapter 4 will require one additional balanced loading for each point along the search direction at which function and gradient information are evaluated. In the case of the quadratic interpolation carried out once, two loadings are necessary at each iteration (including the one necessary to determine the search direction). If the interpolation is refined in

subintervals one more loading will be required for each additional subinterval examined.

As in the SUE and MUC SUE case algorithms calculating the step by interpolation can be used as the second part of hybrid algorithms, after starting solving the problem with a number of MSA iterations.

5.3.4.3 SUE ED Path-Based Algorithms with Alternative Search Directions

The parallel between the BDA search direction for the SUE ED problem and the traditional search direction for the SUE problem mentioned in 5.3.3 can be taken forward by interpreting the BDA search direction as a preconditioned steepest descent direction for the SUE ED problem expressed by the objective function (5.40).

Considering the gradient (5.42) of the objective function (5.40) the BDA search direction \mathbf{t} can be written as:

$$\mathbf{t} = \mathbf{y} - \mathbf{x} = \mathbf{J}\mathbf{J}^{-1}(\mathbf{y} - \mathbf{x}) = \mathbf{J}^{-1}\nabla_{\mathbf{x}}z(\mathbf{x}) \quad (5.60)$$

Where \mathbf{J} is the Jacobian of the link costs, defined as in the SUE case. The (5.60) is similar to (4.35) that showed an analogous relationship for the SUE case. Recalling the consideration in 4.3.5, the BDA search direction can be analogously seen as the expression, in the space of the link flows, of the steepest descent search direction in the space given by the change of coordinates defined by the square root of the Jacobian of the current link costs.

Then, similarly to the SUE case, preconditioned search directions for the SUE ED problem can be put forward. For instance preconditioned conjugate gradient search directions similar to those considered in chapter 4 can be used. The final formulae for determining, for instance, the preconditioned Fletcher-Reeves and Polak-Ribiere search direction for (5.40) are identical to (4.55), (4.58), (4.59) and (4.60) except for the fact that the gradients now refer to (5.40) and the auxiliary solutions are balanced auxiliary solutions.

When calculating the value of the gradient of the objective function along a search direction different from the BDA one for the computation of the optimal step, (5.58) should be changed by substituting $y^{(n)}$ with the flows at the extreme of the modified search direction.

Also the algorithms using modified search directions can be used after starting to solve the SUE ED problem with a number of MSA iterations.

5.3.5 Conclusions

Elastic Demand models can include in traffic assignment models transportation choices different from the one of route by representing them in an aggregate way. Their use has been encouraged until more complex models, accounting explicitly for the many transportation choices affected by changes in network costs, are available.

In the User Equilibrium case the elastic demand problem is well studied and solved reducing it to an assignment on an extended network and only recently the SUE ED problem has been posed so that it can be solved with algorithms closely related to the ones for SUE traffic assignment with fixed demand (Maher and Hughes, 1998b).

The study in this section built on the work of Maher *et al.* (1999) and proposed a new equivalent programme that reduces the SUE ED problem to one in terms of link flows or link costs, which has a unique solution. The relationship between the new programme and the one of Maher *et al.* (1999) has been explained.

A number of algorithms have been proposed, building on those already put forward by Maher *et al.* (1999) and basing them on the formulation of the problem proposed in this section.

5.4 Stochastic User Equilibrium with Multiple User Classes and Elastic Demand (MUC SUE ED)

5.4.1 Introduction

SUE with multiple user classes (MUC) and elastic demand (ED) can be considered together by integrating the frameworks and the algorithms presented in the previous sections. For each user class, the resulting MUC SUE ED models assign OD flows, that are a function of the costs of travelling between each OD pair for each user class, on the path choice pattern resulting from the costs perceived by that user class. At the MUC SUE ED point for a network, flows, costs and demands for each of the user classes modelled are separately in equilibrium. In other words at the MUC SUE ED point no user can improve his perceived cost by changing route or deciding not to travel in the modelled period.

5.4.2 MUC SUE ED Models in the Literature

The MUC SUE ED problem has been considered in the literature by Daganzo (1983) who expressed it as a fixed point problem and solved it with the MSA, to consider also non separable link costs, within the MUC cost and flow structure already appeared in Daganzo (1982) and explained in 5.2.3.

Maher and Zhang (2000) considered the MUC SUE ED problem when the costs for the different user classes are separable and multiples of each other or when they perceive the same systematic cost but with different perception errors (constant across the network), as these settings are equivalent for the scalability of the choice models. They considered the case of separable link costs and put forward objective functions, which are extensions of (5.37) that they devised for SUE ED. They also proposed to solve the problem using algorithms that are extensions of those they used for the SUE ED problem, which are described in section 5.3.

5.4.3 An Objective Function for MUC SUE ED

A MUC SUE ED programme for the multiple user classes framework of Daganzo (1982) and for separable link costs and additive path costs can be written by considering together the programmes for MUC SUE (5.3) and SUE ED (5.37). The resulting programme can be optimised as unconstrained since at the solution path and link flows will satisfy the non-negativity constraints and the path flows between each OD pair will be consistent with those resulting from the elastic demand functions.

The MUC SUE ED programme is expressed by the minimisation of the following objective function:

$$\begin{aligned}
 z(\mathbf{v}, \mathbf{q})_{MUCSUEED} = & -\sum_I \int_0^{v_i} b_i(\omega) d\omega + \sum_I v_i b_i(v_i) - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} S_{rs}^{(k)}(\mathbf{v}) D_{rs}^{(k)}(S_{rs}^{(k)}(\mathbf{v})) + \\
 & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)-1}(q_{rs}^{(k)}) D_{rs}^{(k)}(S_{rs}^{(k)}(\mathbf{v})) + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \int_{q_{rs}^{(k)}}^{q_{rs}^{(k)}} D_{rs}^{(k)-1}(\sigma) d\sigma + \\
 & - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} D_{rs}^{(k)-1}(q_{rs}^{(k)})
 \end{aligned} \tag{5.61}$$

where the control variables are the standardised link flows \mathbf{v} (defined in 5.1) and the OD flows \mathbf{q} . The subscript (k) refers to the user class k of the K present, $\alpha^{(k)}$ and $\beta^{(k)}$ are as for (5.1) and (5.2), $D_{rs}^{(k)}(\cdot)$ and $D_{rs}^{(k)-1}(\cdot)$ are the demand function and its inverse for the OD pair rs and class k and $S_{rs}^{(k)}$ is the corresponding satisfaction.

If one user class only is considered, the (5.61) is equivalent to the objective function for SUE ED (5.37). Similarly, if no elastic demand is considered (5.61) becomes equivalent to the objective function for MUC SUE in terms of link flows. Finally, (5.61) becomes equivalent to the objective function for SUE in terms of link flows if MUC and ED are excluded.

The characteristics of (5.61) are those of the SUE ED programme of Maher *et al.* (1999): it is independent of the choice model, it can be used with any elastic demand

function and also with different elastic demand functions or different parameters for the functions for each OD pair and user class.

The equivalence of the solution of (5.61) to the MUC SUE ED problem can be shown by considering the MUC SUE ED conditions for a network written in terms of standardised link flows and OD flows by user class:

$$v_i - \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} = 0 \quad \forall i \quad (5.62)$$

$$q_{rs}^{(k)} - D_{rs}^{(k)}(S_{rs}^{(k)}) = 0 \quad \forall rs, k \quad (5.63)$$

(5.63) is also equivalent to:

$$D_{rs}^{(k)-1}(q_{rs}^{(k)}) - S_{rs}^{(k)} = 0 \quad \forall rs, k \quad (5.64)$$

The typical terms of the gradient of (5.61) are obtained in detail in appendix A5 and result in:

$$\begin{aligned} \frac{\partial Z_{MUCSUEED}}{\partial v_i} = & \left(v_i - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{pi} \right) \frac{db_i}{dv_i} + \\ & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1}(q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial D_{rs}^{(k)}}{\partial v_i} \end{aligned} \quad (5.65)$$

$$\frac{\partial Z_{MUCSUEED}}{\partial q_{rs}^{(k)}} = \frac{\alpha^{(k)}}{\beta^{(k)}} \left(D_{rs}^{(k)}(S_{rs}^{(k)}) - q_{rs}^{(k)} \right) \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} \quad (5.66)$$

respectively w.r.t. the standardised link flows and the OD trips by class.

The gradient terms (5.65) and (5.66) are both zero when (5.62) and the equivalent (5.63) and (5.64) are satisfied, that is at the MUC SUE ED point for a network.

Considerations analogous to those presented by Maher and Zhang (2000) for the SUE ED programme can be used to show that the Hessian of (5.61) is positive definite at equilibrium and, therefore, that such point is a local optimum. The differences between the two proofs are due to the introduction of several user classes.

The Hessian of (5.61) at equilibrium can be obtained as detailed in appendix A5. It can be written in partitioned form as:

$$\begin{aligned} \nabla^2 z_{MUCSUEED} &= \begin{bmatrix} \left[\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial q_{tu}^{(l)}} \right] & \left[\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial q_{tu}^{(l)}} \right] \\ \left[\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial v_i} \right] & \left[\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial v_j} \right] \end{bmatrix} \\ &= \begin{bmatrix} \nabla_{\mathbf{q}}^2 z_{MUCSUEED} & \nabla_{\mathbf{vq}}^2 z_{MUCSUEED} \\ \nabla_{\mathbf{qv}}^2 z_{MUCSUEED} & \nabla_{\mathbf{v}}^2 z_{MUCSUEED} \end{bmatrix} \end{aligned} \quad (5.67)$$

The expressions of the general terms of each of the four submatrices are reported in appendix A5.

Similarly to the case analysed by Maher and Zhang (2000), the positive definiteness of (5.67) can be shown considering it as a Hermitian matrix, since a real Hermitian matrix is a real symmetric matrix. A Hermitian matrix, partitioned as $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}$ with \mathbf{A} and \mathbf{C} being square matrices, is positive definite if and only if \mathbf{A} is positive definite and $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ is positive definite (this theorem is reported *e.g.* in Horn and Johnson, 1990).

Taking as \mathbf{A} , \mathbf{B} and \mathbf{C} the corresponding submatrices in (5.67), \mathbf{A} is the diagonal matrix given by (see appendix A5):

$$\mathbf{A} = \nabla_{\mathbf{q}}^2 z_{MUCSUEED} = \text{diag} \left[-\frac{\alpha^{(k)}}{\beta^{(k)}} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} \right] \quad (5.68)$$

and is positive definite since the inverse demand functions are strictly decreasing thus the elements of the diagonal are all positive.

The inverse of \mathbf{A} is:

$$\mathbf{A}^{-1} = \text{diag} \left[-\frac{\beta^{(k)}}{\alpha^{(k)}} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \right] \quad (5.69)$$

and is positive definite since all its diagonal terms are positive.

The matrix \mathbf{B} is a rectangular matrix with general term given by:

$$\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial v_i} = \frac{\alpha^{(k)} \partial S_{rs}^{(k)}}{\beta^{(k)} \partial v_i} \quad (5.70)$$

(see again appendix A5 for the derivation). The product $\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ is complicated, with respect to that for the SUE ED objective function considered by Maher and Zhang (2000), by the presence of the multiple user classes. It results in a matrix that can be shown to have general term:

$$\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \left(-\frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} \right) \frac{\partial S_{rs}^{(k)}}{\partial v_j} \quad (5.71)$$

In matrix form this can be written as:

$$\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} = \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} (\nabla_{\mathbf{v}} S_{rs}^{(k)}) \left[\text{diag} \left(-\frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} \right) \right] (\nabla_{\mathbf{v}} S_{rs}^{(k)})^T \quad (5.72)$$

The submatrix \mathbf{C} , again from appendix A5, is:

$$\begin{aligned} \nabla_{\mathbf{v}}^2 z_{MUCSUEED} = & \nabla_{\mathbf{v}} \mathbf{b} + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs} (\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)}) \left(-\nabla_{e^{(k)}} P_{rs}^{(k)} \right) (\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)})^T + \\ & + 2 \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} (\nabla_{\mathbf{v}} S_{rs}^{(k)}) \left[\text{diag} \left(-\frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} \right) \right] (\nabla_{\mathbf{v}} S_{rs}^{(k)})^T \end{aligned} \quad (5.73)$$

Thus $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ can be written by subtracting (5.72) from \mathbf{C} , which results in:

$$\begin{aligned} \mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} = & \nabla_{\mathbf{v}} \mathbf{b} + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs} (\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)}) \left(-\nabla_{e^{(k)}} P_{rs}^{(k)} \right) (\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)})^T + \\ & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} (\nabla_{\mathbf{v}} S_{rs}^{(k)}) \left[\text{diag} \left(-\frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} \right) \right] (\nabla_{\mathbf{v}} S_{rs}^{(k)})^T \end{aligned} \quad (5.74)$$

This matrix is positive definite since it is the sum of a positive definite matrix and a number of positive semidefinite matrices. In fact, the first matrix building it up is the

Jacobian of the common part of the link costs, which is positive definite by definition. The second term is a summation of matrices given by quadratic forms of positive definite matrices (the opposites of the Jacobians of the path choice probabilities for each user class) that result positive semidefinite (due to the definition of the link-path incidence matrix). Similarly, the third term is the summation of quadratic forms of positive definite matrices, that result positive semidefinite.

Thus, since both \mathbf{A} and $\mathbf{C}-\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ are positive definite, the Hessian of (5.61) is positive definite at equilibrium and such point is a local minimum.

5.4.4 An Alternative Objective Function for MUC SUE ED

A further equivalent programme for MUC SUE ED can be developed in the vein of the programme (5.40) or (5.41) proposed for SUE ED, considering the MUC framework of Daganzo (1982), separable link costs and additive path costs.

The MUC SUE ED flows for a network can be found by minimising the objective function:

$$Z_{MUCSUEED}(\mathbf{v}) = -\sum_I \int_0^{v_i} b_i(\omega) d\omega + \sum_I v_i b_i(v_i) - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \int_{S_{0rs}^{(k)}}^{S_{rs}^{(k)}(\mathbf{v})} D_{rs}^{(k)}(\sigma) d\sigma \quad (5.75)$$

Alternatively the MUC SUE ED link costs and OD flows on a network can be obtained by minimising the equivalent objective function (linked to the previous one by a change of variable):

$$Z_{MUCSUEED}(\mathbf{b}) = -\sum_I \int_0^{\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}}} b_i^{-1}(\omega) d\omega - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \int_{S_{0rs}^{(k)}}^{S_{rs}^{(k)}(\mathbf{b})} D_{rs}^{(k)}(\sigma) d\sigma \quad (5.76)$$

As with the objective function (5.61), the cases when either elastic demand or multiple user classes or both are not considered can be seen as particular cases of this formulation. So (5.75) and (5.76) reduce to a function equivalent to the (5.5) and (5.4) when the demand is fixed, to (5.40) and (5.41) when there is only one user

class (for link costs defined in both cases as in (5.1)), and to (4.3) and (4.4) when one user class only and fixed demand are considered.

The MUC SUE and demand equilibrium conditions for a network have been reported above in (5.62) and (5.63). They can be combined, if it is assumed that the demand equilibrium is always satisfied, as it is implied by the objective functions above, to obtain the MUC SUE ED condition:

$$v_i - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)} (S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{pi} = 0 \quad \forall i \quad (5.77)$$

This point is the MUC SUE ED point for the network. In fact, the consistency of the standardised link flows implies the consistency of the link costs by user class. Given a set of paths for each class between each OD pair, the consistency of the link costs by user class entails the consistency of the choice pattern and of the satisfaction (and thus of the demand) between each OD pair and for each user class. Thus each user class is in equilibrium.

The equivalence of the solution to (5.75) and (5.76) to the MUC SUE ED point for a network can be verified by calculating their gradient.

A term of the gradient of (5.75) w.r.t. the standard link flows can be obtained as detailed in appendix A6 and results in:

$$\frac{\partial Z_{MUCSUEED}(\mathbf{v})}{\partial v_i} = \left(v_i - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)} (S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \right) \frac{db_i}{dv_i} = (v_i - w_i) \frac{db_i}{dv_i} \quad (5.78)$$

where the w_i are the auxiliary link flows resulting from loading the demand consistent with the current costs on the path choice pattern resulting from the current costs and the derivatives of the common part of the link costs are strictly positive as the link performance functions are assumed to be strictly increasing.

Differentiating (5.76) w.r.t. the common part of the link costs (as reported in appendix A7) yields the general term of the gradient:

$$\frac{\partial z_{MUCSUEED}(\mathbf{b})}{\partial b_i} = v_i - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)} \left(S_{rs}^{(k)} \right) \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} = v_i - w_i \quad (5.79)$$

In both cases the gradient will be zero only when the current standardised flows coincide with the auxiliary standardised flows obtained by loading the demand consistent with the current costs on the choice pattern resulting from the current costs for the relevant user class that is when the MUC SUE ED condition (5.77) is satisfied.

The analysis of the Hessian of the two versions of the objective function, and therefore of their convexity, parallels that of the Hessian of the SUE programme of Sheffi and Powell (1982) and of the programme (5.40) or (5.41) for SUE ED.

The Hessian of (5.75) is (the derivation is detailed in appendix A6):

$$\begin{aligned} \nabla^2 z_{MUCSUEED}(\mathbf{v}) = & \nabla_{\mathbf{v}} \mathbf{b} + \mathbf{R} \nabla_{\mathbf{v}}^2 \mathbf{b} + \\ & + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} \left(- \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{S_{rs}^{(k)}} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)} \right)^T + \\ & + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} D_{rs}^{(k)} \left(S_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} b \Delta_{rs}^{(k)} \right) \left(- \nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \right)^T \end{aligned} \quad (5.80)$$

and is positive definite at the equilibrium point although its sign elsewhere is not known a priori. This can be seen analysing each of the terms building it up. The first term is the Jacobian of the common part of the link costs, and is diagonal and positive definite thanks to the assumption of separable and strictly increasing costs. The sign of the second term is conditioned by the sign of \mathbf{R} , the diagonal matrix of the difference between the present and the auxiliary balanced solution in terms of standardised link flows, which can take any sign in general but will be zero at equilibrium. The third term is the summation of a number of products of a positive number (the opposite of the derivative of the demand function) by a positive semidefinite matrix. The fourth term is positive semidefinite as it includes the quadratic form of a positive definite matrix (the opposite of the Jacobian of the path choice probabilities) and as the quadratic form includes the link-path incidence matrices. Thus, when \mathbf{R} vanishes at equilibrium, the Hessian is positive definite as it

is the sum of a positive definite matrix and a number of positive semidefinite matrices.

This means that, although nothing can be said about the shape of (5.75) away from the solution, it is convex at the solution and the MUC SUE ED point is a local minimum. To show that it is also a global minimum, it is necessary to look at the Hessian of the demand function in terms of common parts of the link costs.

Differentiating once more the gradient defined in (5.79) w.r.t. the common part of the link costs yields the Hessian of (5.76) that results in:

$$\begin{aligned} \nabla^2 z_{MUCSUEED}(\mathbf{b}) = & \nabla_b \mathbf{b}^{-1} + \sum_K \alpha^{(k)} \sum_{RS} \left(-\frac{dD_{rs}^{(k)}}{dS_{rs}^k} \right) \left(\Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)} \right) \left(\Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)} \right)^T + \\ & + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} D_{rs}^{(k)} \left(S_{rs}^{(k)} \right) \left(\Delta_{rs}^{(k)} \left(-\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \Delta_{rs}^{(k)T} \right) \end{aligned} \quad (5.81)$$

This Hessian is derived in detail in appendix A7 and is positive definite over all the domain of the function, which is therefore convex and has a unique solution.

To check the positive definiteness of (5.81) consider that its first term is diagonal and positive definite (because of the definition of the common part of the link costs: separable and strictly increasing) and the other terms are positive semidefinite. In fact the second term is the product of a positive number (the opposite of the derivative of the demand function) and a positive semidefinite matrix. The third term is the summation of products of positive numbers by a number of positive semidefinite matrices, since the opposite of the Jacobian of the path choice probabilities is positive definite and is included in a quadratic form defined by the link-path incidence matrix.

The one to one correspondence between link flows and costs ensures that the solution to either problem is unique in terms of common link costs and flows.

Moreover, the uniqueness of the common part of the link costs implies the uniqueness of the link costs by user class. Given a set of paths for each class between each OD pair, the uniqueness of the link costs by user class entails the uniqueness of the choice pattern and of the satisfaction (and thus of the demand)

between each OD pair and for each user class. Thus, demands between each OD pair, path flows and path costs are unique for each user class.

5.4.5 Algorithms for the Probit Path-Based MUC SUE ED Problem

5.4.5.1 Introduction

This section describes alternative algorithms that can be used for the solution of the MUC SUE ED problem and that are tested in the next section.

The algorithms proposed solve for MUC SUE ED by minimising (5.75) and are introduced by referring to the description of the corresponding algorithms for SUE reported in chapter 4 as they are extensions of those proposed for SUE, MUC SUE and suggested for SUE ED. They allow us to follow the evolution of the current solution in terms of both path and link flows and as in the SUE, MUC SUE and SUE ED cases they can be also be used in link-based implementations and with other choice models that can be written in closed form or approximated analytically.

The programme defined by the objective function (5.75) has been used to solve the problem in a path-based case, although it refers to link data, as it is convenient to use and has a proven unique solution.

5.4.5.2 The MSA Algorithm for MUC SUE ED

The MSA algorithm for MUC SUE ED is an extension of the corresponding algorithms for SUE ED.

As in general in the MSA, the step is predetermined and given, at each iteration, by $(1+n)^{-1}$ where n is the iteration number. The difference from the other MSA is in the search direction, consistent with the fact that the function optimised is (5.75). The search direction is similar to the one used in the Balanced Demand Algorithm but with the inclusion of multiple user classes and standardised flows. It can be termed the MUC BDA search direction and its i th component is:

$$\sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)} (S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{pi} - v_i \quad (5.82)$$

(5.82) can also be seen in connection with the short objective function for MUC SUE ED (5.75) as corresponding to the traditional search direction for SUE. As with the SUE traditional search direction, taking the dot product with the gradient of (5.75) it is possible to show that (5.82) is always a descent direction.

Given a set of paths between each OD, the search direction expressed in terms of paths corresponds uniquely to the search direction expressed in terms of link flows as in (5.82) therefore the same step applies and the progress of the solution can be followed in either link or path flow terms.

5.4.5.3 MUC SUE ED Algorithms with MUC BDA Search Direction and Optimised Line Search

The MUC BDA search direction (5.82) can be used in conjunction with any of the optimised step calculation methods described and tested in chapter 4. It is considered here only with the quadratic interpolation method to calculate the step, to focus on the comparison between search directions rather than between different step calculation methods, also having seen from chapter 4 that the quadratic method is robust and that, in many cases there is little to choose between different step calculation methods.

The quadratic interpolation corresponds to a linear interpolation of the gradient and is carried out using the value of the gradient of the objective function at the two extremes of the search direction. The step is given by the point at which the interpolated gradient is zero. It can be calculated performing only one interpolation along the search direction or refining it in subintervals chosen to contain the zero of the gradient. The refinement can be carried out to ensure that an improved gradient is found or until the step is calculated to a given precision.

The calculation of the gradient follows the same procedure explained in the MUC SUE case in (5.22)-(5.26) with the only difference that the auxiliary solutions employed must now be intended as balanced auxiliary solutions.

5.4.5.4 MUC SUE ED Algorithms with Alternative Search Directions

The MUC BDA search direction used in the algorithms described above can be interpreted in conjunction with the MUC SUE ED objective function (5.75) and its gradient as in the SUE and MUC SUE cases. Thus it can be seen that the transformation into the space of the standardised flows of a steepest descent search direction in a space related to the flow space by a non-linear change of coordinates, the latter defined by the square root of the Jacobian of the common part of the link costs. In fact, the discussion parallels the one given in 4.3.5.2. with the difference that the preconditioning matrix \mathbf{J}_n , in the MUC SUE ED case is the Jacobian of the common part of the link costs, as it was in the MUC SUE case and as can be seen by comparing the respective expressions of the gradient of the objective function. Moreover, in the present case, the auxiliary solution must be intended as a balanced standardised auxiliary solution.

Continuing the parallel with the interpretation of the traditional search direction presented in chapter 4, it is possible to write algorithms working along preconditioned conjugate gradient directions. The final formulae can be written in the space of the flows and are identical to (4.55), (4.58), (4.59) and (4.60) except for the fact that, as mentioned in the previous paragraph, the preconditioner \mathbf{J}_n is the Jacobian of the common part of the link flows and the gradient of the objective function refers now to the MUC SUE ED function (5.75).

As in the SUE and the MUC SUE cases, although the coefficient combining the previous search direction with the current MUC BDA search direction must be calculated using quantities referred to the links, the search directions can be expressed either in terms of link flows or path flows (for a fixed link-path incidence matrix) and the algorithm can produce updates of the solution both in terms of path and link flows. Moreover, similar algorithms can be used for pure link-based models.

The algorithms tested in the following sections employ the preconditioned Polak-Ribiere and the preconditioned Fletcher-Reeves search directions. The extremes of the MUC BDA search direction define always a vector in the space of the feasible flows. The preconditioned gradient search vector might extend into the semispace of the negative flows, in which case it is shortened so that no flow is less than zero.

Also with the preconditioned conjugate search directions the tests are carried out only using the quadratic interpolation method described in the previous section with step calculated interpolating once, or until an improved gradient is found or to a given precision. The algorithms not implementing precise line search are used to try and save on computational effort although the approximate line search may introduce a further element of approximation (beside the changing preconditioner and the non-linearity of the function) to the conjugacy of the search direction. In the following sections the algorithms refining the line search to a given precision are referred to as ppFR and ppPR (precise preconditioned Fletcher-Reeves and precise preconditioned Polak-Ribiere, respectively), whilst the others are marked as apFR and apPR (approximate preconditioned Fletcher-Reeves and approximate preconditioned Polak-Ribiere, respectively).

The gradient along the search direction is calculated similarly to that described in the previous section. The gradient formula in the present case is (5.30) where the auxiliary solution employed must now be intended as a balanced auxiliary solution.

As a safety device, the dot product of the gradient and of the new direction is calculated at each iteration and, if it is not negative, the preconditioned conjugate gradient direction is discarded and restarted.

5.4.6 Performance of the Algorithms for the MUC SUE ED Problem

5.4.6.1 Introduction

This section is structured as in the MUC SUE case: it opens by introducing the test methodology and the test bed. Then the results of the numerical tests carried out by solving for MUC SUE with algorithms described above are presented separately for each test network used.

5.4.6.2 MUC SUE ED Algorithms Test Methodology

As for the MUC SUE case, the algorithms have been tested with a methodology similar to that employed in chapter 4.

Solving for MUC SUE ED on test networks, the number of loadings and iterations and the computation time necessary to reach a pre-defined level of convergence have been recorded.

The stochastic network loading (here intended as the balanced loading of all the user classes being assigned) has been taken as the elemental operation performed by the traffic assignment algorithms as it seems to reflect the largest part of the computational time involved in the computations, also when the different search directions considered here are applied. Moreover it allows us to avoid referring directly to the computational time that is machine specific.

The statistic used to measure the level of convergence is the same as the in the MUC SUE case (adapted from Maher and Hughes, 1997a):

$$\ln RMSnd = \ln \sqrt{\frac{1}{P} \sum_p \frac{(x_p - y_p)^2}{(0.5(x_p + y_p))^2}} \quad (5.83)$$

where x_p and y_p are respectively the current and the auxiliary standardised flow on path p at the iteration where the statistic is evaluated and P is the number of paths loaded (the paths receiving at least 0.1% of the standardised flow between the relevant OD are included in the statistic).

This convergence statistic in path flow terms captures the convergence also in terms of demand consistently with the balanced demand concept. In fact the auxiliary solution y_p is obtained by considering the demand in balance with the current costs therefore a measure of its distance from the current solution reflects also the convergence in terms of OD demand.

As the equilibrium is approached the statistic (5.83) tends to $-\infty$. In fact, in general, the current and the auxiliary flows will not coincide both because of the result of the assignment procedure and of the calculation of the demand consistent with the current cost. As the equilibrium is approached demand and network costs become consistent as well as path choices and network costs and the square root in (5.83), which is a non-dimensional measure of the distance between the auxiliary and the

current flows, tends to zero. The logarithm is considered to make the progress of the convergence trend clearer.

As in the MUC SUE case, separate statistics similar to (5.83) could have been considered.

The progress of the algorithms is monitored by plotting the evolution of the statistic against the number of multiple user class loadings. The overall convergence speed of an algorithm is assessed by evaluating the number of MUC stochastic loadings it requires to reach the solution defined by a target level of the RMSnd statistics that here is assumed as 10^{-4} corresponding to a value of $\ln\text{RMSnd}$ of -9.21 .

The tests have been carried out using the same two approximation methods used in chapter 4, the Mendell-Elston approximation (Mendell-Elston, 1974) with the optimised calculation order of Kamakura (1989) and the improved Clark approximation (Clark, 1961), that have been described in chapter 3.

5.4.6.3 MUC SUE ED Algorithms Test Bed

Also in the MUC SUE ED case the test bed has been obtained by modifying the grid network of Chen and Alfa (1991), the Sioux Falls network (used *e.g.* in Vythoukias, 1990) and the Headingley network (used *e.g.* in Maher *et al.*, 1999) to accommodate 2 and 3 user classes.

The modifications of the networks have been obtained as described for the MUC SUE case. Tables 5.4, 5.5 and 5.6 report the coefficients α and β (appearing in (5.1) and (5.2.)) and the ratio of the variance to the free flow cost of each link for the different user classes considered. The power function has been considered as the elastic demand function for all user classes and all OD. Tables 5.4, 5.5 and 5.6 also report the elasticities that apply to each class.

Once more it is recalled that the Headingley network contains connectors that are not accounted for in the formulation of the problem being used. Thus the application of the algorithms proposed to the Headingley network should be seen as heuristic.

C+A	UC1	UC2
P	0.85	0.15
α	1.00	1.50
β	0.80	1.20
$\sigma^2/\text{ff cost}$	0.50	0.40
El	0.70	0.70

C+A	UC1	UC2	UC3
P	0.80	0.10	0.10
α	1.00	1.00	2.00
β	0.80	1.20	1.00
$\sigma^2/\text{ff cost}$	0.50	0.40	0.50
El	0.70	0.80	0.80

Table 5.4 – Proportion P , coefficients α and β , elasticity and ratio of the variance to the free flow costs used in the cases of 2 and 3 user classes (UC) on the Chen and Alpha network.

Sioux F	UC1	UC2
P	0.25	0.75
α	1.20	0.90
β	0.70	1.25
$\sigma^2/\text{ff cost}$	0.50	0.40
El	0.70	0.70

Sioux F	UC1	UC2	UC3
P	0.75	0.10	0.15
α	0.90	1.50	0.90
β	0.70	1.20	0.70
$\sigma^2/\text{ff cost}$	0.50	0.50	0.10
El	0.70	0.80	0.70

Table 5.5 – Proportion P , coefficients α and β , elasticity and ratio of the variance to the free flow costs used in the cases of 2 and 3 user classes (UC) on the Sioux Falls network.

Hdngly	UC1	UC2
P	0.75	0.25
α	1.00	0.80
β	0.80	1.20
$\sigma^2/\text{ff cost}$	0.70	0.70
El	0.70	0.70

Hdngly	UC1	UC2	UC3
P	0.60	0.30	0.10
α	1.00	1.00	1.00
β	1.00	1.10	0.80
$\sigma^2/\text{ff cost}$	0.50	0.40	0.50
El	0.70	0.80	0.80

Table 5.6 – Proportion P , coefficients α and β , elasticity and ratio of the variance to the free flow costs used in the cases of 2 and 3 user classes (UC) on the Headingley network.

5.4.6.4 Test Results for the Chen and Alpha Network

The tests on the Chen and Alpha network, with 2 and 3 user classes show that, in general the convergence trends given by an algorithm with the Mendell-Elston and the Clark approximations are similar. This is the case also on the other networks discussed in the following sections, thus the comments reported refer to the results with both approximations and a selection of figures referring to either approximation has been included.

The results obtained show, once more, that the MSA does well at the beginning of the calculations but, on the whole it is outperformed by the algorithm using the MUC BDA search direction and the quadratic interpolation, as in the examples reported in figs. 5.24 and 5.25. Here the algorithms performing the line search once and those checking for the improvement of the gradient have convergence trends that coincide whilst those refining the step to a given precision are less efficient.

Algorithms started with 5, 10, 15 and 20 MSA iterations before using an optimised step length have been tested and in all cases they improve on the algorithm using the line search from the outset as well as on the MSA. Examples of this behaviour are reported again in figs. 5.24 and 5.25.

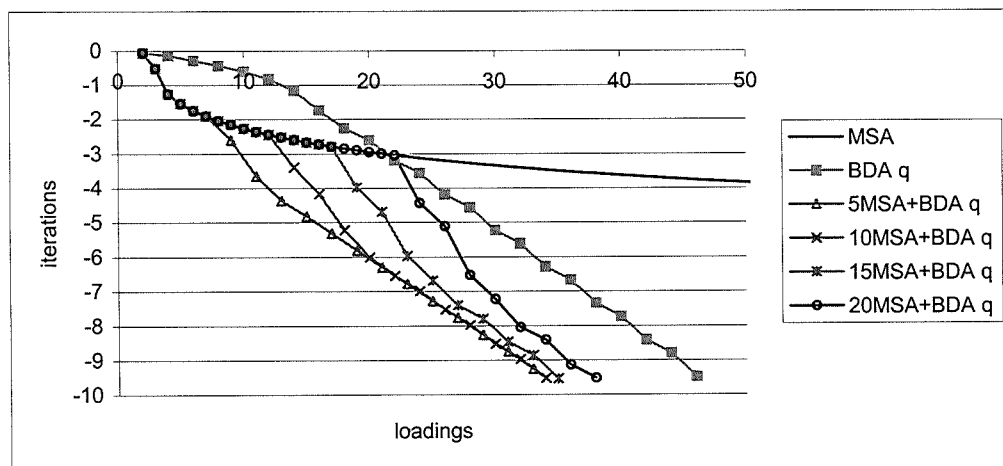


Fig. 5.24 – Comparison of the MSA, the algorithm using the BDA search direction and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Clark approximation, 2 user classes.

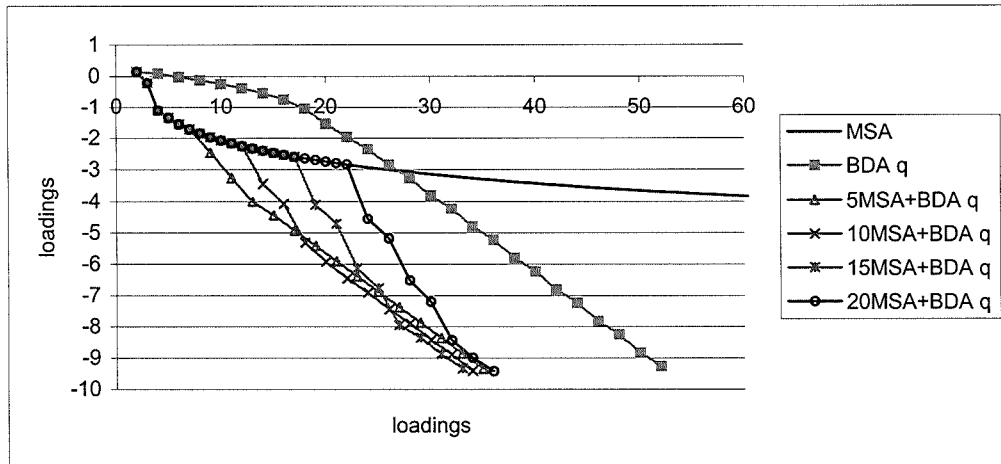


Fig. 5.25 – Comparison of the MSA, the algorithm using the BDA search direction and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Mendell-Elston approximation, 3 user classes.

The preconditioned Fletcher-Reeves search direction used with line search refined to a given precision (ppFR) show convergence trends that outperform from the start of the calculations those of the corresponding algorithms using the traditional search direction, as in the example in fig. 5.26. In fact, the algorithms with step refined to 10^{-2} do as well as those using the traditional search direction with unrefined step.

Using the Fletcher-Reeves search direction coupled with step calculations not refined to a given precision (apFR) gives, also in this case, instabilities of the convergence trends. This behaviour has not been investigated further in the present work but it is interesting to note that this problem is absent when a precise line search is used and when the apFR search direction is not used from the outset but the algorithm is started with a number of MSA iterations. It seems reasonable to infer that the problem with the apFR arises from the information inherited from the first iterations, during which using unrefined line search does not seem to be robust. The instabilities could possibly be obviated by restarting the algorithm when they are detected, though the algorithm with this added safeguard has not been tested here.

Hybrid apFR algorithms started with 5, 10, 15 and 20 MSA iterations, outperform in all cases and from the point the apFR search direction is used the corresponding algorithms using the traditional search direction. Figures reporting the convergence trends are not included as they practically coincide with those of the hybrid apPR algorithms reported below.

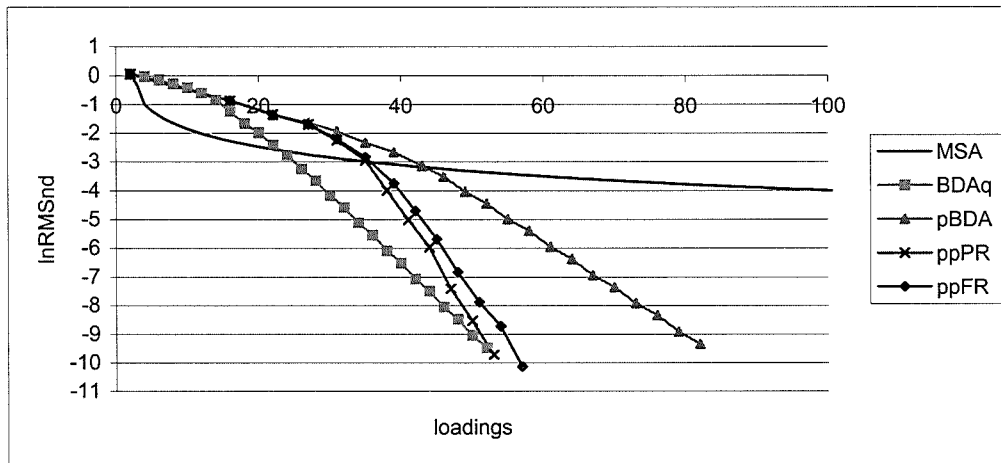


Fig. 5.26 – Comparison of the MSA, the algorithm working along the BDA search direction and returning the step when an improved gradient has been found (BDA q) and algorithms refining the step to a precision of 10^{-3} along the BDA direction (pBDA), the preconditioned Polak-Ribiere search direction (ppPR) and the preconditioned Fletcher-Reeves search direction (ppFR). Chen and Alpha network, Mendell-Elston approximation, 2 user classes.

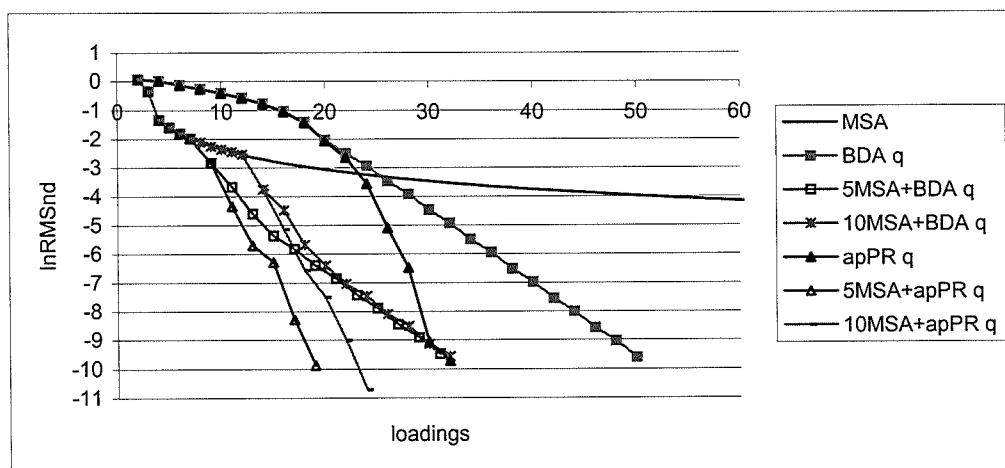


Fig. 5.27 – Comparison of the MSA, the algorithm using the BDA search direction (BDA q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Chen and Alpha network, Clark approximation, 3 user classes.

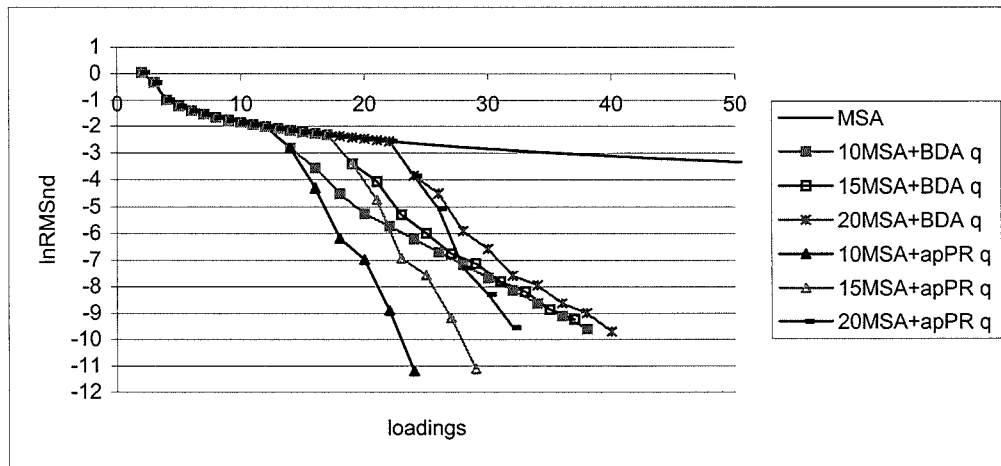


Fig. 5.28 – Comparison of the MSA, and hybrid algorithms using the BDA search direction (BDA q) and the approximate preconditioned Polak-Ribiere (apPR) search direction started with the number of MSA iterations indicated. Chen and Alpha network, Mendell-Elston approximation, 2 user classes.

Algorithms refining the step to a given precision along the preconditioned Polak-Ribiere search direction (ppPR algorithms) are more efficient than the corresponding algorithms using the traditional search direction and marginally more efficient than the corresponding ppFR ones (see for instance the convergence plot in fig. 5.26).

Determining the step length so that an improved gradient is found (algorithms apPR) is more efficient than calculating it to a given precision and the resulting convergence trends do not show the oscillations given by the corresponding apFR algorithm. This algorithm improves also on the one using the BDA search direction from the outset of the calculations, as can be seen from figs. 5.27 and 5.28 which show also that hybrid apPR algorithms improve on the apPR algorithm (although if they are started with 20 MSA iterations in this test case there is little to choose) and on the corresponding algorithms using the BDA search direction. Although not shown in the figures, it is interesting to note that the hybrid algorithms using the apFR and the apPR search direction give practically coincident convergence trends.

5.4.6.5 Test Results for the Sioux Falls Network

Testing the algorithms on the Sioux Falls network showed again a good initial behaviour of the MSA using the MUC BDA search direction. As well as in other cases using optimised steps gives a better overall efficiency and working along the MUC BDA search direction using the MSA step for a few iterations gives an even improved convergence trend. This is shown in the two examples reported in fig. 5.29. The algorithms using the MUC BDA search direction and refining the step to a given precision along it are less efficient than those working with the step obtained with one interpolation or checking that an improved gradient is found, and of the hybrid algorithms. The two types of unrefined line search give the same convergence trends.

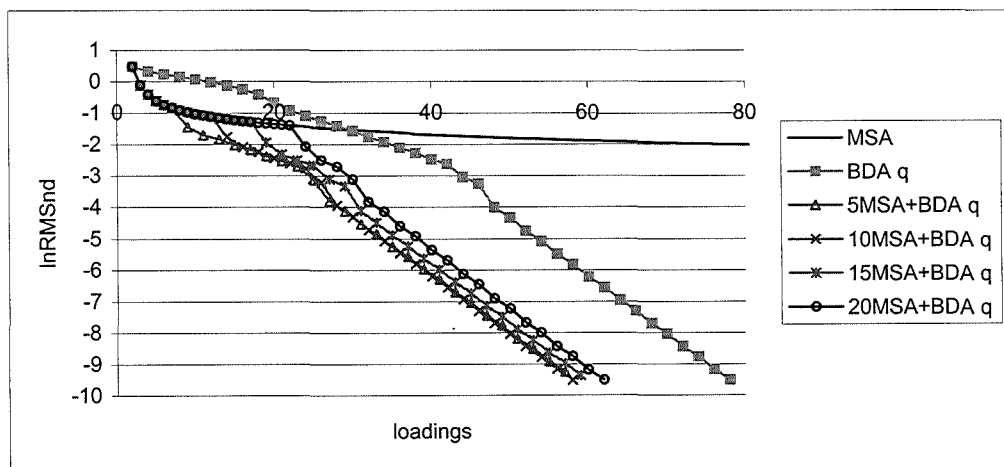


Fig. 5.29 – Comparison of the MSA, the algorithm using the BDA search direction and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network, path set S2, Mendell-Elston approximation, 2 user classes.

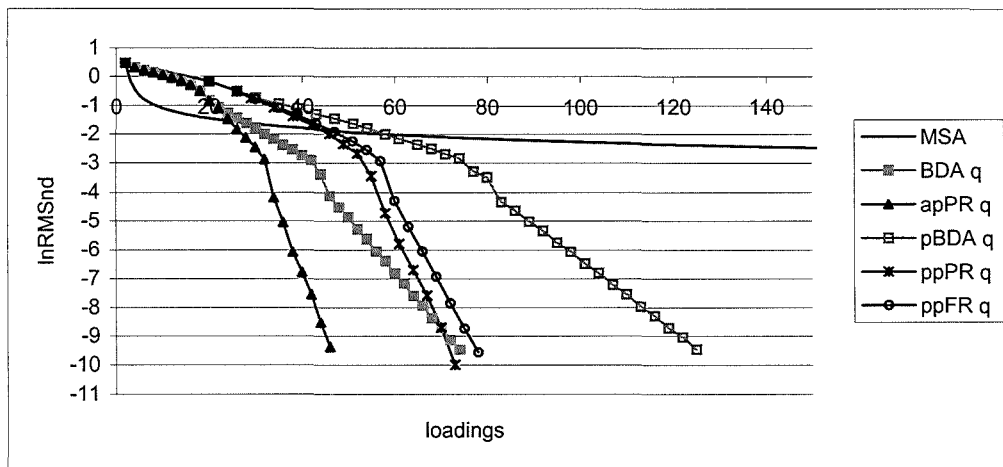


Fig. 5.30 – Comparison of the MSA, the algorithm working along the BDA search direction and returning the step when an improved gradient has been found (BDA q) and algorithms refining the step to a precision of 10^{-3} along the BDA direction (pBDA), the preconditioned Polak-Ribiere search direction (ppPR) and the preconditioned Fletcher-Reeves search direction (ppFR). Sioux Falls network, path set S2, Clark approximation, 2 user classes.

Also the ppFR algorithms, using the pFR search direction and refining the step to a given precision, are more efficient than the corresponding ones working along the BDA search direction (see the example in fig. 5.30). Not refining the step, that is using an apFR algorithm, also in this case causes instabilities of the convergence trends that are not present when the apFR search direction is used as part of hybrid algorithms started with a number of MSA iterations. Algorithms starting with 5, 10, 15 and 20 MSA iteration have been tested and, in all cases, they improve on the corresponding algorithms using the BDA search direction. Results for this sort of hybrid algorithms are not reported as their trends are very similar to those obtained with hybrid apPR algorithms, that have been depicted in figures reported below.

Using the preconditioned Polak-Ribiere search direction along with precise line search (algorithm ppPR) is as efficient as using the corresponding algorithms working along the ppFR search direction (there are slight improvements but there is little to choose between the two). Limiting the step search to a point where an improved gradient is found, that is using an apPR algorithm, gives in most cases even more efficient convergence trends, and certainly more efficient than using the MUC BDA search direction, although in a few cases ppPR algorithms with step

refined only to 10^{-2} are as efficient overall and more efficient at the beginning of the calculations. The hybrid algorithms obtained starting with a number of MSA iterations improve in all cases on the pure apPR algorithm and on the corresponding hybrid methods employing the BDA search direction, as shown in figs. 5.31 and 5.32. As mentioned above there is little to choose between hybrid apFR and apPR algorithms in the test cases examined, since in most cases they practically coincide.

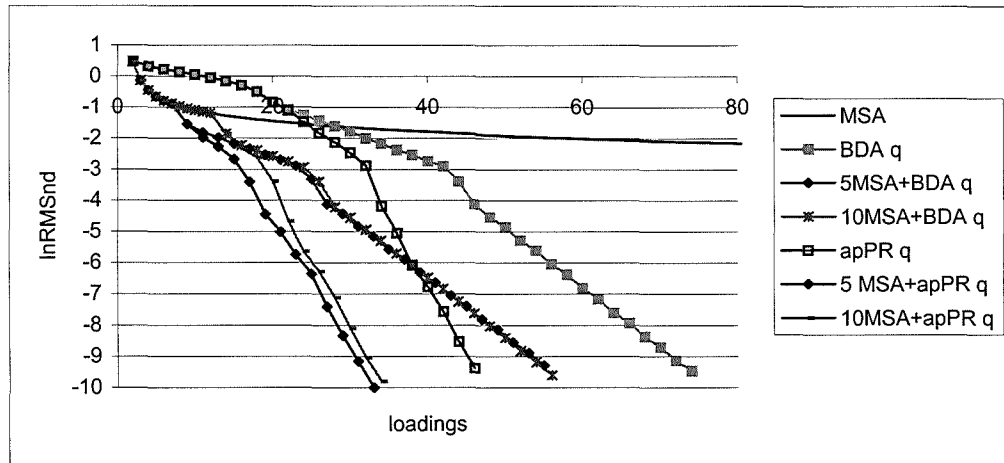


Fig. 5.31 – Comparison of the MSA, the algorithm using the BDA search direction (BDA q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network. Path set S2, Clark approximation, 2 user classes.

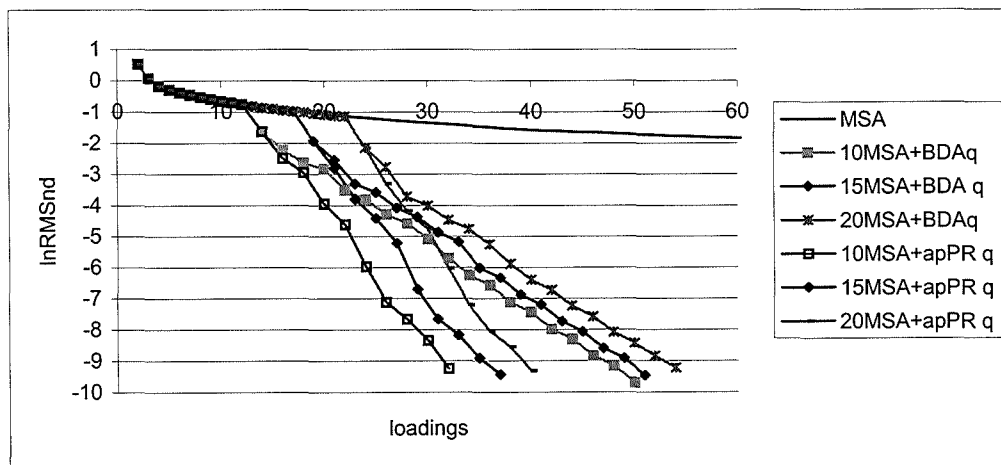


Fig. 5.32 – Comparison of the MSA and hybrid algorithms using the BDA search direction (BDA q) and the approximate preconditioned Polak-Ribiere (apPR) search direction started with the number of MSA iterations indicated. Sioux Falls network, path set S1, Mendell-Elston approximation, 3 user classes.

5.4.6.6 Test Results for the Headingley Network

Finally, the tests on the Headingley network suggest results similar to those obtained above in other cases.

Once more, the MUC BDA MSA does well at the beginning of the optimisation but using an optimised step along the same search direction takes the algorithm to the required level of convergence much faster. Combining the initial good performance of the MSA and the later better behaviour given by calculating the steps results in hybrid algorithms that in the cases investigated (with 5, 10, 15 and 20 MSA iterations to start with) improve always on the MSA and in some cases improve on using the interpolation throughout, whilst in other cases they do so at the beginning of the calculations but there is little to choose between the hybrid algorithms and the one using interpolation throughout, as can be seen comparing figs. 5.33 and 5.34. This discussion refers to cases where the interpolation of the gradient is carried out only to obtain a step ensuring a better value of the gradient than at the present solution, which, in the cases examined, coincide with carrying out a single interpolation. Using interpolation refined to give precision results in algorithms that are less efficient from the start.

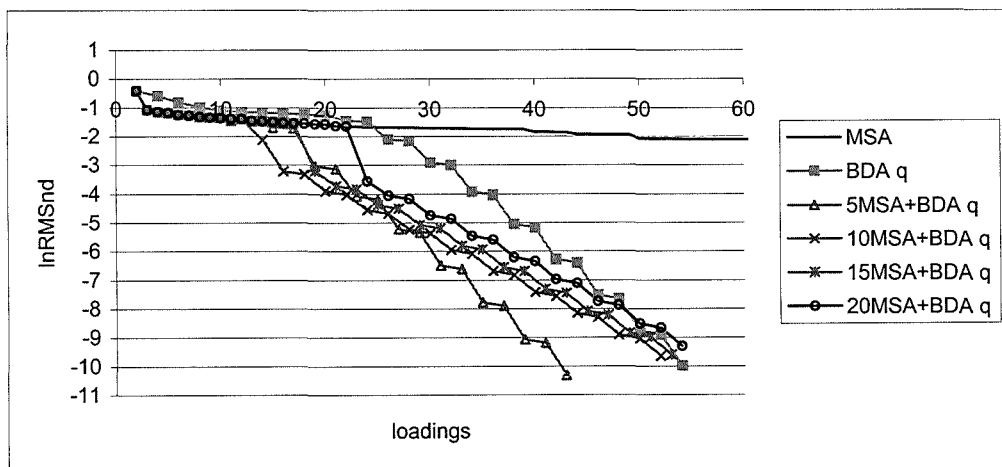


Fig. 5.33 – Comparison of the MSA, the algorithm using the BDA search direction and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, path set H1, Clark approximation, 3 user classes.

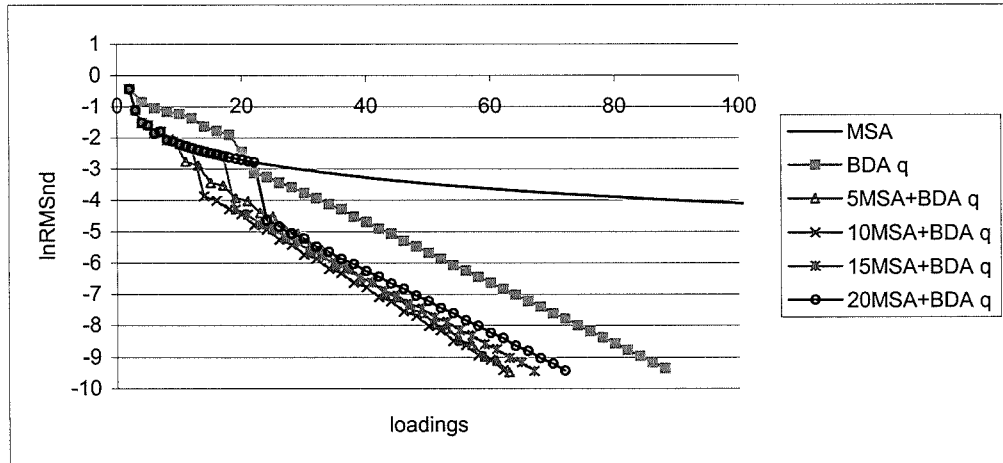


Fig. 5.34 – Comparison of the MSA, the algorithm using the BDA search direction and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, path set H2, Mendell-Elston approximation, 3 user classes.

The improvement due to using a ppFR algorithm, that is using the preconditioned Fletcher-Reeves search direction with refined line search, against using the same method with the MUC BDA search direction is shown in fig. 5.35. Not refining the step also in this case results in instabilities of the algorithm, that, again as in other cases, are avoided if the algorithm is part of a hybrid started with some MSA iterations. The latter sort of algorithms, tested starting with 5, 10, 15 and 20 MSA iterations as in other cases, improve noticeably on the corresponding algorithms using the MUC BDA search direction, from when the apFR search direction is employed. Results for these algorithms are not reported, as they are very similar to those obtained with the corresponding hybrid apPR reported below.

Working along the preconditioned Polak-Ribiere search direction and determining the step to a given precision improves further on the corresponding algorithms using the ppFR direction, as shown again in fig. 5.35. However, the most efficient strategy, as shown in the tests, is to avoid performing further interpolations in subintervals unless an improved gradient has not been found. The algorithm implementing this line search strategy does not show instabilities, probably thanks to its ability to self restart, and starting it with a number of MSA iterations generally gives an improvement of the algorithm efficiency from the beginning of the calculations, although in most cases on this test case there is little to choose between the apPR algorithm and an hybrid apPR algorithm started with 20 MSA iterations. The

noticeable improvement over the corresponding algorithms using the MUC BDA search direction is shown in figs. 5.36 and 5.37.

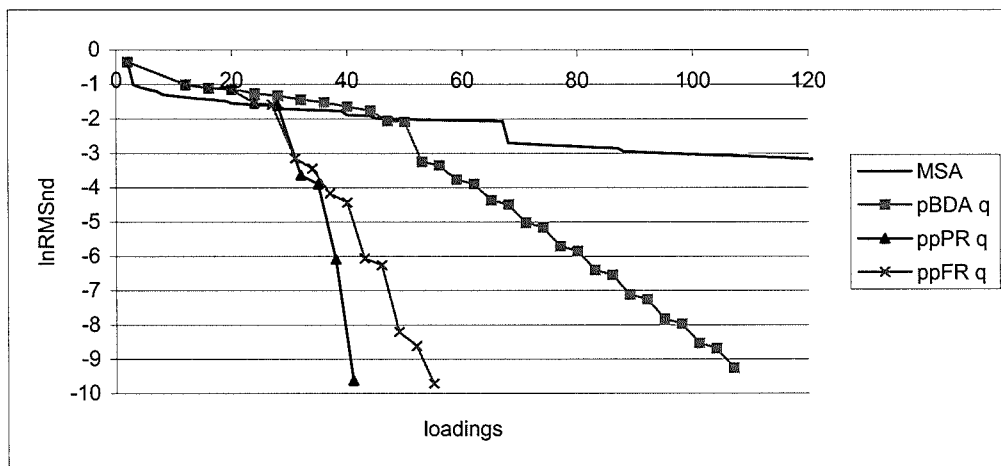


Fig. 5.35 – Comparison of the MSA and algorithms refining the step to a precision of 10^{-3} along the BDA direction (pBDA q), the preconditioned Polak-Ribiere search direction (ppPR q) and the preconditioned Fletcher-Reeves search direction (ppFR q). Headingley network, path set H1, Mendell-Elston approximation, 3 user classes.

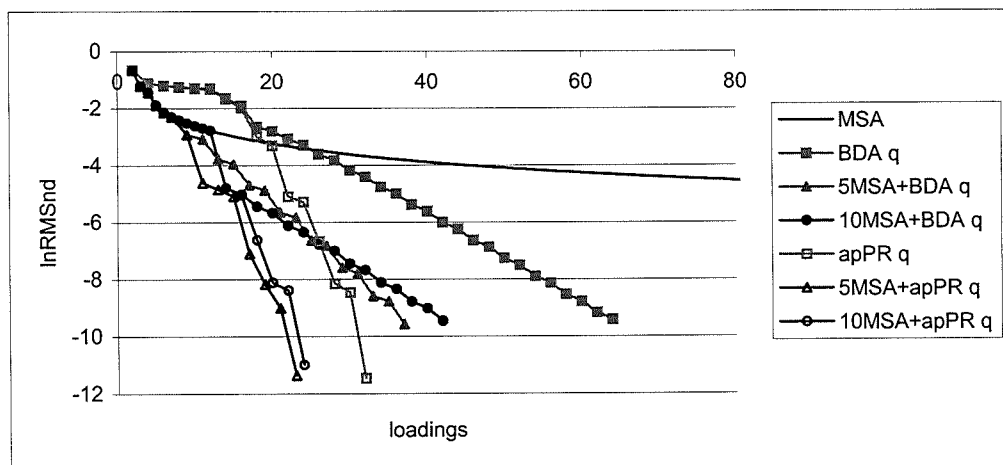


Fig. 5.36 – Comparison of the MSA, the algorithm using the BDA search direction (BDA q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Headingley network, path set H2, Clark approximation, 2 user classes.

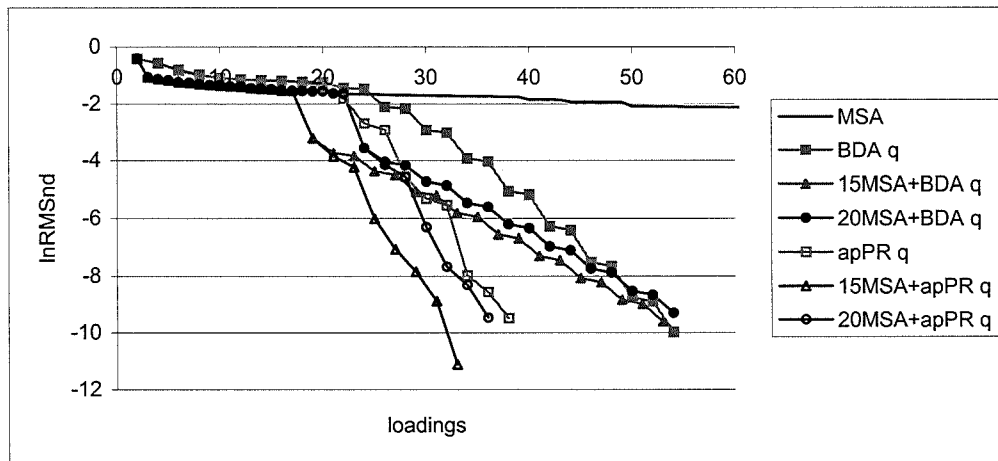


Fig. 5.37 – Comparison of the MSA, the algorithm using the BDA search direction (BDA q), the algorithm using the approximate preconditioned Polak-Ribiere (apPR) search direction and hybrid algorithms started with the number of MSA iterations indicated. Sioux Falls network, path set H1, Clark approximation, 3 user classes.

5.4.7 Conclusions

The MUC SUE ED model studied in this section provides a general formulation of which the SUE, MUC SUE or SUE ED can be seen as particular cases.

The work presented has built on the multiple user class framework of Daganzo (1982) and on the work on SUE ED discussed in the previous section. An extension of the SUE ED objective function of Maher *et al.* (1999) to the MUC SUE ED case has been proposed and an alternative equivalent programme, that reduces the problem to one in terms of either standardised link costs or link flows, has been put forward.

The uniqueness of the solution of the new programme has been discussed and algorithms based on the objective function in terms of standardised link flows have been proposed. These included an MSA working along a Balanced Demand search direction expressed in terms of standardised flows and an algorithm performing line searches along the same search direction by interpolating the objective function as quadratic. The MSA search direction has also been used as a starting point to

propose preconditioned conjugate gradient algorithms based on the conjugate gradient formulae of Fletcher and Reeves and of Polak and Ribiere.

The comparisons of the performance of the algorithms on three test networks gave results that resembled those for the corresponding algorithms in the SUE and MUC SUE cases.

The MSA was overall uncompetitive but was more efficient than other methods at the beginning of the calculations and was included in hybrid algorithms that start with a number of MSA iterations before completing the solution of the problem with other methods.

Algorithms working along different search directions and refining the interpolations to determine the step length to a given precision were less efficient than those interpolating only until an improved gradient was found, which, in most cases, meant carrying out a single interpolation. The algorithms using the BDA search direction and finding the step by interpolating until an improved gradient was found were outperformed by the corresponding algorithms working along the preconditioned Polak-Ribiere search direction, whilst those using the preconditioned Fletcher-Reeves search direction showed some instabilities.

Hybrid algorithms, started with a number of MSA iterations, gave the best overall performances and also in this case the algorithms working along the BDA search direction were outperformed by those using the preconditioned Polak-Ribiere one, which gave results similar to those using the preconditioned Fletcher-Reeves search direction.

5.5 Conclusions

This chapter has discussed the extension of Stochastic User Equilibrium models to include Multiple User Classes (MUC) and Elastic Demand (ED), both separately and together.

The discussion on MUC SUE models started by reviewing the relevant work in the literature. The rest of the study has been based on an objective function suggested by Daganzo (1982) that has been detailed also to solve the problem in terms of standardised link costs. The equivalence of its solution to the MUC SUE conditions and the uniqueness of the solution have also been discussed before proposing a number of algorithms to solve the problem.

The algorithms proposed are extensions of those seen for the SUE case in chapter 4 and include an MSA algorithm and algorithms working along the search direction linking the current and auxiliary solution for the MUC SUE problem. This search direction has then been interpreted as a preconditioned steepest descent search direction, extending the analogous discussion for the SUE case and preconditioned conjugate gradient algorithms based on the formulae of Fletcher and Reeves and Polak-Ribiere have been proposed.

Numerical tests on three networks have shown that the preconditioned Polak-Ribiere search direction outperforms the extension of the traditional search direction to the MUC SUE problem. The preconditioned Fletcher-Reeves search direction gave, when applied without refining the step, instabilities that should be further investigated but question its robustness in this application. The most efficient solution strategy, as in the SUE case, resulted from the use of hybrid algorithms started with a number of MSA iterations before the preconditioned Polak-Ribiere search direction with line search refined only until an improved gradient is found is used. Algorithms with either the Mendell-Elston or the Clark approximation applied to the same test case showed a similar behaviour.

The section on Stochastic User Equilibrium with Elastic Demand (SUE ED) models has been introduced by a review of the relevant literature and of the motivations for including elastic demand in assignment models. The SUE ED problem has been discussed proposing a new equivalent programme that reduces the problem to one in terms of link costs or of link flows. The equivalence of its solution to the SUE ED conditions as well as its uniqueness have been shown. Moreover, the relationship between the new programme and the programme put forward by Maher *et al.* (1999), has been given.

A number of algorithms based on using the Balanced Demand search direction and on using preconditioned conjugate gradient algorithms have been suggested without testing them numerically.

The last section of the chapter has been concerned with solving the SUE assignment problem when both Multiple User Classes and Elastic Demand are included. An extension of the SUE ED objective function of Maher *et al.* (1999) to the SUE ED problem has been suggested but the following discussion has been based on a new programme for MUC SUE ED that has been put forward, which reduces the problem to one in terms of link flows or costs.

After analysing the equivalence and uniqueness of the new programme in either of its versions, the expression in terms of standardised link flows has been used to propose a number of solution algorithms that are extensions of those for MUC SUE and for SUE ED. An MSA algorithm, algorithms working along the Balanced Demand search direction for the MUC problem and preconditioned conjugate gradient algorithms based on the Fletcher-Reeves and Polak-Ribiere formulae have been used. The latter type of algorithm has been derived by interpreting the BDA search direction as a preconditioned steepest descent direction for the MUC SUE ED programme in terms of link flows.

The results of the numerical tests mirrored those of the corresponding tests for SUE and MUC SUE: the preconditioned Polak-Ribiere search direction outperforms the extension of the traditional search direction to the MUC SUE ED problem whilst the

preconditioned Fletcher-Reeves search direction, unless applied with precise calculation of the step or within a hybrid algorithm, gave instabilities. The use of hybrid algorithms, starting with a number of MSA iterations before the preconditioned Polak-Ribiere search direction with unrefined line search is employed, appears to be the most efficient solution method. Also in this case, using either the Mendell-Elston or the Clark approximation within the same algorithm on the same test case gave similar behaviours.

6. CALIBRATION OF MULTINOMIAL PROBIT CHOICE MODELS

6.1 Introduction

The present chapter deals with the application of the multinomial probit (MNP) choice model in the context of simple choices, in the sense defined in chapter 2, that is when the attributes of the options are fixed and do not change as an effect of the choices made. In particular, this chapter is concerned with issues related to the calibration of the MNP model, when an analytical approximation is used to solve the choice function.

The calibration of choice models has been introduced in chapter 2. At the calibration stage the choice model has been specified (or a number of alternative possible specifications have been defined) and data about actual choices are available from surveys along with those data about the characteristics of the decision makers and of the options that will be possibly used to define the latent utilities. The calibration of a choice model consists of estimating the parameters appearing in the model using such surveyed data and evaluating the statistical significance of the estimates. The latter type of information can also help the analyst to choose amongst different specifications of the choice models as, for instance, which attributes of the choice maker and of the alternatives are statistically relevant to model the choice process.

Discrete choice models are most often calibrated using the maximum likelihood method (see *e.g.* Daganzo, 1979; Ben-Akiva and Lerman, 1985; Ortúzar and Willumsen, 1994), which is also used in the present work and produces the parameters with which the model best replicates the sample of choice data. This method in fact, consists of calculating the parameters that maximise the relevant likelihood function. The latter describes the probability distribution of the calibration data as a function of the parameters of the model, thus maximising it yields the parameters using which the model is most likely to return the calibration choice data. The maximum likelihood estimates of the parameters are consistent, asymptotically Normal and asymptotically efficient (see *e.g.* Ben-Akiva and Lerman, 1985).

The actual expression of the likelihood function depends on how the data have been obtained (see *e.g.* Ben-Akiva and Lerman, 1985). In the case of random sampling, that is the only one considered in this chapter, the likelihood function results:

$$L(\boldsymbol{\theta}) = \prod_{s=1}^S P(c_{(s)} | \mathbf{a}_{(s)}, \boldsymbol{\theta}) \quad (6.1)$$

where S is the sample size, that is the number of decision makers whose choices and attributes have been surveyed, and $P(\cdot)$ is the probability that the option $c_{(s)}$ actually chosen by the decision maker s results in being chosen in the model as a function of the attributes $\mathbf{a}_{(s)}$ and of the parameters $\boldsymbol{\theta}$.

In practice the calculations are usually carried out by maximising the logarithm of the likelihood function which has the same maximum, since the logarithm is a strictly monotonic function, but is easier to work with. The log-likelihood function resulting from (6.1) is:

$$\log L(\boldsymbol{\theta}) = \sum_{s=1}^S \log P(c_{(s)} | \mathbf{a}_{(s)}, \boldsymbol{\theta}) \quad (6.2)$$

(6.2) can be maximised using *e.g.* quasi-Newton algorithms that employ the value of the function and of its gradient. With a choice function that cannot be written in closed form, as in the MNP case, the gradient can be calculated numerically. In fact, only in two cases in the literature, one using simulation and the other employing numerical integration, is the gradient of the MNP log-likelihood function obtained analytically (Bolduc, 1999; Sheffi *et al.*, 1982).

When using the probit model solved with approximation methods it is interesting to develop a practical technique to calculate the analytical derivatives of the log-likelihood function with respect to the coefficients to calibrate exploiting the formulation of the approximations. In fact, a procedure maximising the logarithm of the likelihood using analytical derivatives should be expected to be more efficient than one using numerical ones.

In this chapter, a method for obtaining analytical derivatives with the Mendell-Elston approximation is presented and employed to develop a MNP calibration computer program. The program is also tested on some examples.

The rest of the chapter is structured as follows. Section 6.2 reviews the issues related to correctly specifying the MNP model, referring to the limited literature on the subject. Section 6.3 gives an account of methods for MNP calibration that have appeared in the open literature. Then, section 6.4 describes the calibration program developed for the present work and discusses in detail the algorithm used to obtain analytical derivatives with the MNP approximation of Mendell-Elston, mentioning how it may readily be expanded for use with other approximations. Finally, section 6.5 presents results from numerical tests of the algorithms exploring the ability of the model to retrieve accurately the parameters of some known specifications of the choice problem and comparing the time required to solve the calibration problem by algorithms using different ways to obtain the derivatives. The work presented in the chapter is summarised and discussed in section 6.6 that also suggests possible further work on the subject.

6.2 Specification of the Multinomial Probit Model

The specification of choice models, that is of their structure and of the parameters that can actually be estimated, is subject to restrictions that exist because the models are usually considered in terms of utilities but the data to calibrate them, attributes and choice for each surveyed decision maker, give information about a process where the differences of the utilities rather than the utilities themselves determine the outcome (see *e.g.* Bunch, 1991). The utilities, in fact, according to random utility theory, are only used to rank the alternatives. If a constant is added to all utilities, or they are multiplied by a positive constant, their ranking is unaffected, as is the resulting choice. Consequently only the parameters that can be retrieved uniquely from the choice model written in difference with respect to a reference alternative can be estimated (see *e.g.* Bunch, 1991; Bolduc, 1992).

In multinomial logit models this means that alternative specific constants or socio-economic “dummies” may be used in the specification of the systematic utility of all options except one, to fix the level of the utilities of the model. The multinomial logit covariance matrix does not introduce specification issues, as it is fixed and diagonal. In fact its scale, which is its only possible parameter, cannot be distinguished from the parameters in the expression of the systematic utilities (see e.g. Ben-Akiva and Lerman, 1985).

In multinomial probit models the same limitations mentioned for logit models apply to the systematic utility parameters. Moreover, Keane (1992) warned against the “fragile identification” brought about by not excluding from at least one alternative’s utility each of the attributes that have the same value for all alternatives (typically the choice maker’s attributes). This exclusion is not required to obtain formal identification but avoids the problems in retrieving the parameters due to the systematic utility coefficients having effects similar to those of the covariance matrix elements.

The covariance matrix of a multinomial probit model needs particularly careful specification. Detailing the general principle reported above, due to the fact that the outcome of the choice process is determined by the differences of the utilities, only the parameters that can be retrieved uniquely from the covariance matrix of the utilities written in difference with respect to the utility of a reference alternative can be estimated. In general this means that, for n choice alternatives, the covariance matrix of the utilities will have $n(n+1)/2$ entries but the covariance matrix in difference with respect to a reference alternative (referred to, in the following, as the covariance matrix in difference) will have $n(n-1)/2$ parameters and therefore this is the maximum number of parameters that can be specified in the covariance matrix. This number actually reduces to $[n(n-1)/2]-1$ as the scale of the model must also be defined and this can be achieved by fixing an entry of the matrix in difference to a constant. This scaling issue is similar to that of the logit model for which the parameter related to the variance of the Gumbel variates is not distinguishable from the average utility parameters.

One way to deal with this covariance matrix specification issue, which is also the method used in this chapter, is to estimate the covariance matrix in difference w.r.t. a reference alternative, as it contains only the parameters that are actually estimable, after fixing the scale of the model by assuming that one of its diagonal entries is equal to a constant.

Alternatively, the original covariance matrix can be estimated but care should be taken to make sure that the free parameters appearing in it are actually estimable.

Working with the covariance matrix in difference is the method most suggested and commonly used in the literature (see *e.g.* Dansie, 1985; Bunch, 1991; Weeks, 1997; Train, 2002). In particular, Dansie (1985) and Bunch (1991) suggested this method mentioning also that it is in accordance with the original development of choice models, that were initially referred to utility differences, rather than to the more intuitive utilities.

The results presented later in this chapter have been obtained using the first alternative as reference alternative and fixing its first diagonal entry to a constant. The same method is suggested also by Train (2002) but, clearly, other equivalent assumptions are possible. For instance, Bolduc (1999) fixed the first diagonal entry of the covariance matrix in difference w.r.t. the last alternative. Alternatively the scale can be determined by fixing an attribute coefficient appearing across all the systematic utilities, but this is not the general practice.

Once the choice model in difference has been calibrated, obtaining the covariance matrix of the utilities presents the same problems that arise when one defines a covariance matrix of the utilities and uses it directly in the calibration process: only some parameters can be correctly assumed to be estimable, and understanding which is not straightforward, whilst the other entries must be set to zero thus making assumptions about the structure of the problem at hand. This was seen by Bunch (1991) as a restriction of the flexibility of the probit model, to some extent as in Nested Logit models, where the nesting structure is predetermined.

The issues arising in the definition of the covariance matrix can be illustrated by considering the example used by Dansie (1985) and also by Bunch (1991) to show that different covariance matrices of utilities may correspond to the same covariance matrix in difference. Dansie showed that the matrices:

$$\Sigma_A = \begin{bmatrix} 1 & \sigma_{12}^2 & 0 \\ \sigma_{12}^2 & 1 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{bmatrix} \quad (6.3a)$$

$$\Sigma_B = \begin{bmatrix} 1 & \sigma_{12}^2 & 0 \\ \sigma_{12}^2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3b)$$

$$\Sigma_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{bmatrix} \quad (6.3c)$$

are equivalent, although the first one is not identifiable (his paper remarked that Σ_A , used in numerical experiments on the accuracy of the approximation of Clark by Horowitz *et al.*, 1982, was not identifiable). In fact, calculating the covariance matrix in difference w.r.t. a reference option for each of the (6.3) it can be seen that they all correspond to the same covariance matrix in difference \mathbf{T} :

$$\mathbf{T} = \begin{bmatrix} 1 & \sigma^2 \\ \sigma^2 & 1 \end{bmatrix} \quad (6.4)$$

which Dansie suggested to use in model calibration as it includes the only free parameter of this model and is uniquely identifiable.

The example of Dansie shows both issues mentioned above: the need to make assumptions about the actual structure of the problem and the importance to check the correctness of the structure chosen.

In fact, the same matrix in difference (6.4) corresponds to a matrix with homoscedastic errors and correlation between two options and to a matrix with no correlation amongst alternatives but one option's variance different from the other two (excluding the unidentified matrix Σ_A). The choice between the two

specifications will not have an effect of the outcome of the calculated choices, so the flexibility of the MNP is not actually affected, but the interpretation of the choice situation is subject to assumptions made by the analyst.

The fact that Σ_A in (6.3a) is non identifiable although it contains only two free parameters, that is the maximum number of parameters that can be specified in a trinomial probit covariance matrix, underlines the need to check the specification of the covariance matrix of the utilities. A number of ways to carry out this operation has been suggested in the literature. Bunch (1991) gave a test method based on checking the invertibility of the composite transformation mapping the vector of possible parameters to estimate in the covariance matrix to the vector of unique entries in the covariance matrix in difference w.r.t. a reference alternative via the vector of unique entries of the covariance matrix of the utilities. If the Jacobian of this composite transformation has full rank, the transformation is invertible and the parameters assumed for the covariance matrix of the utilities are actually estimable.

Bolduc (1992) put forward a further rank condition, which includes that of Bunch, based on obtaining the transformation matrix that maps the vector of possible parameters to estimate in the covariance matrix to the vector of unique entries of a reference covariance matrix in difference. If the matrix that defines the transformation has rank equal to the number of proposed parameters they are actually estimable.

Equivalently, and more simply, Train (2002) suggested to check explicitly that each parameter in the covariance matrix of the utilities can be uniquely retrieved from those in the covariance matrix in difference by writing their relationships, as spelt out in the general rule. However, when several choice options are included in the model, the number of covariance parameters may become very large and the latter method becomes cumbersome to use.

Independently of any identification problem, the increase in the dimension of the choice set can bring about problems with calibrating the covariance parameters, that

are the most difficult ones to obtain (see *e.g.* Sheffi *et al.*, 1982), and problems with their interpretation.

The latter issue was remarked upon by Horowitz (1991) who was particularly concerned that the potentially large number of estimable elements in a general covariance matrix would bring about problems in testing a model's specification (due to free covariance matrices possibly making up for errors in the systematic utility specification, thus hiding them) and would give models containing high uncertainties due to the difficulty of estimating the covariance matrix entries with the sample sizes commonly used in practice and to the difficulty of assigning the covariance entries for new choice options.

To overcome this type of objections and limit the actual number of parameters used to define the covariance matrix, the use of structured matrices has been recently introduced.

Ben-Akiva and Bierlaire (1999) mention the possibility of using a factor analytic formulation that defines the vector \mathbf{U} of n utilities as

$$\mathbf{U} = \mathbf{V} + \boldsymbol{\varepsilon} = \mathbf{V} + \mathbf{F}\boldsymbol{\xi} \quad (6.5)$$

where $\boldsymbol{\xi}$ is a vector of standard Normal variables of dimension m and the $[n \times m]$ matrix \mathbf{F} maps the random factors to the n utilities. A particular instance of this formulation is the case in which $\boldsymbol{\xi}$ has dimension n and \mathbf{F} is the Cholesky factorisation of the covariance matrix. Another important instance of the factor analytic structure (6.5) is the one resulting in a first order autoregressive error, proposed for use with the MNP by Bolduc (1992). In this case the utilities can be written as

$$\mathbf{U} = \mathbf{V} + \boldsymbol{\varepsilon} \text{ with } \boldsymbol{\varepsilon} = \rho \mathbf{W}\boldsymbol{\varepsilon} + \mathbf{T}\boldsymbol{\xi} = (\mathbf{I} - \rho \mathbf{W})^{-1} \mathbf{T}\boldsymbol{\xi} \quad (6.6)$$

where \mathbf{W} is a matrix of weights that determines the influence of each error component on the others, ρ is a parameter (restricted to $]-1;1[$) and \mathbf{T} is a diagonal matrix of standard deviations applied to the standard Normal vector $\boldsymbol{\xi}$. Bolduc suggested that the matrix \mathbf{W} , describing the correlation among the alternatives and

that could be seen as a matrix of measures of contiguity of the alternatives, can be built up of estimable parametric functions but can also reduce to a simple Boolean matrix.

Bolduc (1999) used this formulation for a choice situation with a total number of 9 alternatives which, potentially, implied the need to estimate 35 parameters in a totally free covariance matrix. Using a first order autoregressive structure, the choice problem was described with a maximum of 9 free parameters (8 of which were standard deviations) employing a contiguity matrix \mathbf{W} defined a priori as a Boolean matrix written considering the characteristics of the problem.

An alternative method of structuring the covariance matrix is to use a network description of the problem. This technique has been employed by Yai *et al.* (1997) and Yai and Shimizu (1998): they analysed a route choice problem and exploited its natural network representation to obtain the structure of covariance matrices, which are then defined by few parameters. It is a procedure very similar to that used in multinomial probit traffic assignment, where the problem's covariance matrix is built as mentioned in chapter 4 (see also Sheffi, 1985).

6.3 Multinomial Probit Calibration Methods in the Literature

The literature on MNP calibration can be classified according to the sort of method used to evaluate the choice function: analytical or simulation based. Analytical approximations and numerical integration have been used in the past but most recent works are based on simulation, following the introduction of the GHK simulator (see *e.g.* Geweke *et al.*, 1994) illustrated in chapter 3. Quasi-Newton optimisation algorithms are generally used to solve for the parameters corresponding to the maximum likelihood.

The Clark approximation and the maximum likelihood method have been used in the two software packages CHOMP and QUAIL (cited respectively in Daganzo, 1979, and Kamakura, 1989) and Kamakura (1989) used the Langdon and the Mendell-Elston approximations in a maximum likelihood program. In particular, in CHOMP

numerical derivatives were used along with the Davidon-Fletcher-Powell (DFP) optimisation method (see Scales, 1985).

Numerical integration has been used for 3 choice options, but actually reducing the MNP integral to a unidimensional one, by Hausman and Wise (1978). Sheffi *et al.* (1982) discussed a transformation of the trivariate probit integral that reduces it to a univariate integral and illustrated a method for obtaining analytical derivatives of the likelihood function based on the transformation they proposed. This method was used in the development of the program TROMP (Sparmann and Daganzo, 1982, cited in Sheffi *et al.*, 1982 and Sparmann *et al.*, 1983), for the calibration of trivariate probit models.

Simulated maximum likelihood was first used by Lerman and Manski (1981) with their crude frequency simulator described in section 3.3. A further program based on a similar frequency simulator coupled with a quasi-Newton algorithm has been implemented by Lam (1991) (cited by Liu and Mahmassani, 2000) and further developed by Liu and Mahmassani (2000) who included a first stage of the parameters' optimisation performed with a genetic algorithm.

In most recent works on simulated maximum likelihood probit calibration the choice probabilities are calculated using the GHK simulator described in chapter 3 (Geweke *et al.*, 1994; Munizaga and Ortúzar, 1997; Munizaga *et al.*, 1997; Yai *et al.*, 1997; Yai and Shimizu, 1998; Bolduc, 1999; Munizaga *et al.*, 2000). In particular Bolduc (1999) used simulated maximum likelihood and the GHK simulator calculating analytically, rather than numerically, the first derivatives of the log-likelihood function used to solve the problem with a BHHH quasi-Newton optimisation algorithm (see Scales, 1985). The GHK simulator has also been used to calibrate the probit model with the method of simulated moments (McFadden, 1989; Geweke *et al.*, 1994).

6.4 Multinomial Probit Calibration Using the Mendell-Elston Approximation: Method

6.4.1 Introduction

A program for MNP calibration including the Mendell-Elston approximation has been developed and employed to carry out the numerical tests presented in this chapter.

The program maximises the logarithm of the MNP likelihood for a given set of choices and attributes by varying the parameters of the model. Random samples of data are assumed to be available for the calibration and therefore the log-likelihood function (6.2) is used.

The MNP choice function is solved by expressing it as a MVN integral in difference that is calculated using the analytical approximation of Mendell-Elston with a routine written for this application. The Mendell-Elston method is implemented with the calculation order suggested by Kamakura (1989), discussed in chapter 3. Only the probability of the chosen option for each sample data entry is calculated to save on computational effort. Thus the probabilities obtained are not normalised.

The maximisation of the likelihood function is carried out with a DFP-BFGS quasi-Newton method, although the simpler DFP method and the steepest ascent method may also be used, being simplifications of the DFP-BFGS.

The DFP-BFGS is the most efficient variable metric method in the literature (see *e.g.* Scales, 1985). As the other optimisation methods considered in this thesis, it finds the solution to the optimisation problem by moving iteratively by a suitable step along a search direction, from the current solution point to the next one, until a convergence criterion is satisfied. The search vectors \mathbf{p}_n are generated similarly to the those for the Newton method as:

$$\mathbf{p}_n = -\mathbf{H}_n \mathbf{g}_n \quad (6.7)$$

where \mathbf{g}_n is the gradient of the objective function at iteration n and \mathbf{H}_n is an approximation to the inverse of the Hessian of the quadratic approximation to the objective function at the same iteration. \mathbf{H}_n is not obtained directly (as it would in the Newton method) but is calculated using information on the function values and its gradient at each point explored. The method to approximate this inverse Hessian is what characterises the DFP-BFGS method (for details see Scales, 1985) which has quadratic convergence rate on quadratic functions. A particular point of interest is that this method generates an approximation to the inverse Hessian matrix at the solution point that may be used to evaluate the significance of the parameters estimated.

The partial derivatives of the likelihood function with respect to the parameters being calibrated are calculated numerically with the method of Ridder (as proposed by Press *et al.*, 1992) or analytically, according to the algorithm developed for this work and described in section 6.4.2.

The optimisations along the search directions are carried out by first characterising an interval containing the maximum of the log-likelihood function and then localising it by quadratic interpolation.

As mentioned above, to simplify the specification issue, the program calibrates MNP models expressed in terms of differences of the utilities with respect to the first choice option. Therefore, the estimates directly obtained from the calculations are those of the coefficients appearing in the systematic utilities and those of the elements of the covariance matrix of the model in difference with respect to the first choice option. The first diagonal entry of such reference covariance matrix in difference is not calibrated but set to a fixed value specified in the input to determine the scale of the model. The transformation of the model in difference into a possible version in terms of utilities is not considered here as it may have different solutions depending on the problem at hand and as it can be performed separately from the calibration of the model in difference.

To ensure the positive definiteness of the covariance matrix retrieved by the calibration algorithm, functional transformations of its terms, so that correlations would be within the interval $]-1;1[$ and variances would be strictly positive, have initially been considered in the program. However, such transformations may increase the complexity of the problem. Rather than calculating directly the covariance matrix in difference, the simpler device of calibrating its Cholesky decomposition has been eventually adopted and has been used to obtain the results reported in this chapter.

The program developed here does not include the possibility of considering different choice sets for different choice makers in a sample. Such different structure of the choice problem, however, can be readily included using the method given in Bolduc (1999).

6.4.2 Analytical Derivatives for the Mendell-Elston Log-Likelihood

6.4.2.1 Introduction

This section describes the method used to obtain the analytical derivatives of the MNP choice probability and of the resulting log-likelihood for a randomly sampled data set with the Mendell-Elston approximation and mentions how the same method can be adapted for use with other MNP approximations.

In solving an optimisation problem, analytical derivatives of the objective function are normally preferable to their numerical counterparts as they should be more precise and more efficiently calculated.

The calculation of the numerical derivatives entails an iterative process, potentially requiring several evaluations of the objective function (which, in the case of the MNP log-likelihood, is a relatively expensive operation), that stops when the derivatives are evaluated within a certain pre-specified precision.

The analytical derivatives of the MNP log-likelihood function calculated with the Mendell-Elston approximation can be obtained by exploiting the structure of the approximation calculations and can be computed along with the objective function at each point explored by an optimisation algorithm.

Given a set of parameters, the calculation of the contribution to the log-likelihood function of each data point in the set of observations used to calibrate the model can be divided in the following four main stages:

- P1. Calculation of the utilities and of the covariance matrix of the utilities in difference w.r.t. the utility of the alternative chosen (the covariance matrix of the utilities and the covariance matrices in difference w.r.t. each of the utilities are actually fixed for all data points).
- P2. Normalisation of the utility differences and of the covariance matrix in difference.
- P3. Recursive application of the approximation of Mendell-Elston and calculation of the probability of choice of the option actually chosen.
- P4. Calculation of the contribution of the data point to the value of the log-likelihood function.

The contributions to the derivatives of the log-likelihood function due to each data point are calculated analytically together with the probabilities according to the following stages:

- D1. Calculation of the derivatives of the probability of choice of the option actually chosen w.r.t. the parameters appearing at the beginning of the application of the Mendell-Elston approximation.
- D2. Calculation of the derivatives of the probability of the option actually chosen w.r.t. the parameters being calibrated *i.e.* the control variables of the optimisation.
- D3. Calculation of the contribution of the data point to the derivatives of the log-likelihood function w.r.t. the control variables of the optimisation.

The differentiation calculations at point D1 are carried out along with the operations at point P3 whilst the others follow. The description of the calculation method reported here is organised according to the three stages D1, D2, D3 listed above.

6.4.2.2 Calculation of the Derivatives of the Probability of the Option Actually Chosen w.r.t. the Parameters Appearing at the Beginning of the Application of the Mendell-Elston Approximation

The initial parameters of the Mendell-Elston calculations are, for each data point, those appearing in the MVN integral corresponding to the MNP choice function for the option actually chosen ch :

$$P_{ch}(\boldsymbol{\beta}_{(ch)}, \mathbf{0}, \mathbf{P}_{(ch)}) = \int_{z_1=-\infty}^{\beta_1} \int_{z_2=-\infty}^{\beta_2} \dots \int_{z_j=-\infty}^{\beta_j} \int_{z_{j-1}=-\infty}^{\beta_{j-1}} \left[(2\pi)^{J-1} |\mathbf{P}_{(ch)}| \right]^{\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{Y})^T \mathbf{P}_{(ch)}^{-1} (\mathbf{Y}) \right] dy_1 dy_2 \dots dy_j \dots dy_{j-1} \quad (6.8)$$

(the notation is as for (3.8) in section 3.2). The parameters appearing in (6.8) are the initial upper limits of integration β_j of the integral of the normalised MVN distribution $\mathbf{Y}=\text{MVN}(\mathbf{0},\mathbf{P})$ and the correlations ρ_{ji} between the utilities in difference w.r.t. the one of the actually chosen option, which is taken as reference for the Mendell-Elston calculations. In the discussion that follows these parameters are indicated respectively as $\beta_{j|0}$ and $\rho_{ji|0}$ where the addition to the subscript remarks that they are the parameters at the beginning of the calculation of (6.8) with the Mendell-Elston method. Moreover they are referred to collectively as the vector $\boldsymbol{\gamma} = [\beta_{1|0}, \dots, \beta_{j-1|0}, \rho_{12|0}, \dots, \rho_{j-1, j-2|0}]$ and any one entry of $\boldsymbol{\gamma}$ is generically referred to as γ_m .

The derivatives of the Mendell-Elston MNP probabilities w.r.t. these parameters can be obtained by applying the chain rule of differentiation to the parameters entering the calculations at each stage of the application of the approximation. To clarify the differentiation method and what is meant by the parameters entering the calculations

at each stage, it is useful to recall the way probabilities are obtained in the Mendell-Elston approximation.

Given the MVN integral (6.8), the Mendell-Elston method considers the distribution of the integrand and the marginal integral of one of the variates, say y_i , $\Phi(\beta_i)$, and approximates as multivariate Normal the conditional distribution of the remaining variates.

The standard deviation of one of the remaining variates, say y_j , conditional on $y_i < \beta_i$ results in:

$$\sigma_{j|i}^2 = 1 - \rho_{ji}^2 \frac{\varphi(\beta_i)}{\Phi(\beta_i)} \left(\frac{\varphi(\beta_i)}{\Phi(\beta_i)} + \beta_i \right) \quad (6.9)$$

The correlation between any two of the remaining variates, say y_j and y_k , conditional on $y_i < \beta_i$ is:

$$\rho_{jk|i} = \frac{\rho_{jk} - \rho_{ij}\rho_{ik} \frac{\varphi(\beta_i)}{\Phi(\beta_i)} \left(\frac{\varphi(\beta_i)}{\Phi(\beta_i)} + \beta_i \right)}{\sigma_{j|i} \sigma_{k|i}} \quad (6.10)$$

Considering a normalised conditional distribution, the integration limit of any of the remaining variates, say y_j , conditional on $y_i < \beta_i$, is calculated directly from its conditional mean as:

$$\beta_{j|i} = \frac{\left(\beta_j + \rho_{ji} \frac{\varphi(\beta_i)}{\Phi(\beta_i)} \right)}{\sigma_{j|i}} \quad (6.11)$$

Thus a n dimensional MVN integral can be calculated approximately by reducing it, in stages, to the product of n marginal univariate Normal integrals, by applying the approximation repeatedly.

The MNP choice probability for the chosen alternative in a choice set is given by the relevant MVN integral in difference (6.8). The approximated choice probability P_{ch}

obtained with the Mendell-Elston approximation (without considering the normalisation of the probabilities to their sum), results in:

$$P_{ch} = \Phi(\beta_{1|0})\Phi(\beta_{2|1})\Phi(\beta_{3|2})\dots\Phi(\beta_{J-1|J-2}) = \prod_{j=1}^{J-1} \Phi(\beta_{j|j-1}) \quad (6.12)$$

where $\beta_{j|j-1}$ are the upper integration limits for the univariate Normal integrals obtained according to the Mendell-Elston formula (6.11) and J is the dimension of the MNP choice problem, that corresponds to a MVN integral of dimension $J-1$. In the rest of this section, each of the marginal univariate Normal integral calculations and the related conditioning and approximation as MVN of the remaining variates, if any, is referred to as the stage or conditioning level in the calculations. The subscript for $\beta_{j|j-1}$ and other parameters, in (6.12) as well as in the rest of this section, indicates the conditioning level at which they are used (j) and the last level on which they are conditioned ($j-1$). Level 0 is the starting point: that is when no conditioning has been applied yet.

It follows from the rules of differentiation that the derivative of (6.12) w.r.t. a parameter γ_m is:

$$\frac{\partial P_{ch}}{\partial \gamma_m} = \sum_{j=1}^{J-1} \left(\frac{\partial \Phi(\beta_{j|j-1})}{\partial \gamma_m} \prod_{k \neq j} \Phi(\beta_{k|k-1}) \right) \quad (6.13)$$

The $\Phi(\beta_{j|j-1})$ are obtained as part of the process to calculate the choice probabilities

whilst each of the $\frac{\partial \Phi(\beta_{j|j-1})}{\partial \gamma_m}$ can be calculated using the chain rule and exploiting

the structure of the approximation calculations as described in the rest of this section.

At each stage of the Mendell-Elston calculations, $\Phi(\beta_{j|j-1})$ is only a function of $\beta_{j|j-1}$.

Therefore it is:

$$\frac{\partial \Phi(\beta_{j|j-1})}{\partial \beta_{j|j-1}} = \phi(\beta_{j|j-1}) \quad (6.14)$$

while the derivatives w.r.t. the other parameters of the MVN integral for this conditioning level are 0.

If $j-1$ is zero, this is the first level of conditioning, that is the first time the approximation is applied to the MVN integral being calculated. Thus $\beta_{j|j-1}$ is $\beta_{1|0}$ that is the upper integration limit appearing in (6.8) for the first variate included in the Mendell-Elston calculations and (6.14) is the only non-zero element of the vector of the derivatives $\frac{\partial \Phi(\beta_{1|0})}{\partial \gamma_m}$. The calculation of the derivatives of $\Phi(\beta_{1|0})$ w.r.t. γ is thus complete.

If this is not the first time the approximation is applied, it can be seen from (6.11) that $\beta_{j|j-1}$ is in turn a function only of some of the parameters defining the MVN integral at the previous conditioning level, namely only of $\beta_{j|j-2}$, that is the limit of integration, at the previous conditioning level, of the variate whose marginal integral is considered at level j , of $\beta_{j-1|j-2}$ the limit of integration of the variate whose marginal integral has been taken at the previous level, and of $\rho_{j,j-1|j-2}$ the correlation, at the previous level, between those two variates. To apply the chain rule of differentiation it is necessary to know the value of the relevant derivatives, that can be obtained considering (6.9) and (6.11) and are:

$$\frac{\partial \beta_{j|j-1}}{\partial \beta_{j|j-2}} = \frac{1}{\sigma_{j|j-1}} \quad (6.15)$$

$$\frac{\partial \beta_{j|j-1}}{\partial \beta_{j-1|j-2}} = \frac{\rho_{j,j-1|j-2} \sigma_{j|j-1} B_{j-1} + 0.5 \beta_{j|j-1} \rho_{j,j-1|j-2}^2 D_{j-1}}{\sigma_{j|j-1}^2} \quad (6.16)$$

$$\frac{\partial \beta_{j|j-1}}{\partial \rho_{j,j-1|j-2}} = \frac{\sigma_{j|j-1} A_{j-1} + \beta_{j|j-1} \rho_{j,j-1|j-2} C_{j-1}}{\sigma_{j|j-1}^2} \quad (6.17)$$

where:

$$A_{j-1} = \frac{\varphi(\beta_{j-1|j-2})}{\Phi(\beta_{j-1|j-2})} \quad (6.18)$$

$$B_{j-1} = \frac{\partial A_{j-1}}{\partial \beta_{j-1|j-2}} = \frac{-\beta_{j-1|j-2} \varphi(\beta_{j-1|j-2}) \Phi(\beta_{j-1|j-2}) - (\varphi(\beta_{j-1|j-2}))^2}{(\Phi(\beta_{j-1|j-2}))^2} \quad (6.19)$$

$$C_{j-1} = A_{j-1} (A_{j-1} + \beta_{j-1|j-2}) = \frac{\varphi(\beta_{j-1|j-2})}{\Phi(\beta_{j-1|j-2})} \left(\frac{\varphi(\beta_{j-1|j-2})}{\Phi(\beta_{j-1|j-2})} + \beta_{j-1|j-2} \right) \quad (6.20)$$

$$D_{j-1} = \frac{\partial C_{j-1}}{\partial \beta_{j-1|j-2}} = 2A_{j-1}B_{j-1} + B_{j-1}\beta_{j-1|j-2} + A_{j-1} \quad (6.21)$$

The derivatives of $\beta_{j|j-2}$ w.r.t. the other parameters conditional on level $j-2$ are zero.

If j is 2 and level $j-2$ is level 0, applying the chain rule once using the (6.14) and the

(6.15)-(6.17) allows us to obtain the vector $\left[\frac{\partial \Phi(\beta_{2|1})}{\partial \gamma_m} \right]$ of the partial derivatives of

$\Phi(\beta_{2|1})$ w.r.t. the parameters in γ :

$$\left[\frac{\partial \Phi(\beta_{2|1})}{\partial \gamma_m} \right] = \frac{d\Phi(\beta_{2|1})}{d\beta_{2|1}} \begin{bmatrix} \frac{\partial \beta_{2|1}}{\partial \beta_{1|0}} \\ \frac{\partial \beta_{2|1}}{\partial \rho_{2|1|0}} \\ \frac{\partial \beta_{2|1}}{\partial \beta_{2|1}} \\ \frac{\partial \beta_{2|1}}{\partial \beta_{2|0}} \end{bmatrix}^T = \varphi(\beta_{2|1}) \begin{bmatrix} \frac{\partial \beta_{2|1}}{\partial \beta_{1|0}} \\ \frac{\partial \beta_{2|1}}{\partial \rho_{2|1|0}} \\ \frac{\partial \beta_{2|1}}{\partial \beta_{2|1}} \\ \frac{\partial \beta_{2|1}}{\partial \beta_{2|0}} \end{bmatrix}^T \quad (6.22)$$

If $\Phi(\beta_{2|1})$ is obtained as part of calculations involving more than two variates, *i.e.* more than two conditioning levels, there will be further entries in the column vector in (6.22) whose value will be zero because they will be referred to parameters in γ that have an effect only at further conditioning levels.

If the level of conditioning considered is at least the third, which means that this is the third time that a marginal integral is calculated, and j in $\beta_j|_{j-1}$ is equal to 3 or more, the parameters w.r.t. which the derivatives are taken in (6.15), (6.16) and (6.17) are a function of those at the previous level of conditioning. In particular, $\beta_j|_{j-2}$ and $\beta_{j-1}|_{j-2}$ are function of parameters at the previous level of conditioning similarly to $\beta_j|_{j-1}$ above (therefore according to formulae similar to 6.11) and can thus be differentiated according to formulae similar to (6.15), (6.16) and (6.17). The $\rho_{j,j-1|j-2}$ is a function of $\rho_{j,j-1|j-3}$, $\rho_{j-1,j-2|j-3}$, $\rho_{j-2|j-3}$ and $\beta_{j-2}|_{j-3}$ as can be seen considering formulae (6.10) and (6.9). The corresponding derivatives are:

$$\frac{\partial \rho_{j,j-1|j-2}}{\partial \rho_{j,j-1|j-3}} = \frac{1}{\sigma_{j,j|j-2} \sigma_{j-1,j-1|j-2}} \quad (6.23)$$

$$\frac{\partial \rho_{j,j-1|j-2}}{\partial \rho_{j-1,j-2|j-3}} = C_{j-1} \left[\frac{-\rho_{j,j-2|j-3}}{\sigma_{j,j|j-2} \sigma_{j-1,j-1|j-2}} + \rho_{j,j-1|j-2} \frac{\rho_{j-1,j-2|j-3}}{\sigma_{j-1|j-2}^2} \right] \quad (6.24)$$

$$\frac{\partial \rho_{j,j-1|j-2}}{\partial \rho_{j,j-2|j-3}} = C_{j-1} \left[\frac{-\rho_{j-1,j-2|j-3}}{\sigma_{j,j|j-2} \sigma_{j-1,j-1|j-2}} + \rho_{j,j-1|j-2} \frac{\rho_{j,j-2|j-3}}{\sigma_{j|j-2}^2} \right] \quad (6.25)$$

$$\begin{aligned} \frac{\partial \rho_{j,j-1|j-2}}{\partial \beta_{j-2|j-3}} = & -\frac{\rho_{j,j-2|j-3} \rho_{j-1,j-2|j-3} D_{j-1}}{\sigma_{j,j|j-2} \sigma_{j-1,j-1|j-2}} + \\ & + \frac{\rho_{j,j-1|j-2} D_{j-1}}{2} \left(\frac{\rho_{j,j-2|j-3}^2}{\sigma_{j|j-2}^2} + \frac{\rho_{j-1,j-2|j-3}^2}{\sigma_{j-1|j-2}^2} \right) \end{aligned} \quad (6.26)$$

The derivatives of $\rho_{j,j-1|j-2}$ w.r.t. other parameters conditional on level $j-3$ are zero.

Thus, if level $j-3$ is level 0 and the vector $\left[\frac{\partial \Phi(\beta_{3|2})}{\partial \gamma_m} \right]$ of partial derivatives of $\Phi(\beta_{3|2})$

w.r.t the parameters γ appearing at conditioning level 0 is being calculated, the formulae above can be used to apply the chain rule as follows:

$$\begin{aligned} & \left[\frac{\partial \Phi(\beta_{3|2})}{\partial \gamma_m} \right] = \\ & = \frac{d\Phi(\beta_{3|2})}{d\beta_{3|2}} \begin{bmatrix} \frac{\partial \beta_{3|2}}{\partial \beta_{3|2}} \\ \frac{\partial \beta_{2|1}}{\partial \beta_{3|2}} \\ \frac{\partial \beta_{3|2}}{\partial \rho_{32|1}} \\ \frac{\partial \beta_{3|2}}{\partial \beta_{3|1}} \\ \frac{\partial \beta_{3|1}}{\partial \beta_{3|1}} \end{bmatrix}^T \begin{bmatrix} \frac{\partial \beta_{2|1}}{\partial \beta_{1|0}} & \frac{\partial \beta_{2|1}}{\partial \rho_{21|0}} & \frac{\partial \beta_{2|1}}{\partial \beta_{2|0}} & \frac{\partial \beta_{2|1}}{\partial \rho_{31|0}} & \frac{\partial \beta_{2|1}}{\partial \rho_{32|0}} & \frac{\partial \beta_{2|1}}{\partial \beta_{3|0}} \\ \frac{\partial \beta_{32|1}}{\partial \beta_{1|0}} & \frac{\partial \beta_{32|1}}{\partial \rho_{21|0}} & \frac{\partial \beta_{32|1}}{\partial \beta_{2|0}} & \frac{\partial \beta_{32|1}}{\partial \rho_{31|0}} & \frac{\partial \beta_{32|1}}{\partial \rho_{32|0}} & \frac{\partial \beta_{32|1}}{\partial \beta_{3|0}} \\ \frac{\partial \beta_{3|1}}{\partial \beta_{1|0}} & \frac{\partial \beta_{3|1}}{\partial \rho_{21|0}} & \frac{\partial \beta_{3|1}}{\partial \beta_{2|0}} & \frac{\partial \beta_{3|1}}{\partial \rho_{31|0}} & \frac{\partial \beta_{3|1}}{\partial \rho_{32|0}} & \frac{\partial \beta_{3|1}}{\partial \beta_{3|0}} \end{bmatrix} \end{aligned} \quad (6.27)$$

$$= \varphi(\beta_{3|2}) \begin{bmatrix} \frac{\partial \beta_{3|2}}{\partial \beta_{3|2}} \\ \frac{\partial \beta_{2|1}}{\partial \beta_{3|2}} \\ \frac{\partial \beta_{3|2}}{\partial \rho_{32|1}} \\ \frac{\partial \beta_{3|2}}{\partial \beta_{3|1}} \\ \frac{\partial \beta_{3|1}}{\partial \beta_{3|1}} \end{bmatrix}^T \begin{bmatrix} \frac{\partial \beta_{2|1}}{\partial \beta_{1|0}} & \frac{\partial \beta_{2|1}}{\partial \rho_{21|0}} & \frac{\partial \beta_{2|1}}{\partial \beta_{2|0}} & 0 & 0 & 0 \\ \frac{\partial \beta_{32|1}}{\partial \beta_{1|0}} & \frac{\partial \beta_{32|1}}{\partial \rho_{21|0}} & 0 & \frac{\partial \beta_{32|1}}{\partial \rho_{31|0}} & \frac{\partial \beta_{32|1}}{\partial \rho_{32|0}} & 0 \\ \frac{\partial \beta_{3|1}}{\partial \beta_{1|0}} & 0 & 0 & \frac{\partial \beta_{3|1}}{\partial \rho_{31|0}} & 0 & \frac{\partial \beta_{3|1}}{\partial \beta_{3|0}} \end{bmatrix}$$

If $\Phi(\beta_{3|2})$ is used as part of a calculation with more than three conditioning levels its derivatives w.r.t. the parameters appearing in γ but not in (6.27) will be zero.

If j in $\beta_{j|j-1}$ is four or more, and therefore the derivatives of $\Phi(\beta_{4|3})$, $\Phi(\beta_{5|4})$, and so on, are being calculated w.r.t. the parameters $\gamma = [\beta_{1|0}, \dots, \beta_{j-1|0}, \rho_{12|0}, \dots, \rho_{j-1, j-2|0}]$ at level 0, the chain rule can be applied similarly to above. In general, the vector of

derivatives $\left[\frac{\partial \Phi(\beta_{j|j-1})}{\partial \gamma_m} \right]$ can be written using the chain rule for all relevant levels as:

$$\left[\frac{\partial \Phi(\beta_{j|j-1})}{\partial \gamma_m} \right] = \frac{d\Phi(\beta_{j|j-1})}{d\beta_{j|j-1}} \prod_{k=j-1}^1 \mathbf{E}_{k|k-1} \quad (6.28)$$

where $\mathbf{E}_{k|k-1}$ is the matrix of derivatives of the parameters at level k w.r.t. the parameters at level $k-1$. The differentiation formulae, (6.15) to (6.17) and (6.23) to (6.26), cover all the cases necessary to form the non-zero entries of any of the matrices $\mathbf{E}_{k|k-1}$.

For each conditioning level, the order in which the parameters are included in the matrices of derivatives $\mathbf{E}_{k|k-1}$ is as they would be in a vector stacking the unique entries of a symmetric matrix which has as diagonal entries the $\beta_{k|k-1}$ for the variates whose marginal integral has not yet been taken at level k and, as the off-diagonal entries, the correlations between those variates at level k .

This structure of the matrices of derivatives is the one used in the implementation of the procedure in the computer program developed. The actual coding of the procedure is complicated by the fact that the order in which the variates are included in the calculations may vary to improve the precision of the result (see 3.5.3) so it is also necessary to keep track of which are the parameters, at each level of conditioning, relevant to the calculations at further conditioning levels. Although the details of the implementation of the routine are not discussed here, it is worth mentioning that part of the calculation effort is saved by observing that some entries of the matrices of derivatives at each stage can simply be taken from the matrices of

derivatives used to write (6.28) to calculate the vector $\left[\frac{\partial \Phi(\beta_{j|j-1})}{\partial \gamma_m} \right]$ at previous

stages. This can be seen from the description above by observing that the vector of derivatives appearing as the second term in the product in (6.22) gives also the first three entries of the first line in the $[3 \times 6]$ matrix in (6.27) which are also all the non-zero terms of that line. In fact, if there are three levels of calculations the column vector in (6.22) is a column vector with three added entries equal to zero and that vector gives the whole first line of the $[3 \times 6]$ matrix in (6.27).

Similarly, if there were a 4th level of calculations and a $\Phi(\beta_{4|3})$ to differentiate, the $[3 \times 6]$ matrix in (6.27) would be all that is necessary to form the first 3 lines of the $[6 \times 10]$ matrix constituting the fourth and last element of the product (6.28) (because the remaining entries on those lines are zero), and the $[1 \times 3]$ matrix in (6.22) would form the first 3 entries of that $[3 \times 6]$ matrix (the other entries on that line are again zero).

Once all the vectors of $\left[\frac{\partial \Phi(\beta_{j|j-1})}{\partial \gamma_m} \right]$ for a data point have been obtained using the procedure illustrated above and the $\Phi(\beta_{j|j-1})$ have been calculated as part of the process to compute the probability of the option chosen for that data point, the entries of the vector $\left[\frac{\partial P_{ch}}{\partial \gamma_m} \right]$ of the derivatives of the non-normalised probability of the option actually chosen with respect to the parameters γ at the beginning of the Mendell-Elston calculations can be obtained according to formula (6.13).

6.4.2.3 Calculation of the Derivatives of the Probability of the Option Actually Chosen w.r.t. the Control Variables of the Optimisation

At this point in the calculations the derivatives of the probability of the option actually chosen still need to be referred, using again the chain rule, to the control variables of the calibration problem, that are the coefficients appearing in the systematic utilities of the alternatives and the Cholesky factors of the reference covariance matrix in difference. The derivatives w.r.t. the coefficients appearing in the systematic utilities can be obtained immediately with one additional chain rule step whilst the others require more calculations.

In the previous section $\beta_{j|j-1}$ referred to the variate entering the calculations at level j . If the processing order is optimised the variate entering the calculations at level j is not necessary that of place j in the vector of utilities in difference w.r.t. the utility of the chosen option. However, for simplicity, in the description that follows it is assumed that the variates are processed in the order in which they appear in the vector of utilities in difference w.r.t. the chosen option. Thus $\beta_{j|j-1}$ corresponds also to the j th variate in the vector of utilities in difference w.r.t. the chosen option and, similarly, $\rho_{ij|j-1}$ corresponds to the covariance of place ij in the covariance matrix of the utilities in difference. In the program actually coded the processing order of the variates is optimised and the present differentiation step is used to refer the derivatives to the parameters as they are ordered in the vector of utilities in difference w.r.t. the chosen option by knowing which corresponds to the variate entered in the calculations at each step.

This last part of the procedure is similar to the procedures suggested by Sheffi *et al.* (1982) and Bolduc (1999), when allowance is made for the different formulations of the problem, as they also proposed procedures to obtain analytical derivatives based, as is necessary, on the chain rule, but for numerical integration and for the GHK simulator.

The integration limits and correlations of the normalised MVN distribution in (6.8) will have been obtained from the distribution of utilities in difference w.r.t the chosen alternative as:

$$\beta_{j|0} = \frac{V_{ch} - V_l}{\omega_{jj(ch)}} = \frac{\alpha_{0ch} + \sum_E \alpha_e a_{ech} - \alpha_{0l} - \sum_E \alpha_e a_{el}}{\omega_{jj(ch)}} \quad (6.29)$$

$$\rho_{ij|0} = \frac{\omega_{ij(ch)}^2}{\omega_{ii(ch)}\omega_{jj(ch)}} \quad (6.30)$$

where the subscript *ch* indicates the chosen option and the subscript *l* indicates the option whose utility in difference is on line *j* of the vector of utilities in difference w.r.t. the utility of the chosen option. α_{0l} is the alternative specific constant for option *l*, α_e is the coefficient for attribute *e*, a_{el} is the value of the attribute *e* for option *l* and $\omega_{ij(ch)}^2$ is the entry of position *ij* of the covariance matrix of the utilities in difference w.r.t. the chosen option.

Starting from the derivatives $\left[\frac{\partial P_{ch}}{\partial \gamma_m} \right]$ of the probability of the chosen option obtained

with (6.13) and applying once more the chain rule considering (6.29) allows us to obtain the derivatives of the probability of the option actually chosen for the data point w.r.t. the parameters appearing in the systematic part of the utility.

In fact, if α_{0k} is an alternative specific constant (ASC):

$$\frac{\partial \beta_{j|0}}{\partial \alpha_{0k}} = \begin{cases} -\frac{1}{\omega_{jj(ch)}} & \text{if } \alpha_{0k} \text{ is the ASC of option } l (k = l) \\ \frac{1}{\omega_{jj(ch)}} & \text{if } \alpha_{0k} \text{ is the ASC of the option whose} \\ & \text{prob. is being calc. (} k = ch \text{)} \\ 0 & \text{otherwise} \end{cases} \quad (6.31)$$

If α_e is a coefficient of an attribute:

$$\frac{\partial \beta_{j|0}}{\partial \alpha_e} = \frac{\delta_{ech} a_{ech} - \delta_{el} a_{el}}{\omega_{jj(ch)}} \quad (6.32)$$

where δ_{el} is the Kronecker delta indicating whether α_e appears in the systematic utility of alternative l , and a_{el} is the attribute for α_e in alternative l .

As can be seen from (6.30) the derivatives of $\rho_{ij|0}$ w.r.t. any of the parameters in the systematic utility will be zero.

The vector $\left[\frac{\partial P_{ch}}{\partial \alpha_e} \right]$ of derivatives of the probability of the option actually chosen w.r.t. the coefficients in the systematic utilities of the alternatives are thus finally obtained as:

$$\left[\frac{\partial P_{ch}}{\partial \alpha_e} \right] = \left[\frac{\partial P_{ch}}{\partial \gamma_m} \right] \left[\frac{\partial \gamma_m}{\partial \alpha_e} \right] \quad (6.33)$$

where $\left[\frac{\partial P_{ch}}{\partial \gamma_m} \right]$ was obtained as described in the previous section and the matrix

$\left[\frac{\partial \gamma_m}{\partial \alpha_e} \right]$ is obtained by calculating its entries with (6.31) and (6.32).

Both $\beta_{j|0}$ and $\rho_{ji|0}$ are functions of the elements of the covariance matrix in difference w.r.t. the chosen alternative. $\beta_{j|0}$ has a derivative that is non-zero only when differentiated w.r.t. $\omega_{jj(ch)}^2$. In this case the derivative is:

$$\frac{\partial \beta_{j|0}}{\partial \omega_{jj(ch)}^2} = -\frac{1}{2} \frac{V_{ch} - V_j}{\omega_{jj(ch)}^3} \quad (6.34)$$

$\rho_{ji|0}$ has a derivative different from zero when differentiated w.r.t. $\omega_{ji(ch)}^2$, $\omega_{ii(ch)}^2$, $\omega_{jj(ch)}^2$. The first of those derivatives is simply:

$$\frac{\partial \rho_{ji|0}}{\partial \omega_{ji(ch)}^2} = \frac{1}{\omega_{jj(ch)} \omega_{ii(ch)}} \quad (6.35)$$

and the others are:

$$\frac{\partial \rho_{ji|0}}{\partial \omega_{ii(ch)}^2} = -\frac{1}{2} \frac{\omega_{ji(ch)}^2}{\omega_{ii(ch)}^3 \omega_{jj(ch)}} = -\frac{1}{2} \frac{\rho_{ji|0}}{\omega_{ii(ch)}^2} \quad (6.36)$$

$$\frac{\partial \rho_{ji|0}}{\partial \omega_{jj(ch)}^2} = -\frac{1}{2} \frac{\omega_{ji(ch)}^2}{\omega_{ii(ch)} \omega_{jj(ch)}^3} = -\frac{1}{2} \frac{\rho_{ji|0}}{\omega_{jj(ch)}^2} \quad (6.37)$$

Thus the derivatives of the probability of the option actually chosen for the data point referred to the unique entries of the relevant covariance matrix in difference

can be obtained by multiplying the vector $\left[\frac{\partial P_{ch}}{\partial \gamma_m} \right]$ of derivatives referred to the initial

Mendell-Elston parameters γ by a matrix $\left[\frac{\partial \gamma_m}{\partial \omega_{jk(ch)}^2} \right]$ of the derivatives of those

parameters w.r.t. the parameters of the covariance matrix in difference obtained with the formulae (6.34)-(6.37):

$$\left[\frac{\partial P_{ch}}{\partial \omega_{jk(ch)}^2} \right] = \left[\frac{\partial P_{ch}}{\partial \gamma_m} \right] \left[\frac{\partial \gamma_m}{\partial \omega_{jk(ch)}^2} \right] \quad (6.38)$$

The derivatives $\left[\frac{\partial P_{ch}}{\partial \omega_{jk(ch)}^2} \right]$ still need to be referred to the elements of the reference

covariance matrix (that here is the one in difference with w.r.t. the first alternative) if they are not already, and then to the actual control variables of the problem: the

Cholesky factors of the reference covariance matrix in difference, except the one in position (1,1) that is fixed. If alternative control variables are considered, the latter step must be altered accordingly.

If the chosen option for a data point is not the reference one, option one, the vector of derivatives $\left[\frac{\partial P_{ch}}{\partial \omega_{jk(ch)}^2} \right]$ is multiplied by the matrix $\left[\frac{\partial \omega_{jk(ch)}^2}{\partial \omega_{pq(1)}^2} \right]$ of the derivatives of the unique entries of the matrix in difference used in the calculations w.r.t. the unique elements of the reference matrix in difference. Each column of $\left[\frac{\partial \omega_{jk(ch)}^2}{\partial \omega_{pq(1)}^2} \right]$ corresponds to a unique element of the reference matrix in difference and each row to a unique element of the current matrix in difference. There is a matrix $\left[\frac{\partial \omega_{jk(ch)}^2}{\partial \omega_{pq(1)}^2} \right]$ for each option except the reference option. Since they are fixed, they have been pre-written in the calibration program developed. They have been obtained simply by considering the operation to transform the covariance matrix $\mathbf{\Omega}_{(1)}$ of the utilities in difference w.r.t. the utility of the reference option into the covariance matrix $\mathbf{\Omega}_{(n)}$ of the utilities in difference w.r.t. a different option:

$$\mathbf{\Omega}_{(n)} = \mathbf{\Gamma}_{(n)} \mathbf{\Omega}_{(1)} \mathbf{\Gamma}_{(n)}^T \quad (6.39)$$

where $\mathbf{\Gamma}_{(n)}$ is the matrix that allows us to change a problem in difference w.r.t. option one to a problem in difference w.r.t. option n . It has been described by Bunch (1991) and is obtained by simply taking the matrix $\mathbf{\Delta}_{(n)}$ described in chapter three, and deleting the column of the reference option, option one in this case. The elements of the column of $\left[\frac{\partial \omega_{jk(ch)}^2}{\partial \omega_{pq(1)}^2} \right]$ for an element of $\mathbf{\Omega}_{(1)}$ are obtained from (6.39) as the unique elements of $\mathbf{\Omega}_{(n)}$ when instead of $\mathbf{\Omega}_{(1)}$ a matrix $\underline{\mathbf{\Omega}}_{(1)}$ with entry 1 in the position of the element of $\mathbf{\Omega}_{(1)}$ of interest and 0 in the other places is used.

Once the vector $\left[\frac{\partial P_{ch}}{\partial \omega_{pq(1)}^2} \right]$ of derivatives of the probability of the option actually chosen for the considered data point w.r.t. the unique entries of the reference

covariance matrix in difference is obtained, these derivatives are further referred to the Cholesky factors $c_{ij(1)}$ of such a covariance matrix, that are used as control variables of the likelihood maximisation problem. The vector $\left[\frac{\partial P_{ch}}{\partial c_{ij(1)}} \right]$ of those derivatives is calculated simply by multiplying the vector $\left[\frac{\partial P_{ch}}{\partial \omega_{pq(1)}^2} \right]$ of the derivatives of the probability w.r.t. the unique elements of the reference covariance matrix in difference by the matrix $\left[\frac{\partial \omega_{pq(1)}^2}{\partial c_{ij(1)}} \right]$ of the derivatives of the elements of the latter w.r.t. the elements of its Cholesky factorisation. This matrix is fixed at each solution point explored.

The final result of the operations described in this section are the vectors $\left[\frac{\partial P_{ch}}{\partial \alpha_e} \right]$ and $\left[\frac{\partial P_{ch}}{\partial c_{ij(1)}} \right]$ of the derivatives of the Mendell-Elston choice probability of the option actually chosen for a data point w.r.t. the control variables of the calibration problem.

6.4.2.4 Calculation of the Derivatives of the Log-Likelihood Function w.r.t. the Control Variables of the Optimisation

Finally, the derivatives of the log-likelihood function for a set of surveyed data and for a set of model parameters can be obtained by knowing the choice probabilities of the chosen options and their derivatives, as calculated above.

The notation used in the previous section can be compacted by writing together the control variables of the optimisation as the vector $\theta = (\alpha_{01}, \dots, \alpha_{eN}, c_{12(1)}, \dots, c_{N-1, N-1(1)})$. Thus, the derivative of the log-likelihood function (6.2) w.r.t. θ_i , one of the control variables of the problem, is:

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left(\sum_{s=1}^S \log P_s(\boldsymbol{\theta}) \right) = \sum_{s=1}^S \frac{\partial}{\partial \theta_i} (\log P_s(\boldsymbol{\theta})) = \sum_{s=1}^S \frac{1}{P_s(\boldsymbol{\theta})} \frac{\partial P_s(\boldsymbol{\theta})}{\partial \theta_i} \quad (6.40)$$

where $P_s(\boldsymbol{\theta})$ is a short expression for the probability of choice of the option actually chosen by the choice maker s in the sample of data expressed as a function of the vector $\boldsymbol{\theta}$ whilst $\frac{\partial P_s(\boldsymbol{\theta})}{\partial \theta_i}$ is its derivative w.r.t. the control variable θ_i , obtained as described in the previous sections.

6.4.2.5 Extension of the Method to Consider Probabilities Normalised to Their Sum

The method proposed above and the program used for the experiments reported in this chapter do not include the normalisation of the probabilities to their sum, thus limiting the calculations to the choice probability of the option chosen for each sample data point.

Alternatively, indicating with a dash the probabilities normalised to their sum, expressed as:

$$P'_{ch} = \frac{P_{ch}}{\sum_{j=1}^J P_j} \quad (6.41)$$

where j is one of the J options in the choice set, their derivatives w.r.t. the control variables of the log-likelihood optimisation problem should be consistently obtained as:

$$\frac{\partial P'_{ch}(\boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left(\frac{P_{ch}(\boldsymbol{\theta})}{\sum_{j=1}^J P_j(\boldsymbol{\theta})} \right) = \frac{\frac{\partial P_{ch}(\boldsymbol{\theta})}{\partial \theta_i} \sum_{j=1}^J P_j(\boldsymbol{\theta}) - P_{ch}(\boldsymbol{\theta}) \sum_{j=1}^J \frac{\partial P_j(\boldsymbol{\theta})}{\partial \theta_i}}{\left(\sum_{j=1}^J P_j(\boldsymbol{\theta}) \right)^2} \quad (6.42)$$

This formula requires the calculation of the choice probabilities and of the derivatives for the non-chosen alternatives, some of which may have a very small choice probability, approximated as zero by the program. In such cases the derivative could be approximated as zero. This device is currently adopted in the

program coded to deal with points in the solution space particularly far from the optimum (that have, however, not been encountered in the calculations to obtain the results reported later) and assumes, approximately, that in the neighbourhood of a point where the probability is negligible its variation is also negligible. This heuristic procedure seems to work well in practice and the goodness of the assumption and of the approximation obtained are also confirmed when comparing the analytical derivatives thus calculated with the numerical ones obtained for the same points.

6.4.2.6 Analytical Derivatives with Other Approximations

The method of application of the chain rule of differentiation discussed in section 6.4.2.2 to obtain the derivatives of the probability of the chosen option exploiting the structure of the Mendell-Elston approximation can be adapted for use with other approximations.

For instance, the method described can be readily applied when the (6.8) is solved using the Taylor series approximation. Focussing on the first order Taylor series approximation, the differentiation can be organised similarly according to the levels of conditioning. The entries of the matrices of derivatives $E_{k|k-1}$ in (6.28) can be calculated with formulae different from those for the Mendell-Elston case only because the conditional covariance matrices do not depend on the limit of integration of the conditioning variate. This can be readily seen by comparing the formulae for the Mendell-Elston and for the Taylor series approximations reported in chapter 3.

When the improved Clark approximation is used to solve the MNP choice function posed as in (3.5), the chain rule can be applied similarly but considering the different structure of the approximation. The compact formula for the application of the chain rule is again as (6.28) but the levels or stages in the calculations are defined by the approximation as Normal of the maximum of two variates, rather than by a conditioning as in the Mendell-Elston case. Observing the formulae of the approximation, the dependence of the parameters used at a calculation level on those at the previous calculation level can be readily characterised. Thus the formulae for the derivatives necessary to obtain the matrices corresponding to the $E_{k|k-1}$ in (6.28) can be obtained. Since in this case the formulation of the MNP choice function (3.5),

rather than the (6.8), is solved the following calculations to refer the derivatives to the control variables of the optimisation problem are simplified.

6.5 Multinomial Probit Calibration Using the Mendell-Elston Approximation: Numerical Tests

6.5.1 Introduction

The computer program for MNP calibration implementing the Mendell-Elston approximation and using either numerical derivatives or the analytical derivative calculation method described in section 6.4, has been employed to carry out experiments to confirm the efficiency of using analytical derivatives and to test the ability of the MNP solution method used and of the likelihood framework to retrieve the parameters of known models as a function of the sample size in cases with 3 and 4 choice options.

An in depth study of the second subject would require extensive numerical tests and the results presented here are intended simply to give a first idea of the results that could be obtained.

6.5.2 Test Methodology

The efficiency of using analytical derivatives compared to using numerical ones and the retrieval accuracy of the MNP model solved with the Mendell-Elston approximation have been tested by recording the calculation time and comparing the calibration results in a number of test cases with 3 or 4 choice options.

To carry out the tests, a number of model structures have been characterised, then for each model structure the “true” parameters have been set and 10 artificial data sets consisting of the choice made and the values of the attributes appearing in the model for 10000 instances of choice have been generated with a computer program. These data sets are intended to replicate the data that would be collected in a survey.

The data generation program obtains the attributes of the options as the product of a base data drawn from a specified uniform distribution for each individual (*i.e.* for each data point) and the ratio between the attribute and the base data, which is drawn from a different specified uniform distribution. Once calculated, the attributes appearing in each alternative's utility are weighted according to the set model coefficients and summed to form the systematic parts of the utilities. It should be noted that the attributes are all option specific, that is attributes for the same coefficients have been drawn from different distributions for each choice option.

The random terms of the utilities are drawn from their distribution as specified in the input and summed to the systematic part of the utilities. The utilities are then ranked and the resulting choice is provided as output along with the value of the corresponding attributes used.

For each of the ten data sets generated for each true model, calibrations were carried out using subsets of 500, 1000, 2500, 5000, 7500 and 10000 data points. Thus for each structure, 10 models have been calibrated for each sample size.

The calibration results presented later in the chapter include the average calculation time for each model structure and sample size and the aggregate calibration results, meant to represent the likely range of estimates retrievable for each sample size.

Although an analyst will normally have some initial idea about the sign of the coefficients appearing in the systematic utility, the calibrations have been started with systematic utility parameters set to zero and Cholesky factors of the covariance matrix in difference with the first utility all set to one except that in position (1,1) that is fixed for scale identification and set to the corresponding value used in generating the data so that the retrieved parameters are directly comparable with those used for data generation.

Such a starting point without use of prior information has been used to leave the determination of the coefficients completely to the calibration procedure.

The convergence criterion to terminate the calibration calculations is based on the norm of the gradient of the log-likelihood function: when it is less than 10^{-2} (a threshold that may be changed from the program input) convergence is assumed to be achieved. Also additional secondary checks on the variations of the log-likelihood and of the parameters are performed along with the gradient check, to confirm the agreement of the gradient and the function.

The Mendell-Elston approximation employed follows the calculation order of Kamakura (1989) discussed in chapter 3. To avoid instabilities in the calculations, the order is fixed at each iteration to the one followed at the current solution. This means that the same order is used to perform the line search and compute the numerical derivatives, which, this way, are consistent with the analytical ones, as verified. The calculation order is fixed also across iterations when final convergence is approached. This is detected by observing again the gradient norm. For the experiments whose results are reported here, the order is fixed when the gradient norm is less than 10.

The calculation times have been measured on a desktop computer Pentium II, 350 MHz.

6.5.3 Test Bed

All true models used to obtain the artificial data for the experiments reported in this chapter are MNP models. The covariance matrices calibrated from the data are those of the utilities in difference w.r.t. the utility of the first option and are indicated for each model structure. Also indicated, but only for reference, are the true covariance matrices of the utilities to which they correspond.

Three model structures have been employed for the 3 choice option cases.

First, data obtained from a very simple MNP model with one attribute and identical and independent errors have been generated. The model structure, referred to as 3ai, is:

$$\begin{aligned}
V_1 &= \alpha_1 + \alpha_3 X_{11} \\
V_2 &= \alpha_3 X_{12} \\
V_3 &= \alpha_2 + \alpha_3 X_{13}
\end{aligned}
\quad
\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}
\quad
\Omega_{(1)} = \begin{bmatrix} \omega_{11}^2 & \omega_{12}^2 \\ \omega_{21}^2 & \omega_{22}^2 \end{bmatrix}
\quad (6.43)$$

$\Omega_{(1)}$ is, in this case as the following ones, the covariance matrix of the utilities in difference w.r.t. that of the first option, which is the covariance matrix that is actually estimated directly from the calculations. To simplify the notation, in (6.43) and in the rest of the chapter the indication of the reference option has been omitted from the subscripts of the elements of $\Omega_{(1)}$.

The parameters used to generate the data are listed in the following table 6.1.

Parameter	Value	Parameter	Value
α_1	-0.5	σ^2	1.0
α_2	-1.0	$\omega_{11}^2 = \omega_{22}^2$	2.0
α_3	-1.5	ω_{12}^2	1.0

Table 6.1 – Parameters for the model structure 3ai.

A second test case with three options has been obtained by including three attributes in the systematic utility of each option and using again an i.i.d. covariance matrix:

$$\begin{aligned}
V_1 &= \alpha_1 + \alpha_3 X_{11} + \alpha_4 X_{21} + \alpha_5 X_{31} \\
V_2 &= \alpha_3 X_{12} + \alpha_4 X_{22} + \alpha_5 X_{32} \\
V_3 &= \alpha_2 + \alpha_3 X_{13} + \alpha_4 X_{23} + \alpha_5 X_{33}
\end{aligned}
\quad
\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}
\quad (6.44)$$

$$\Omega_{(1)} = \begin{bmatrix} \omega_{11}^2 & \omega_{12}^2 \\ \omega_{21}^2 & \omega_{22}^2 \end{bmatrix}$$

The actual parameters used to generate the data for this test case (referred to as 3ci) are listed in table 6.2.

Parameter	Value	Parameter	Value
α_1	-2.0	α_5	-0.25
α_2	-1.0	σ^2	2.0
α_3	-1.0	$\omega_{11}^2 = \omega_{22}^2$	4.0
α_4	-0.05	ω_{12}^2	2.0

Table 6.2 – Parameters for the model structure 3ci.

Finally, a set of choice data for a three option case with the following structure has been considered:

$$\begin{aligned}
 V_1 &= \alpha_1 + \alpha_3 X_{11} + \alpha_4 X_{21} + \alpha_5 X_{31} \\
 V_2 &= \alpha_3 X_{12} + \alpha_4 X_{22} + \alpha_5 X_{32} \\
 V_3 &= \alpha_2 + \alpha_3 X_{13} + \alpha_4 X_{23} + \alpha_5 X_{33}
 \end{aligned}
 \quad
 \Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & 0 \\ \sigma_{12}^2 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{33}^2 \end{bmatrix}
 \quad (6.45)$$

$$\Omega_{(1)} = \begin{bmatrix} \omega_{11}^2 & \omega_{12}^2 \\ \omega_{21}^2 & \omega_{22}^2 \end{bmatrix}$$

and the choice data have been generated using the true parameters reported in table 6.3. This case is referred to in the following as 3cb.

Parameter	Value	Parameter	Value
α_1	-2.0	σ_{12}^2	0.8
α_2	-1.0	σ_{22}^2	2
α_3	-1.0	σ_{33}^2	2
α_4	-0.05	ω_{11}^2	2.9
α_5	-0.25	ω_{12}^2	1.7
σ_{11}^2	2.5	ω_{22}^2	4.5

Table 6.3 – Parameters for the model structure 3cb.

Two, similar, four options cases have been included in the tests.

First, a model referred to as 4ci, with 3 attributes entering the systematic utility of each option according to the following structure has been used:

$$\begin{aligned}
 V_1 &= \alpha_1 + \alpha_4 X_{11} + \alpha_5 X_{21} + \alpha_6 X_{31} \\
 V_2 &= \alpha_4 X_{12} + \alpha_5 X_{22} + \alpha_6 X_{32} \\
 V_3 &= \alpha_2 + \alpha_4 X_{13} + \alpha_5 X_{23} + \alpha_6 X_{33} \\
 V_4 &= \alpha_3 + \alpha_4 X_{14} + \alpha_5 X_{24} + \alpha_6 X_{34}
 \end{aligned}
 \quad
 \Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix}
 \quad (6.46)$$

$$\Omega_{(1)} = \begin{bmatrix} \omega_{11}^2 & \omega_{12}^2 & \omega_{13}^2 \\ \omega_{21}^2 & \omega_{22}^2 & \omega_{32}^2 \\ \omega_{31}^2 & \omega_{32}^2 & \omega_{33}^2 \end{bmatrix}$$

The ten sets of choice data have been generated using the parameters listed in table 6.4.

Parameter	Value	Parameter	Value
α_1	-2.0	α_6	-0.25
α_2	-1.0	σ^2	1.0
α_3	-1.5	$\omega_{11}^2 = \omega_{22}^2 = \omega_{33}^2$	4.0
α_4	-1.0	$\omega_{12}^2 = \omega_{13}^2 = \omega_{23}^2$	2.0
α_5	-0.05	-	-

Table 6.4 – Parameters for the model structure 4ci

Then a structure similar to the previous one but with option's utilities partially correlated has been used:

$$\begin{aligned}
 V_1 &= \alpha_1 + \alpha_4 X_{11} + \alpha_5 X_{21} + \alpha_6 X_{31} \\
 V_2 &= \alpha_4 X_{12} + \alpha_5 X_{22} + \alpha_6 X_{32} \\
 V_3 &= \alpha_2 + \alpha_4 X_{13} + \alpha_5 X_{23} + \alpha_6 X_{33} \\
 V_4 &= \alpha_3 + \alpha_4 X_{14} + \alpha_5 X_{24} + \alpha_6 X_{34}
 \end{aligned}
 \quad
 \Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & 0 & 0 \\ \sigma_{12}^2 & \sigma_{22}^2 & 0 & 0 \\ 0 & 0 & \sigma_{33}^2 & \sigma_{34}^2 \\ 0 & 0 & \sigma_{34}^2 & \sigma_{44}^2 \end{bmatrix}
 \quad (6.47)$$

$$\Omega_{(1)} = \begin{bmatrix} \omega_{11}^2 & \omega_{12}^2 & \omega_{13}^2 \\ \omega_{21}^2 & \omega_{22}^2 & \omega_{32}^2 \\ \omega_{31}^2 & \omega_{32}^2 & \omega_{33}^2 \end{bmatrix}$$

The true parameters used in the ten instances of choice data generated for this structure are reported in table 6.5.

Parameter	Value	Parameter	Value
α_1	-2.0	σ_{12}^2	1.0
α_2	-1.0	$\sigma_{33}^2 = \sigma_{44}^2$	2.5
α_3	-1.5	σ_{34}^2	1.2
α_4	-1.0	ω_{11}^2	2.0
α_5	-0.05	$\omega_{22}^2 = \omega_{33}^2$	4.5
α_6	-0.25	$\omega_{12}^2 = \omega_{13}^2$	1.0
$\sigma_{11}^2 = \sigma_{22}^2$	2	ω_{23}^2	3.2

Table 6.5 – Parameters for the model structure 4cb

6.5.4 Calculation Time Results

The calculation times required by algorithms employing analytical and numerical derivatives to perform the calibration of the models described in the previous section using subsets of 500, 1000, 2500, 5000, 7500, 10000 choice data have been recorded to obtain confirmation that using the analytical differentiation technique explained earlier in the chapter is actually less computationally expensive than using numerical differentiation.

Figures 6.1-6.5 show clearly the important difference of the calculation times employed by the Mendell-Elston procedure when the gradient at each iteration is calculated analytically and when it is calculated numerically.

The times reported in fig 6.2 and 6.3 and in fig 6.4 and 6.5 are rather similar for corresponding differentiation method and sample size as was expected considering the similarity of the problems and the corresponding number of calibrated parameters.

The more than linear increase in calculation time showed in fig 6.4 for the numerical differentiation procedure in cases with 10000 observations is due to difficulties with

the calculations of the derivatives very close to the solution that, however, occurred very rarely. These problems were never encountered when using analytical differentiation.

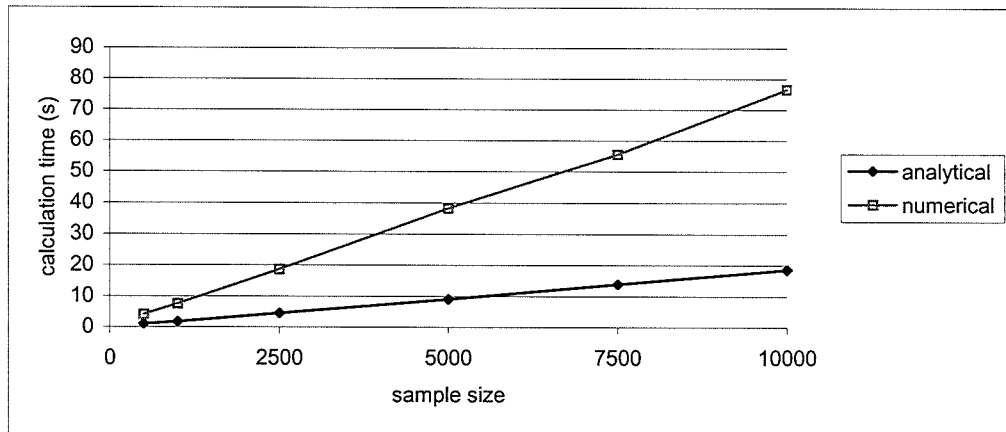


Fig 6.1 - Comparison of calculation times for the model structure 3ai.

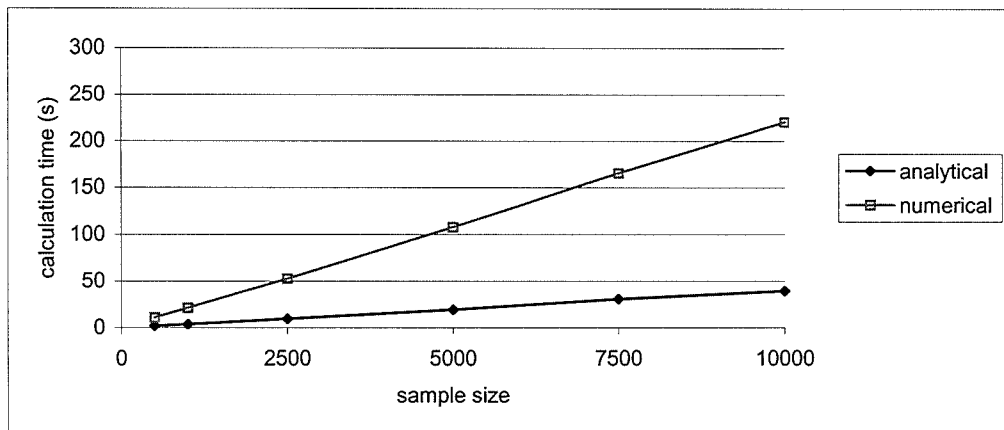


Fig 6.2 - Comparison of calculation times for the model structure 3ci.

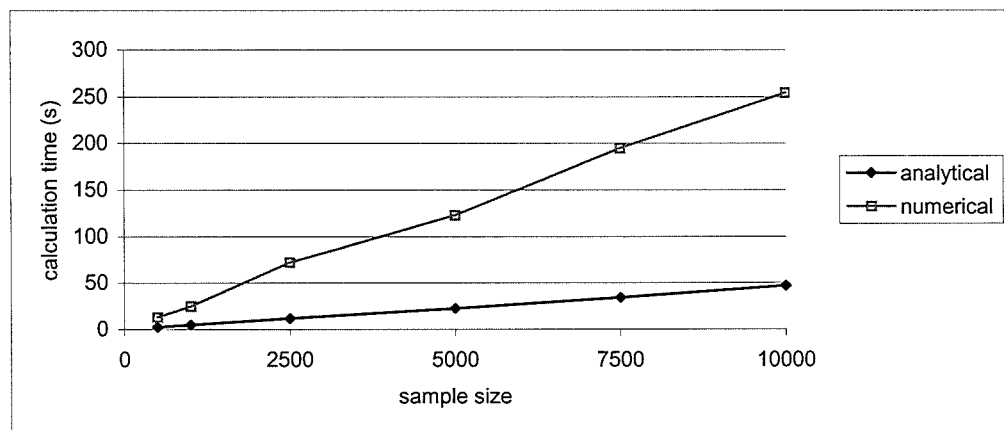


Fig 6.3 - Comparison of calculation times for the model structure 3cb.

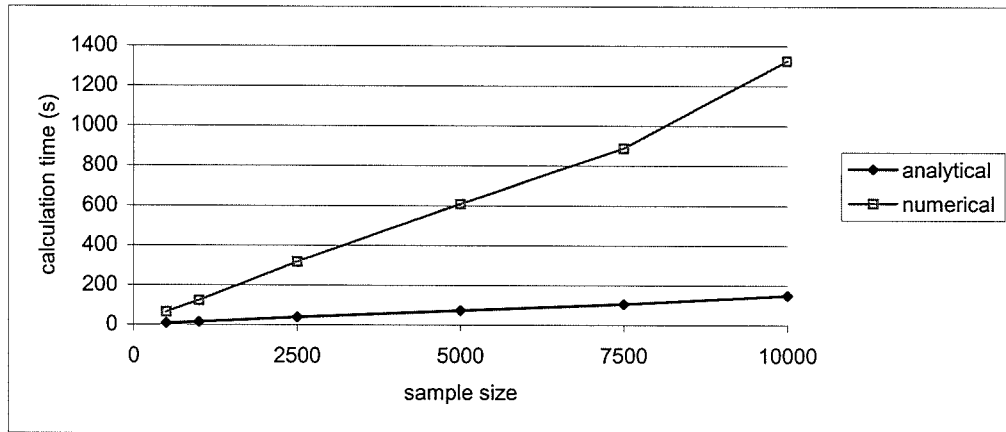


Fig 6.4 - Comparison of calculation times for the model structure 4ci.

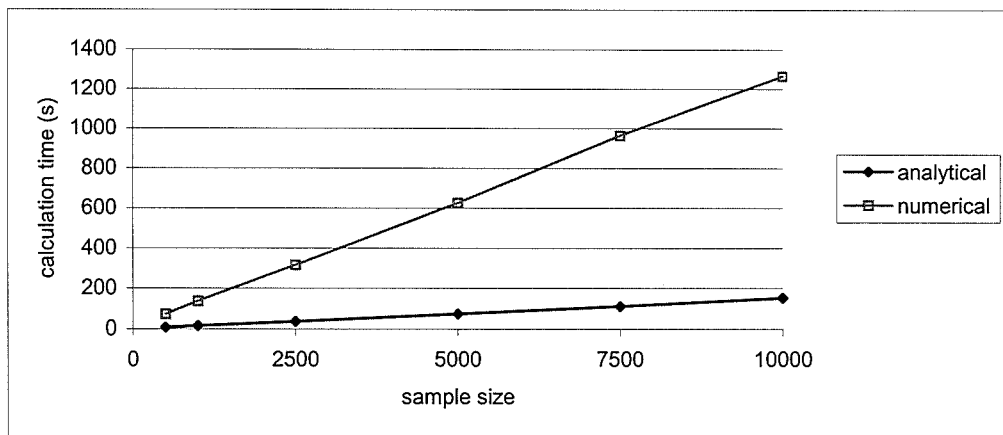


Fig 6.5 - Comparison of calculation times for the model structure 4cb.

6.5.5 Parameter Retrieval Results

The results on parameter retrieval are presented by model structure. It should be remarked that the results obtained by solving the log-likelihood maximisation with analytical and numerical derivatives are consistent, as should be expected of a correct procedure, thus only the results obtained from one of the procedures are presented.

The results reported, such as those in figure 6.6, refer to the average quality of the parameter estimates obtained from the ten calibrations performed for each model structure and sample size. In fact, fig. 6.6 and the following similar ones report the mean of the estimates and the extremes of its 95% confidence interval from the 10

calibration exercises against the sample size used. The confidence intervals are obtained by assuming that the estimates of the parameters are Normally distributed. This is, asymptotically, the case for the coefficients in the systematic utility but not for those in the covariance matrix that are obtained as the product of Cholesky factors directly estimated. Testing the Normality of the covariance matrix entries estimates for each sample size was not totally conclusive but did not exclude the Normality assumption. The relevant graphs reported should therefore be intended as indicative and resulting from an approximate assumption. Also the lines connecting the markers for the data points are only indicative.

The parameters of the simple model 3ai are estimated in all cases with the correct sign. Observing figures 6.6-6.10 it is evident that the quality of the estimates, especially of those of the covariance matrix entries, improves when the sample size is 2500 or more. In fact, it should be particularly noted the lower accuracy in estimating the alternative specific constants and the covariance matrix entries for lower sample sizes.

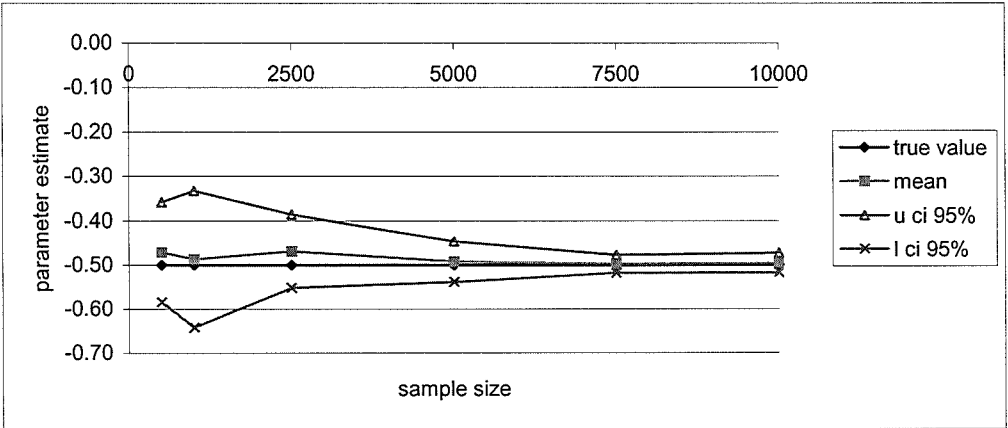


Fig. 6.6 - Estimates of alternative specific constant α_1 and true value against sample size for model 3ai.

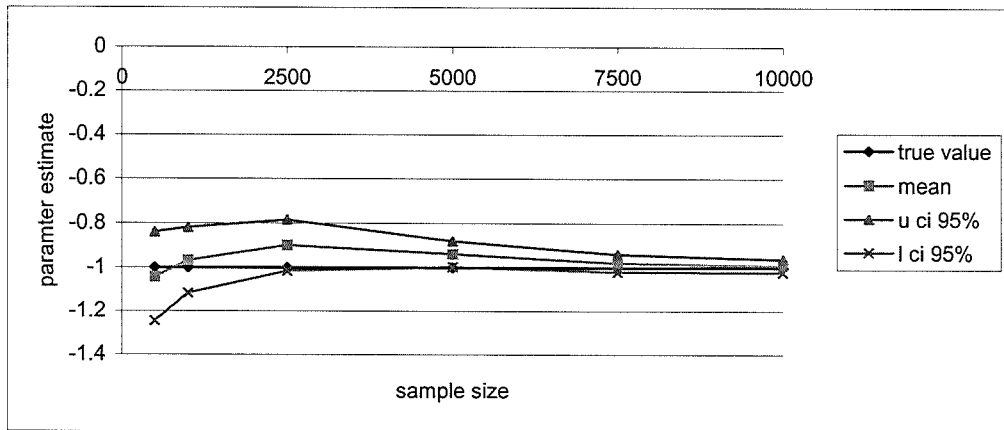


Fig. 6.7- Estimates of alternative specific constant α_2 and true value against sample size for model 3ai.

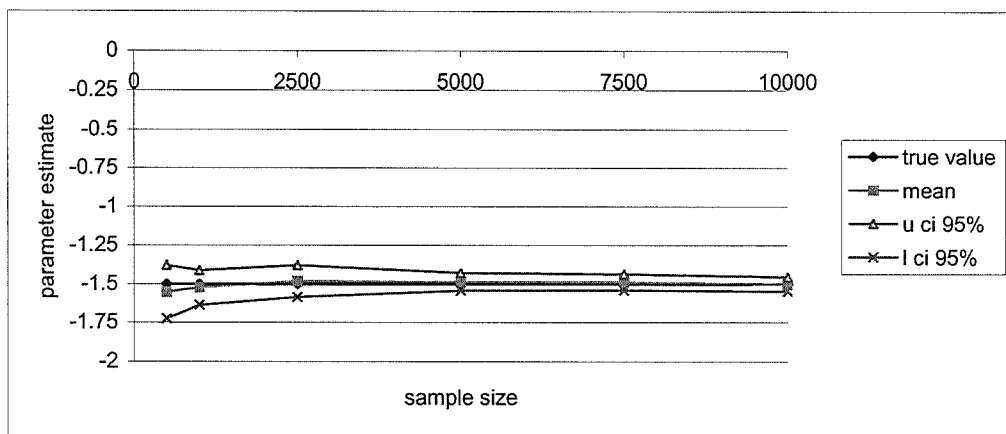


Fig. 6.8 – Estimates of coefficient α_3 (for the only attribute) and true value against sample size for model 3ai.

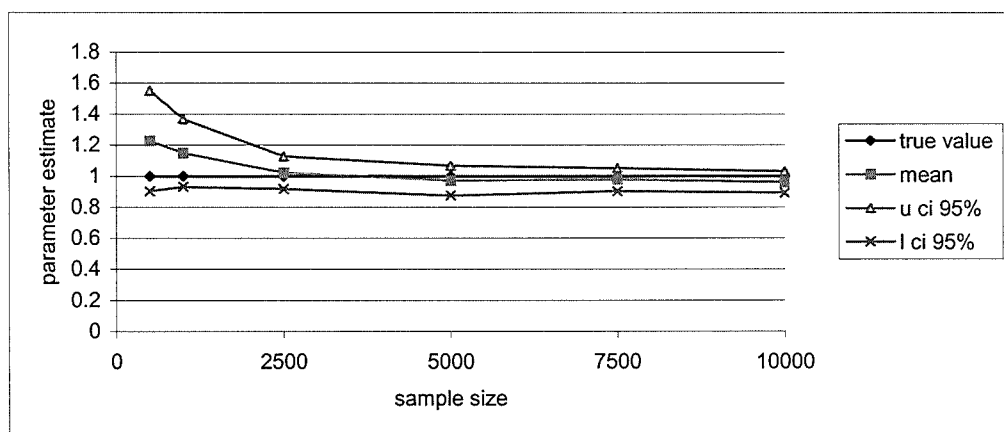


Fig. 6.9 – Estimates of entry ω_{12}^2 of the reference covariance matrix and true value against sample size for model 3ai.

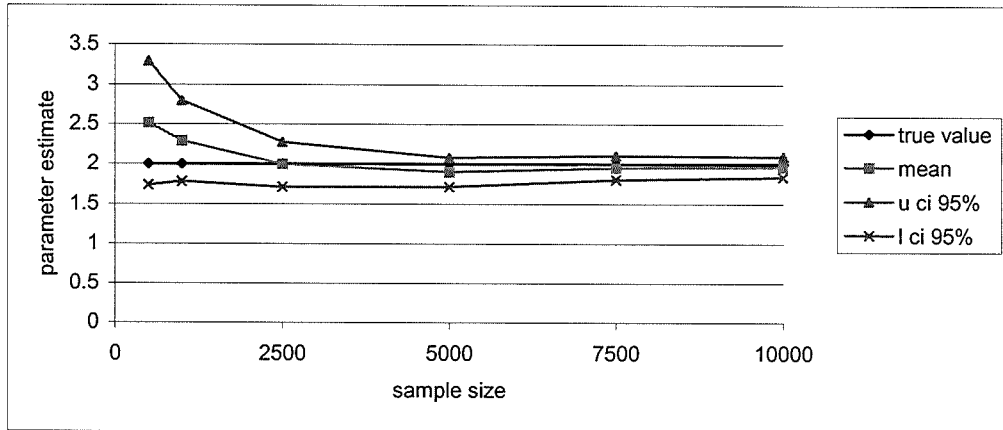


Fig 6.10 – Estimates of entry ω_{22}^2 of the reference covariance matrix and true value against sample size for model 3ai.

Figure 6.11-6.17 show that also in the 3ci case the quality of the results improves when the sample size exceeds 2500, although in all cases the parameters are recovered with the correct sign. The estimates of the alternative specific constants do not always include the true value in the 95% confidence interval but they are close estimates of them. A similar consideration is valid for two of the estimates of the other parameters in the systematic utility, reported in figs. 6.13-6.15. The small confidence interval width reported in the figures for the coefficients of the attributes suggests, as seems to be the case also in the other examples, that they are easier to estimate correctly.

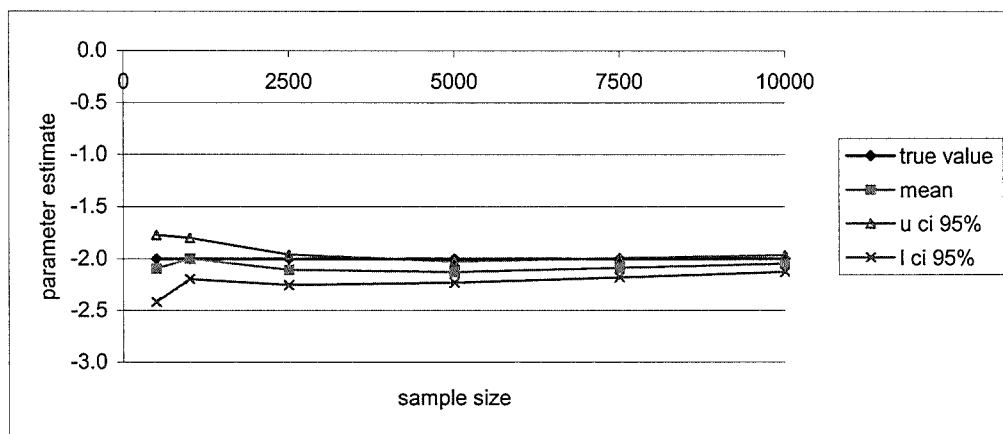


Fig. 6.11 - Estimates of alternative specific constant α_1 and true value against sample size for model 3ci.

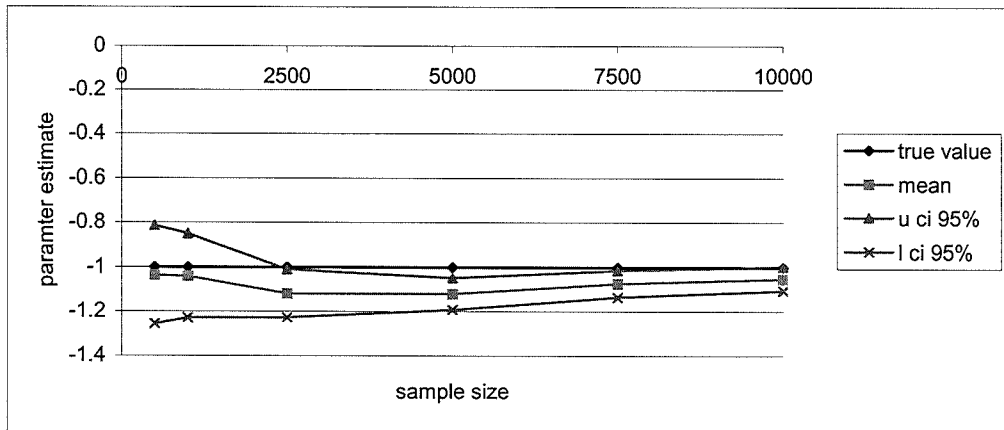


Fig. 6.12 - Estimates of alternative specific constant α_2 and true value against sample size for model 3ci.

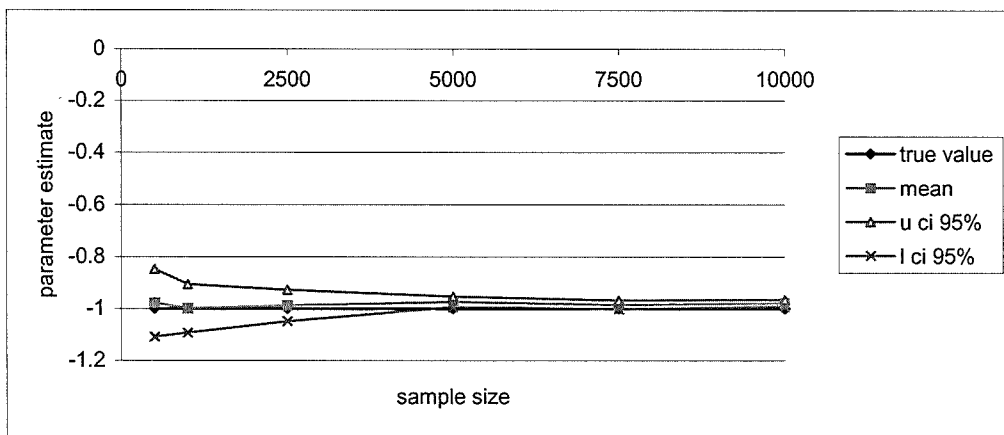


Fig. 6.13 – Estimates of coefficient α_3 (for the first attribute) and true value against sample size for model 3ci.

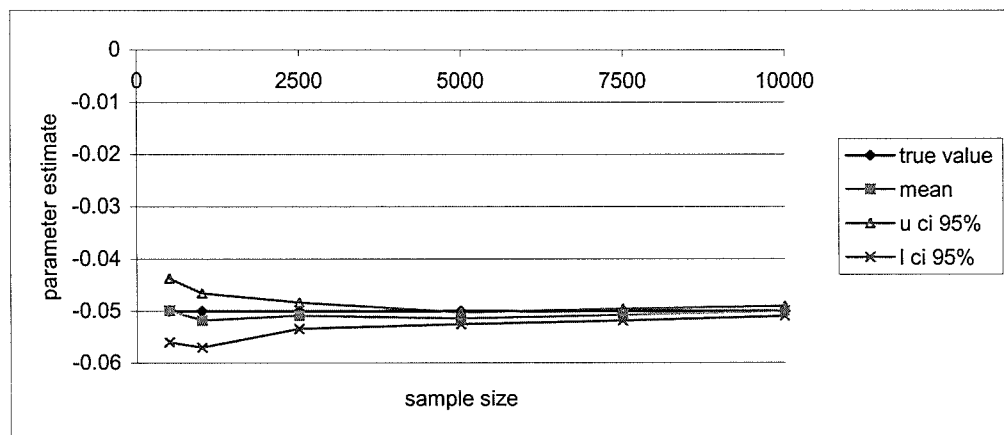


Fig 6.14 – Estimates of coefficient α_4 (for the second attribute) and true value against sample size for model 3ci.

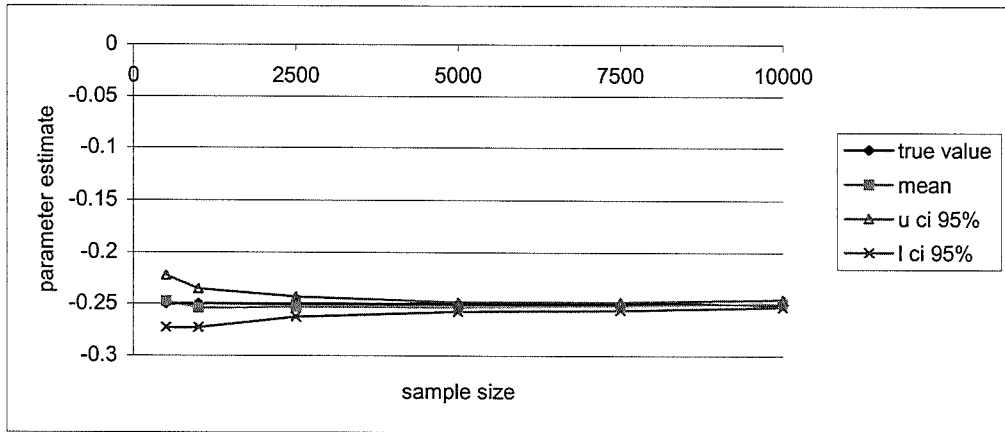


Fig 6.15 – Estimates of coefficient α_5 (for the third attribute) and true value against sample size for model 3ci.

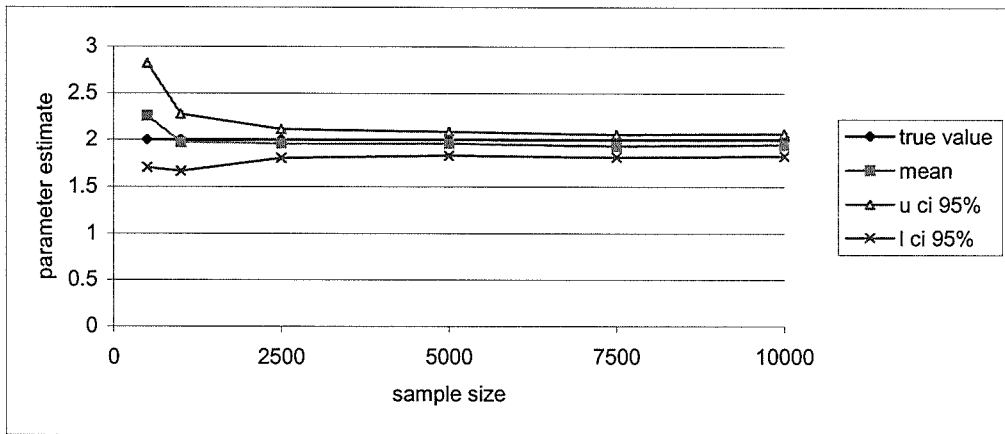


Fig. 6.16 – Estimates of entry ω_{12}^2 of the reference covariance matrix and true value against sample size for model 3ci.

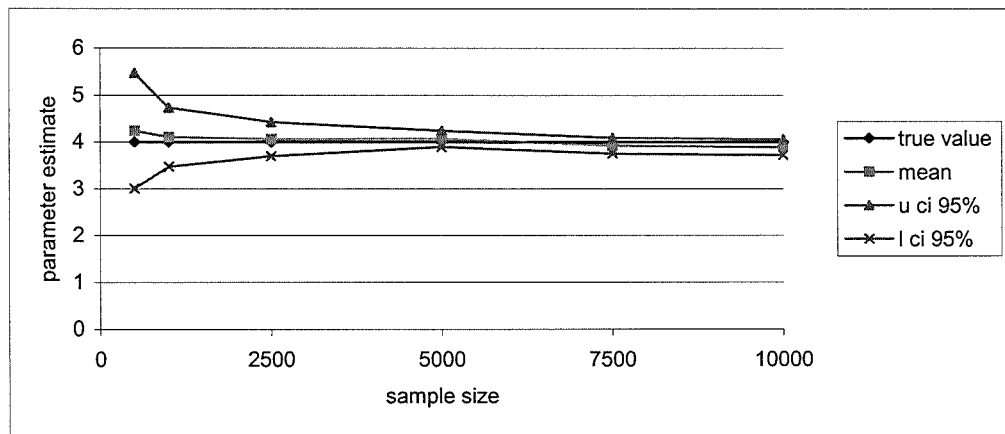


Fig 6.17 – Estimates of entry ω_{22}^2 of the reference covariance matrix and true value against sample size for model 3ci.

Figures 6.18-6.24 show the mean and the 95% confidence interval of the mean of the estimates from the ten calibrations for each sample size for model structure 3cb.

The sign of the parameters is correctly estimated except in one case with sample size 500 when the off-diagonal covariance matrix entry is incorrectly estimated as negative. The 95% confidence intervals include the true values of the parameters but their widths suggest the difficulty of estimating the model with the smaller sample sizes. In particular, as in the previous cases, the estimates are consistently of better quality when the sample size exceeds 2500.

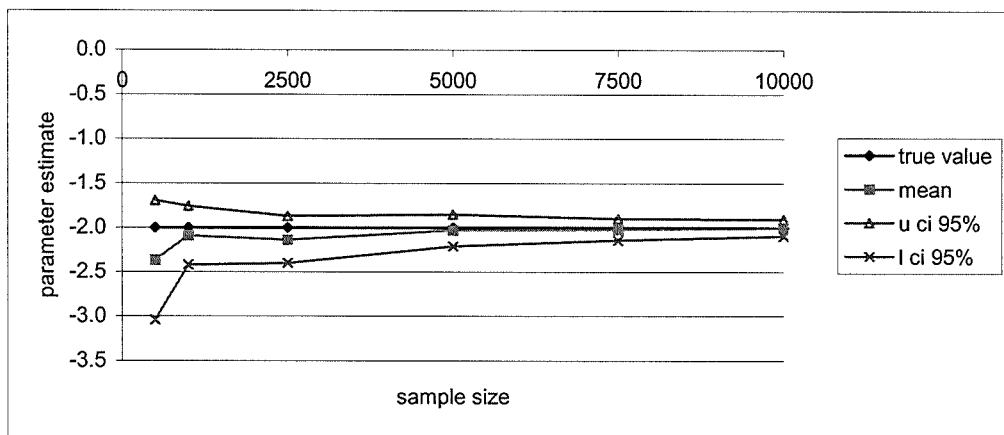


Fig. 6.18 - Estimates of alternative specific constant α_1 and true value against sample size for model 3cb.

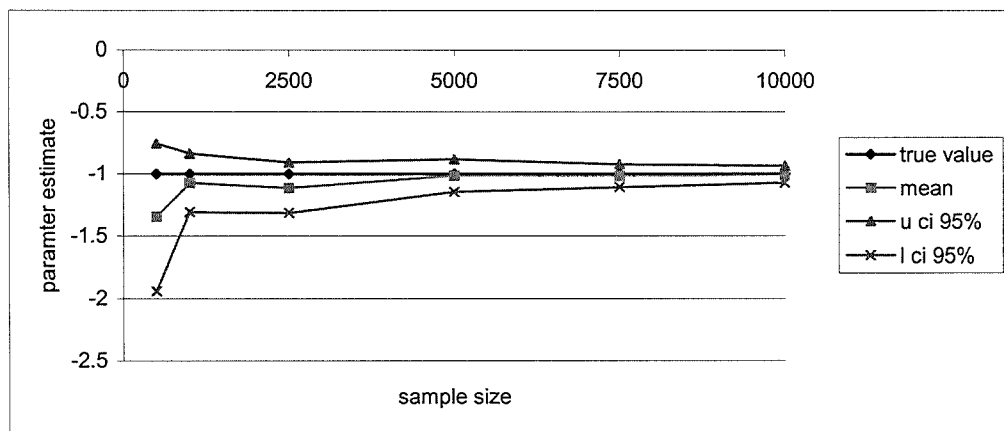


Fig. 6.19 - Estimates of alternative specific constant α_2 and true value against sample size for model 3cb.

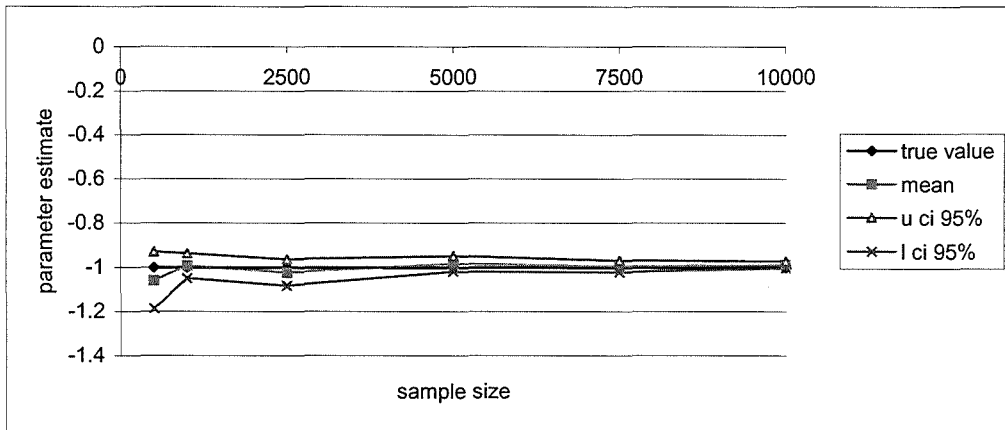


Fig. 6.20 – Estimates of coefficient α_3 (for the first attribute) and true value against sample size for model 3cb.

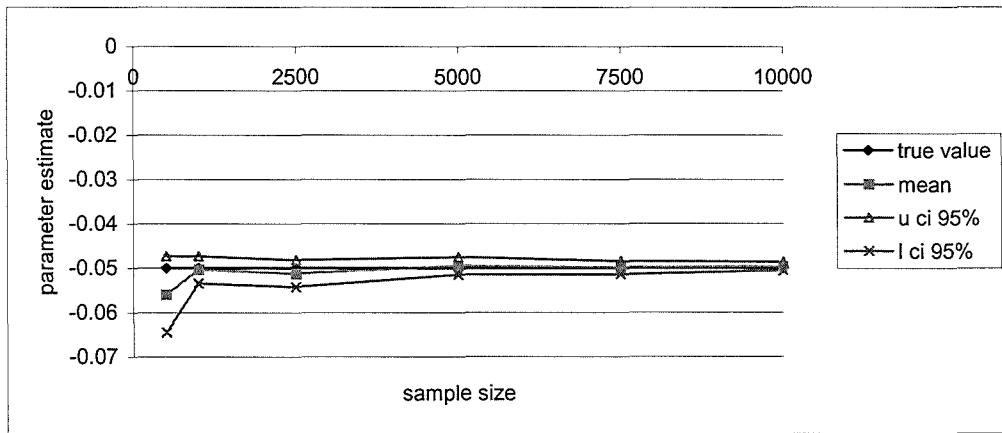


Fig 6.21 – Estimates of coefficient α_4 (for the second attribute) and true value against sample size for model 3cb.

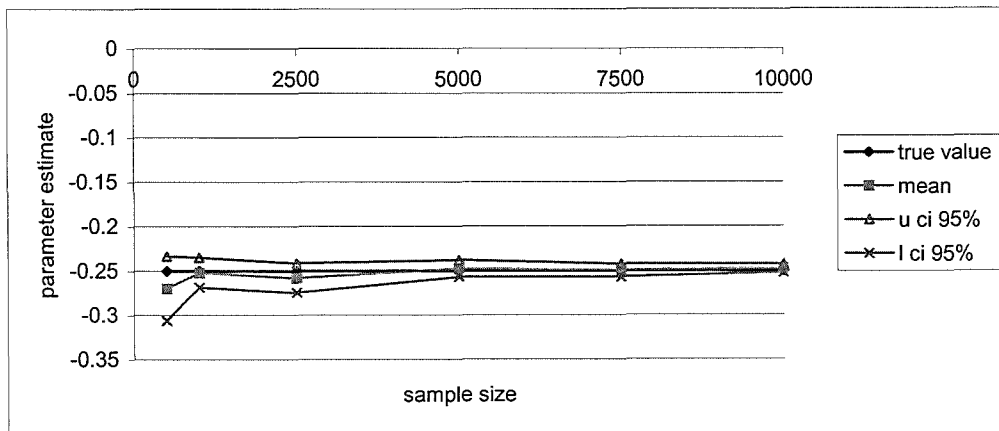


Fig 6.22 – Estimates of coefficient α_5 (for third attribute) and true value against sample size for model 3cb.

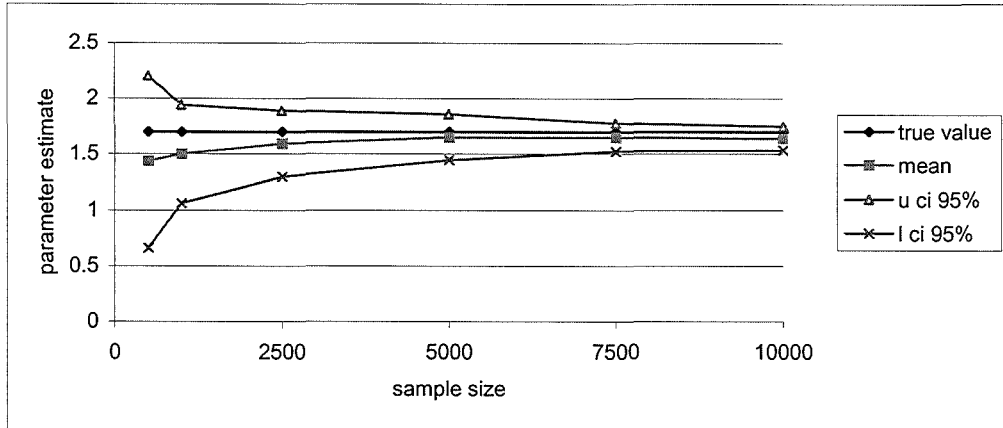


Fig. 6.23 – Estimates of entry ω_{12}^2 of the reference covariance matrix and true value against sample size for model 3cb.

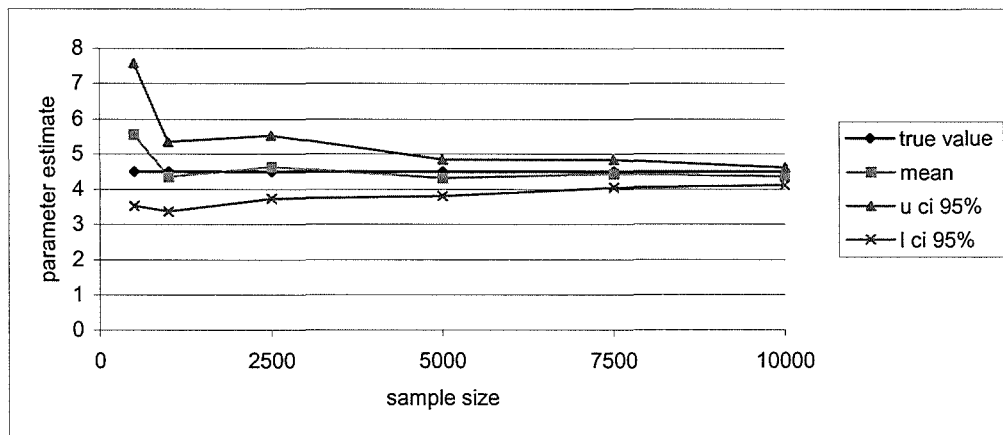


Fig 6.24 – Estimates of entry ω_{22}^2 of the reference covariance matrix and true value against sample size for model 3cb.

Figures 6.25-6.36 show that, on average, the calibration program estimates correctly the magnitude of the coefficients for the 4ci model, although with sample sizes of 500 and 1000 the confidence intervals are particularly large for both alternative specific constants and entries of the reference covariance matrix. This is also due to estimates of the wrong sign in a small number of cases with 500 and 1000 observations. The width of the confidence interval of the estimates seems to vary very little in all cases with sample sizes of 2500 and over. It is interesting to notice the difficulty of estimating the covariance matrix entries with sample sizes of less than 5000. On the other hand, even with small sample sizes that cause a large variation of the estimates of the parameters retrieved, the estimates of the attribute coefficients are rather good.

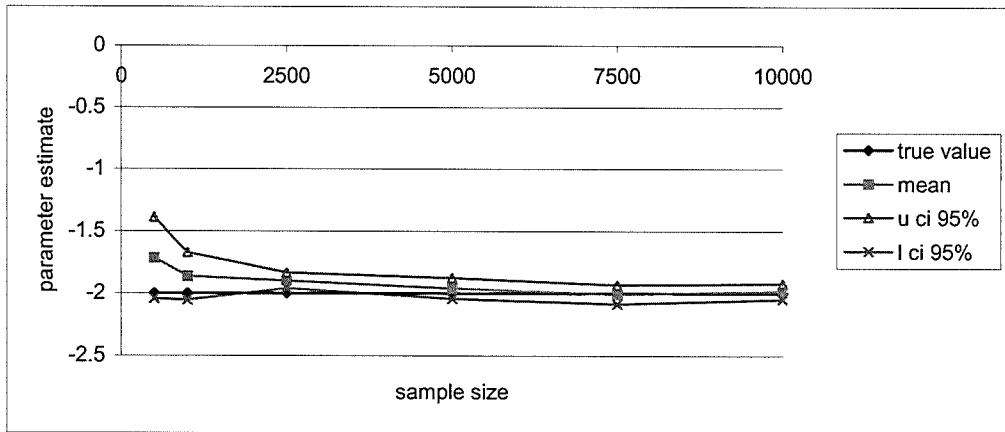


Fig 6.25 – Estimates of alternative specific constant α_1 and true value against sample size for model 4ci.

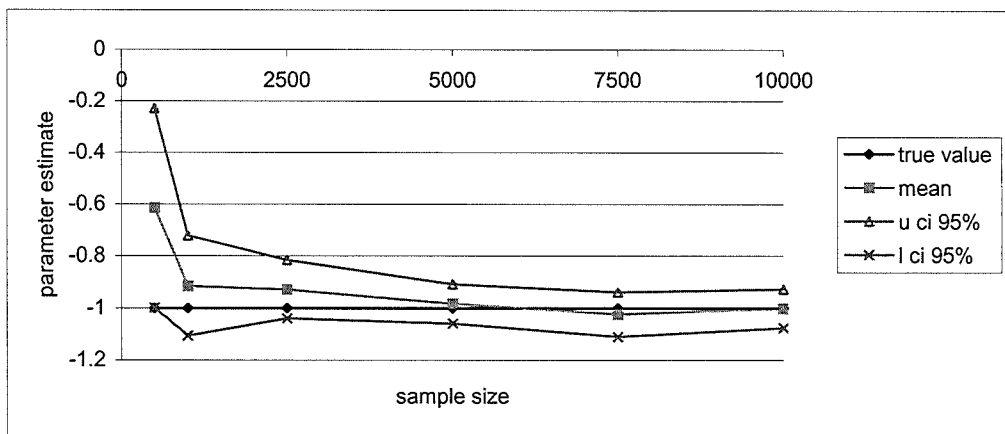


Fig 6.26 – Estimates of alternative specific constant α_2 and true value against sample size for model 4ci.

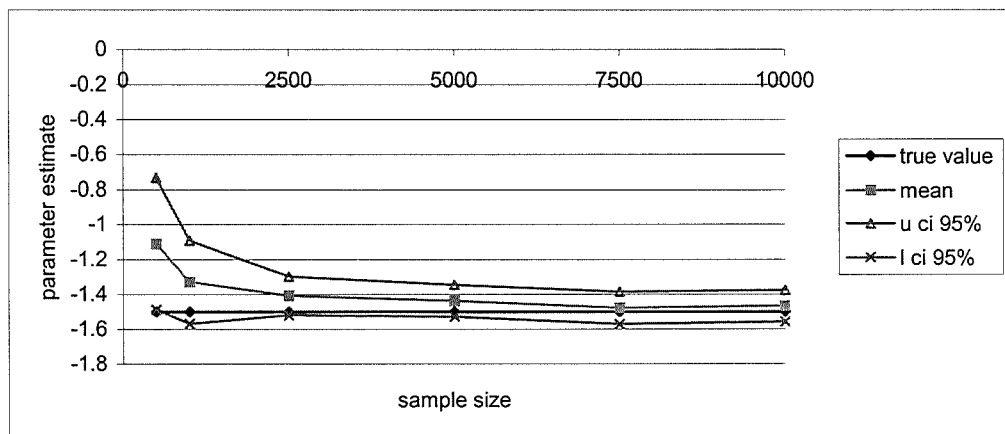


Fig 6.27 – Estimates of alternative specific constant α_3 and true value against sample size for model 4ci.

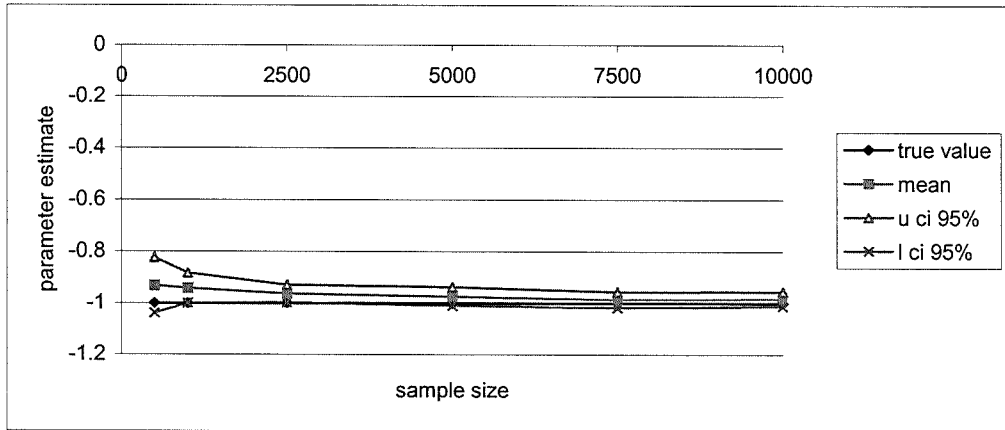


Fig 6.28 – Estimates of coefficient α_4 (for first attribute) and true value against sample size for model 4ci.

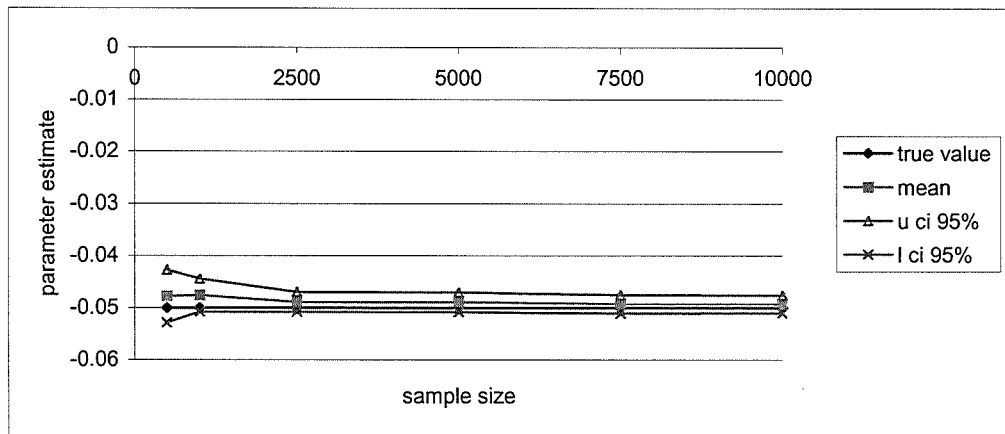


Fig 6.29 – Estimates of coefficient α_5 (for second attribute) and true value against sample size for model 4ci.

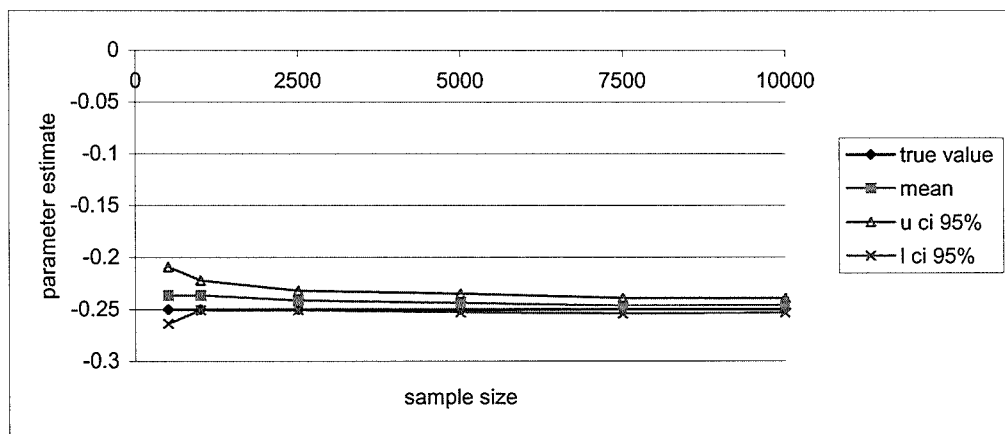


Fig 6.30 – Estimates of coefficient α_6 (for third attribute) and true value against sample size for model 4ci.

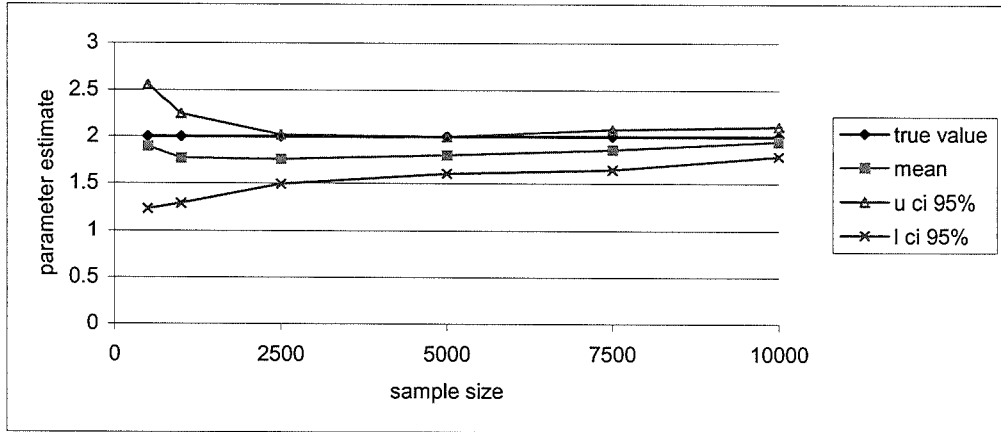


Fig 6.31 – Estimates of entry ω_{12}^2 of the reference covariance matrix and true value against sample size for model 4ci.

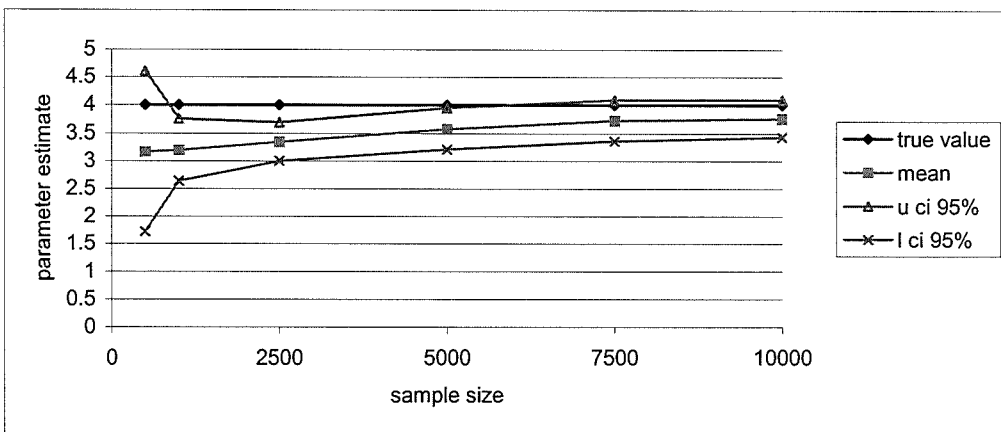


Fig 6.32 – Estimates of entry ω_{22}^2 of the reference covariance matrix and true value against sample size for model 4ci.

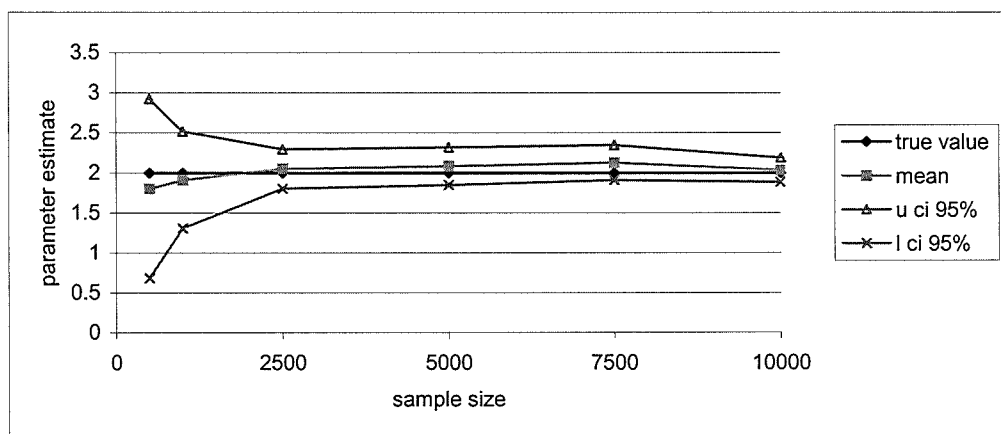


Fig 6.34 – Estimates of entry ω_{13}^2 of the reference covariance matrix and true value against sample size for model 4ci.

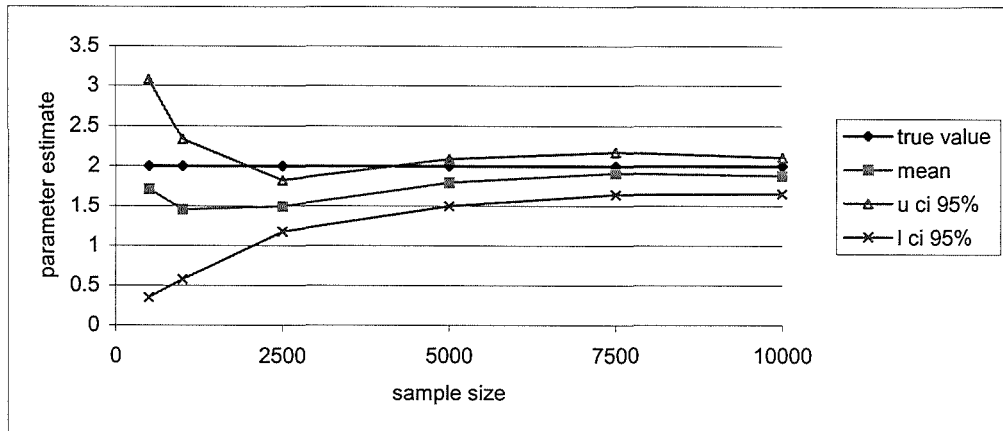


Fig 6.35 – Estimates of entry ω_{23}^2 of the reference covariance matrix and true value against sample size for model 4ci.

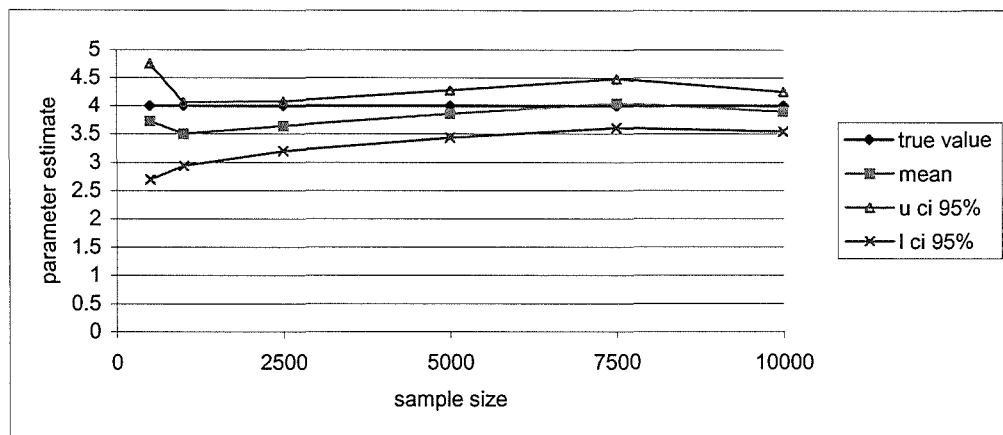


Fig 6.36 – Estimates of entry ω_{33}^2 of the reference covariance matrix and true value against sample size for model 4ci.

The results for the case 4cb, that differs from the previous one only for the covariance matrix, are reported in fig. 6.37-6.47 and confirm the difficulties of estimating the model with sample sizes not exceeding 2500 when all the possible coefficients are left free to vary. In particular, the estimates of the elements of the reference covariance matrix obtained are of very poor quality with small sample sizes and, although they improve with larger sample sizes, their precision seems to remain limited. An increase in quality of the estimate could probably be achieved by introducing restrictions to the covariance matrix (even imposing simply $\omega_{12}^2 = \omega_{13}^2$, through the corresponding Cholesky factors).

In some cases with sample size 500 and 1000, the off diagonal entries of the reference covariance matrix, and in particular ω_{12}^2 and ω_{13}^2 are estimated as negative (which implies a negative correlation between the corresponding variates). These results are clearly incorrect, and need further investigation, but do not introduce negative elements in the other covariance matrices in difference and are not encountered with larger sample sizes.

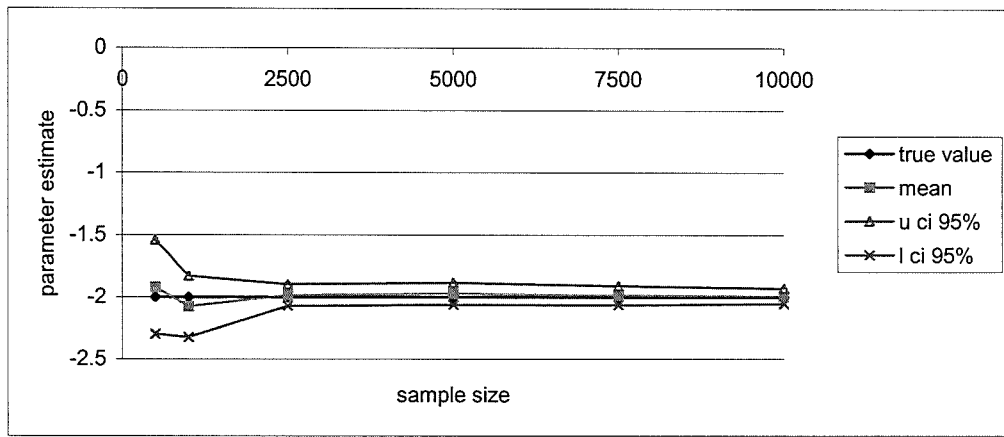


Fig 6.37 – Estimates of alternative specific constant α_1 and true value against sample size for model 4cb.

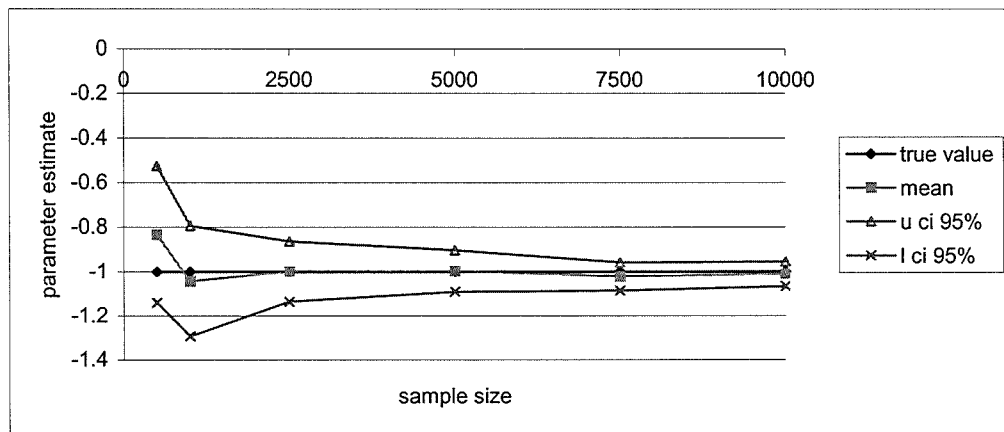


Fig 6.38 – Estimates of alternative specific constant α_2 and true value against sample size for model 4cb.

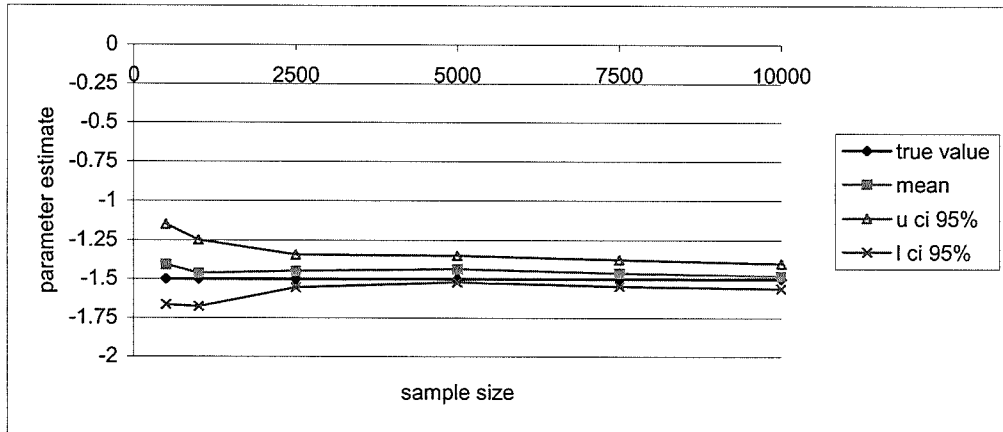


Fig 6.39 – Estimates of alternative specific constant α_3 and true value against sample size for model 4cb.

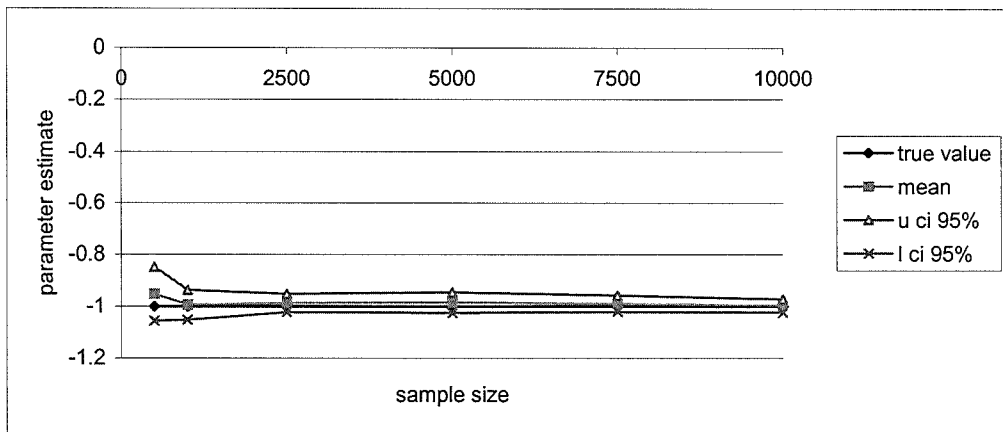


Fig 6.40 – Estimates of coefficient α_4 (for the first attribute) and true value against sample size for model 4cb.

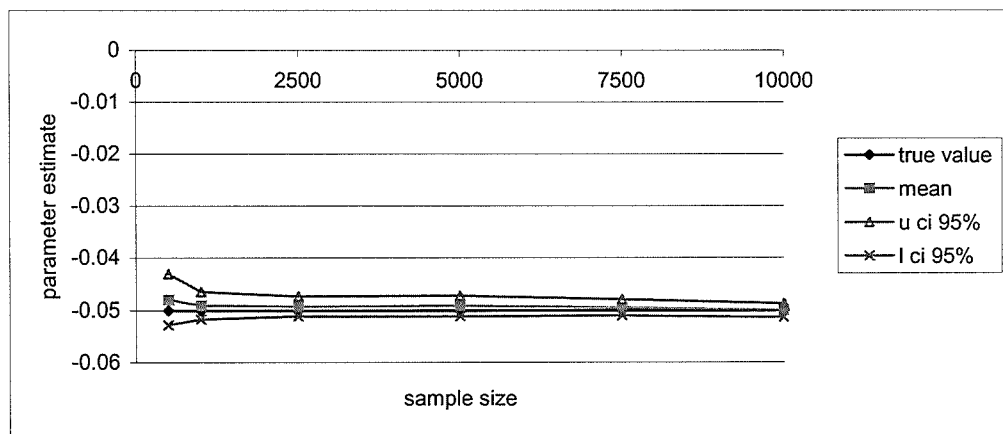


Fig 6.41 – Estimates of coefficient α_5 (for the second attribute) and true value against sample size for model 4cb.

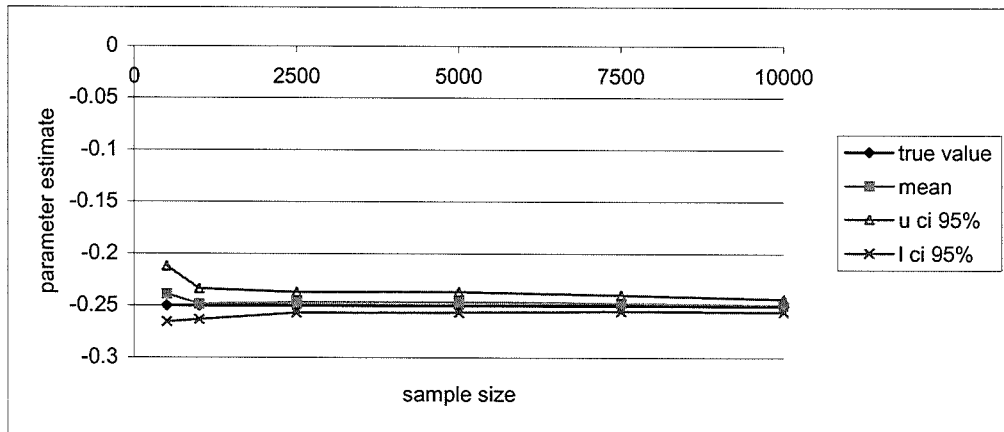


Fig 6.42 – Estimates of coefficient α_6 (for the third attribute) and true value against sample size for model 4cb.

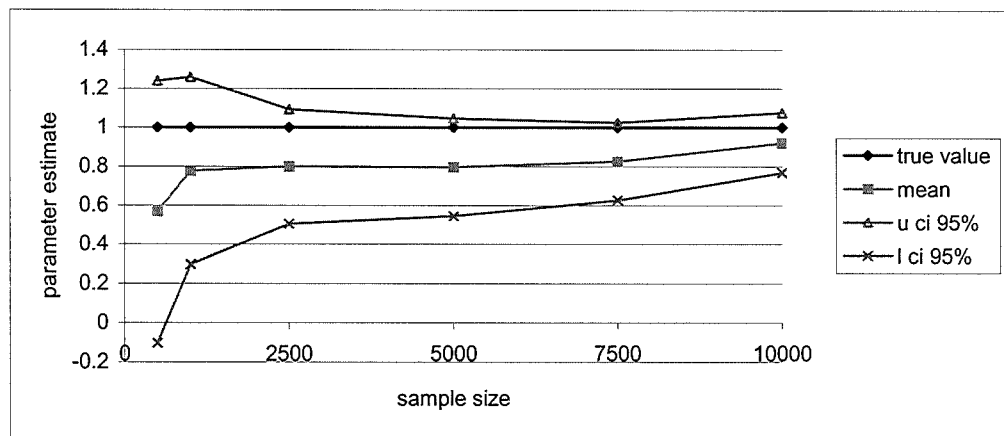


Fig 6.43 – Estimates of entry ω_{12}^2 of the reference covariance matrix and true value against sample size for model 4cb.

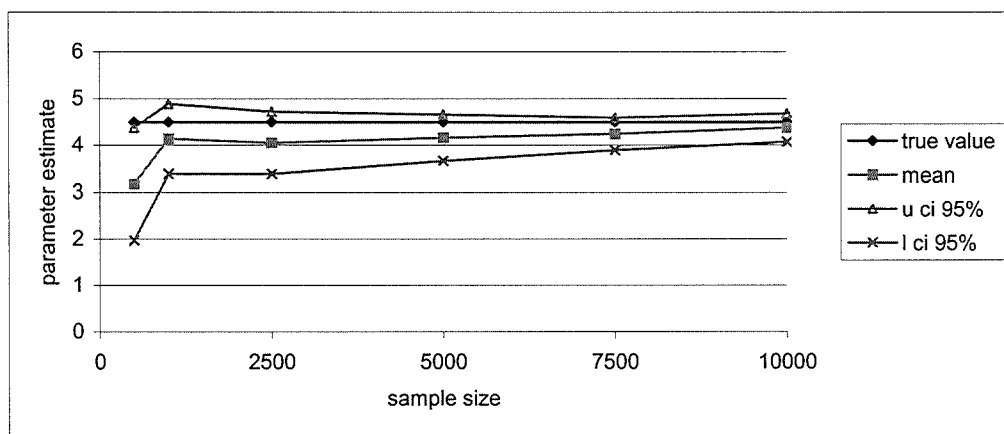


Fig 6.44 – Estimates of entry ω_{22}^2 of the reference covariance matrix and true value against sample size for model 4cb.

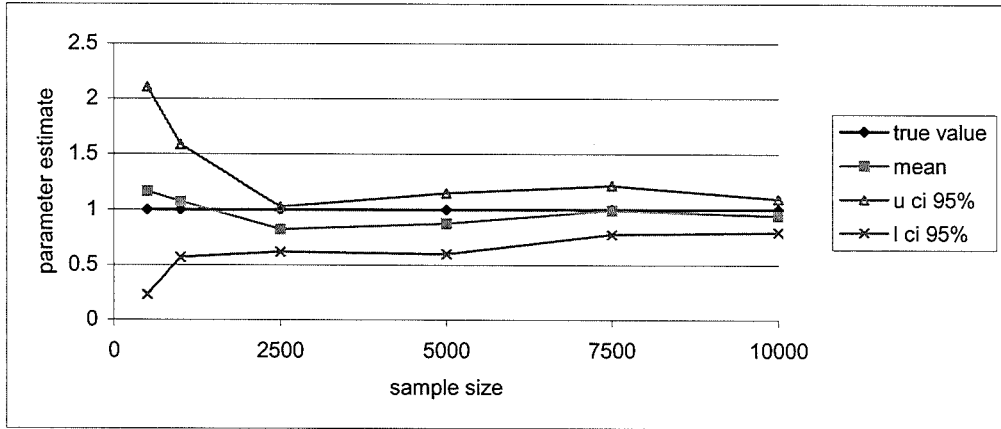


Fig 6.45 – Estimates of entry ω_{13}^2 of the reference covariance matrix and true value against sample size for model 4cb.

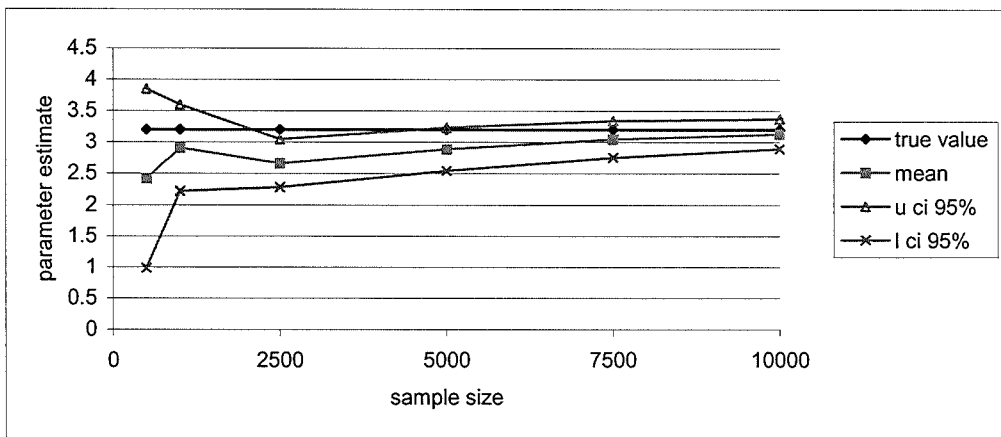


Fig 6.46 – Estimates of entry ω_{23}^2 of the reference covariance matrix and true value against sample size for model 4cb.

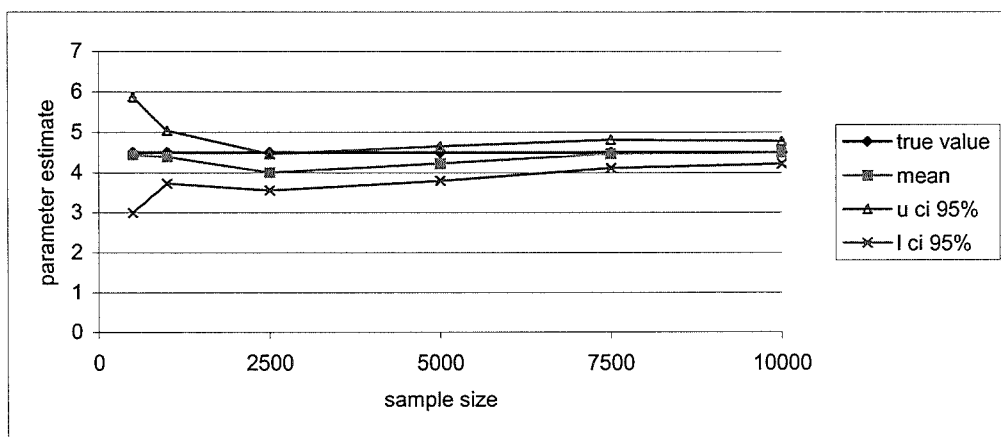


Fig 6.47 – Estimates of entry ω_{33}^2 of the reference covariance matrix and true value against sample size for model 4cb.

6.5.6 Conclusions

The data presented, although not obtained from an extensive series of tests, confirm the efficiency of using analytical derivatives for calibrating multinomial probit choice models solved with the Mendell-Elston approximation.

The calibration results reported refer to average estimates from ten calibrations for samples of different sizes and aimed at suggesting the possible confidence intervals of the estimates that can be expected when all the model's parameters are left free to vary. An analyst calibrating a model will probably have an initial idea of the sign and of the possible magnitude of the coefficients but here the systematic utility coefficients have been started from a value of zero and the entries of the reference covariance matrix in difference have been left free to vary to check to what extent the model can retrieve them.

It was generally observed, in the cases with 3 and 4 options calibrated, that samples with 2500 data or less give, on average, estimates of poor quality especially for the alternative specific constants and for the covariance matrix entries. The quality of the estimates improved with larger sample sizes. The need to use large sample sizes to obtain results of good quality is in agreement with previous findings on models with 3 options where the utility covariance matrix was estimated (Maher *et al.*, 1999) and with other results reported in the literature on probit model calibration: for instance, Keane (1992) used a sample size of 8000 for a small trinomial probit example, Munizaga and Ortúzar (1997) had to use a sample size of 8000 to obtain correctly a correlation, the only parameter they calibrated in the utility covariance matrix of a model with four alternatives, although they needed only 2000 data when they calibrated only a variance. Similar smaller sample sizes were used in other cases *e.g.* Bolduc (1999) used a sample of 1299 observation but estimated a model with a structured covariance matrix.

Investigations introducing restrictions on the covariance matrix estimated were not carried out here since the main objective of the chapter is the introduction of the analytical differentiation procedure, although it is likely that they would result in improved precision of the estimates for the same sample sizes. However, the calibration results obtained let us conclude that the flexibility of the multinomial

probit model comes at the cost of needing large samples of data to calibrate it if all parameters are free to vary and the analyst is trying to use the model to reveal the structure of the choice situation.

6.6 Conclusions

This chapter was concerned with the calibration of multinomial probit choice models when an analytical approximation method is used to solve the choice function. The calibration problem has been introduced by recalling its presentation given in chapter two and the likelihood method to calibrate a choice model has been described. The introduction to the problem has included a review of the issues related to the specification of the models to be estimated, focussing in particular on the specification of the covariance matrix and on the techniques to check it.

After recalling the basic formulae of the Mendell-Elston approximation, a method developed for calculating analytically the derivatives of the Mendell-Elston MNP probabilities and log-likelihood function has been described. It exploits the structure of the Mendell-Elston approximation and can be extended to other approximations with similar calculation structure.

The correctness and the efficiency of the differentiation method developed have been tested by implementing it in a computer program that has been used to calibrate a small number of artificial test models of known characteristics with 3 or 4 choice options.

Comparing the calculation time data with those obtained with the same program when it uses numerical derivatives confirmed the efficiency of using analytical differentiation. The correspondence of the derivatives at a point in the solution space and of the final results confirmed the correctness of the analytical differentiation procedure.

The same test calibrations were used to obtain a picture of the dependence of the accuracy of the results on the sample size suggesting the need to use large sample sizes for calibrating a multinomial probit model with all possible parameters free to vary.

The work presented here could be developed in several ways. For instance the calibration program written could be extended to consider normalised probabilities and other approximations as well as to include the numerical integration method of Genz (1992, 1993), used in chapter 3 to obtain reference data, and to deal with more choice options. To save on computational effort, the extension to consider normalised probabilities could probably solve a first approximation of the problem without the normalisation of the probabilities and use it only to pinpoint the solution.

Moreover, experiments on retrieval accuracy similar to those proposed here could be carried out also with larger choice sets and on a larger range of model structures, building a database of information about the precision with which results can be obtained. This work could be used to study the problem of optimising the design of the sample of data for a required accuracy of the estimates, also when the sample is not simply random but is organised in strata.

One more issue that can be studied using an extension of the program employed in this chapter is the possible multimodality of the likelihood of multinomial probit models. As it is difficult to obtain general results on this matter, an extensive series of experiments on different series of model structures could be used to investigate numerically the existence of multiple log-likelihood optima also trying to characterise the causes of such multiple optima. Very little work in the literature (Daganzo, 1979; Sparmann *et al.*, 1983; Liu and Mahmassani, 2000) is concerned with this problem and none offer a general comprehensive treatment of it. The possible multimodality of the MNP log-likelihood function is often disregarded in practice but is important to make sure that the parameters resulting from a model calibration are not different from the real maximum likelihood ones, otherwise the resulting model is inherently incorrect.

7. SUMMARY AND CONCLUSIONS

This chapter closes the thesis by summarising the main points on the work and the results presented and suggesting further work that could be carried out on the subject.

The **first chapter** introduced transportation models and their application for the evaluation of changes of transport systems. It was remarked that many phenomena studied in transportation are the result of the choices of many individuals and that, as a consequence, transportation models are often applications of choice models.

The importance of using a sound choice model for obtaining reliable results from a transportation model was underlined and it was stated the intention to study the application to traffic assignment and model calibration of analytical solutions to one such model, the multinomial probit choice model, that is well known but not widely used.

The multinomial probit choice model and the main alternative choice models based on the random utility framework used in transportation were reviewed in the **second chapter**. This survey, especially focussed on models for mode choice and traffic assignment, showed that, although the best known and simplest models like the multinomial logit and the hierarchical logit are widely used, much recent research work has tried to extend the flexibility and the applicability of the models available.

Two strands of work in this direction can be characterised in the literature. One is based on extending the choice situations that can be analysed with models whose choice function is written in closed form and that are related to the logit model. Examples of this type of new models are those obtained as particular instances of the GEV model (*e.g.* the paired combinatorial logit and the cross-nested logit model) or of the mother logit model (as the C-logit model). These models have been applied to

traffic assignment to obviate the well known inability of the multinomial logit model to account correctly for network topology.

The second strand of published work, that has appeared very recently, focuses on the development and the application of the mixed logit model, that has been shown to be able to approximate all the other choice models, but can only be solved by simulation, using procedures initially developed for the multinomial probit model.

The discussion in chapter two recalled that the mixed logit model has an advantage over the multinomial probit only when non-Normal random effects need to be modelled. It also remarked that the new models developed in the first stream of work characterised in the literature try to extend the capabilities of closed form models to match those already available in the multinomial probit model, whose choice function cannot be written in closed form.

Thus, the idea of working on analytical approximations to the multinomial probit model can be seen as a way of obtaining the same result sought in the work in the literature on new closed form models (a flexible choice model that can be solved analytically) starting from a different point, that is trying to solve a flexible model that cannot be written in closed form with an approximate, but analytical, formulation. Since the work in this thesis is mainly concerned with traffic assignment applications, it was also remarked that having a choice model solved analytically is convenient also as it allows us to develop and employ efficient solution algorithms such as those discussed in chapters four and five, devoted to traffic assignment. The use of analytical multinomial probit methods is also convenient for the calibration problem as it obviates the need to choose the number of simulation replications.

In fact, the analytical approach to the solution of the multinomial probit choice function is not new and the most notable attempt at it has been the introduction of the Clark approximation in the late 1970s. The more recent limited interest enjoyed by such approach in the transportation literature is, perhaps, due to doubts about the precision of the approximations after the limited precision of the method of Clark was noted in the literature.

Therefore, the first step to pursue the analytical treatment of the multinomial probit model has been to compare the precision and the computing cost of a number of alternative analytical methods including that of Clark. The results of this work have been reported in **chapter three** where it was explained that the multinomial probit choice function can be solved directly or as an equivalent multivariate Normal integral. The latter problem recurs in several disciplines and analytical solution techniques borrowed from some of them have been compared with others already used in transportation on a series of artificial case studies devised to replicate traffic assignment choice situations. The comparisons showed that the likely precision varies with the methods used although all give less precise results for actual low choice probabilities. Comparing the precision and the calculation time it was suggested that the analytical solution of the multinomial probit model with good accuracy is feasible and the best methods to carry it out are two of those approximating the multivariate Normal integral: the approximation of Mendell-Elston and the approximation of Tang and Melchers, the first originally developed in biostatistics and the second initially put forward for structural reliability applications.

The results in chapter three can be used for choosing the best method to employ in traffic assignment but are not necessarily readily transferable to cases where only the MVN integral is used, as they have been obtained by normalising the single probabilities to their sum.

One interesting general point from the results presented is the importance of the processing order of the variates within an approximation method for obtaining accurate results. In fact, the approximations examined are, in all cases, iterative applications of a basic approximation that is employed as many times as the dimension of the problem requires and the order in which the variates are included in the approximation seriously influences the results. The tests for the Taylor approximation also showed that refining the elemental approximation is less effective than optimising the calculation order. Therefore should other

approximations be considered for practical use it is important that their optimal calculation order, if any can be devised, is characterised.

Chapter three was mainly concerned with the viability of analytical approximations different from numerical integration but a further important point resulting from it is that the use of numerical integration in transportation applications is worth pursuing further. The numerical integration method of Genz, that was used to obtain reference data, could be employed directly by requiring a smaller precision of the results. Although also with this setting it will probably remain more time consuming than most approximation methods, it overcomes the problems that limited the application of previously published integration methods to cases with up to 4 options.

In closing the comments on the work reported in chapter three, it should be said that, although the comparisons were based on an extensive set of choice situations, it could be interesting to extend them further to include comparisons of the final results of traffic assignment models on real networks and of the final results of calibration algorithms, also considering results obtained from the probit solution methods by simulation currently mostly used.

The **fourth chapter** investigated different algorithms for the solution of the Stochastic User Equilibrium traffic assignment model. Two multinomial probit approximation methods were used in the implementation of the algorithms but the same algorithms can be used also with other analytical approximations and choice models solved analytically. In fact, they are based on the direct use of the value of the SUE objective function of Sheffi and Powell (1982) and of its gradient, thus requiring those quantities to be evaluated correctly and not from simulation.

The investigations on different ways to determine the step length along the MSA search direction showed that there is little scope for improvement on the interpolation methods in the literature (that approximate the function as a cubic or a quadratic along the search direction) by using similar methods that employ information taken at different points but requiring the same number of stochastic loadings. The interesting results obtained using predetermined steps different from

the MSA were undermined by the possibility of different performances on different networks whereas methods obtaining the steps by interpolation should adapt to the problem at hand. Any of the interpolation methods studied can be used in a practical algorithm and none of them should probably be expected to perform systematically better than the others.

When the possibility of using different search directions was considered to obtain more efficient algorithms, the interpretation of the MSA search direction as a preconditioned steepest descent direction for the objective function of Sheffi and Powell was put forward. Building on this way of reading the traditional search direction, two preconditioned conjugate gradient algorithms, based respectively on the formula of Fletcher and Reeves and on the formula of Polak and Ribiere, were proposed and tested.

Both algorithms improved on the corresponding ones using the traditional search direction but only the algorithm of Polak and Ribiere proved to be robust when used with a line search refined only to obtain an improved objective function gradient, which is generally the most efficient strategy. Although the problems showed by the Fletcher and Reeves algorithm deserve further analysis, it had been expected from the literature on optimisation that the method of Polak and Ribiere would prove to be more suitable for non-quadratic problems.

Possible further work extending that presented in chapter four could involve analysing the performance of those algorithms on larger networks and when the set of paths is built by column generation, rather than being given at the outset of the assignment, and is therefore not fixed until close to the solution.

The effectiveness of the preconditioned conjugate gradient search directions, is an interesting result also as it underlines the importance of looking at the objective function of the problem at hand to devise more effective algorithms. Only a limited investigation on the shape of the objective function has been carried out by observing its Hessian matrix and has been reported before testing the algorithms, and more investigations on this subject in a larger number of test cases would be interesting also to understand better the effect of using the traditional search

direction at the beginning of the problem calculations, when, as in the cases examined, the Hessian matrix of the SUE objective function is not positive definite.

However, the results reported suggested that the original problem is ill-conditioned and can be simplified by a preconditioning like the one inspired by the traditional search direction.

The simple preconditioning given by the Jacobian of the cost functions can be used also to develop further preconditioned algorithms such as quasi-Newton and limited memory quasi-Newton algorithms. With these types of algorithms, however, it would not be possible to keep track of the current solution in path flow terms as the simple relationship between search direction in path and link flow terms exploited in this thesis would not hold. They could be used for link based algorithms. However, since the Fletcher-Reeves preconditioned steepest descent algorithms encountered difficulties when not applied with a precise determination of the step length or not started with a number of MSA iterations, it is likely that a quasi-Newton algorithm would encounter the same difficulties in the same circumstances. Limited memory quasi-Newton algorithms, that are typically restarted after one or few iterations, should obviate these difficulties but would not take full advantage of the information they gather when they are closer to the solution, where the objective function seems to be better behaved.

The experiments in **chapter five** confirmed the results on the performance of the algorithms investigated for the SUE problem also for the multiple user class extension, where the algorithms actually solve a SUE problem in terms of standardised flows, and for the more complex MUC SUE ED problem, where the algorithms work again on a problem in terms of standardised flows but this is made more difficult by the possible variation of the demand.

The new objective functions for the SUE ED and MUC SUE ED cases, proposed in chapter five, simplify noticeably the SUE ED formulation in the literature and the MUC SUE ED formulation that can be directly taken from it. It is interesting that, thanks to those formulations, the extensions of the SUE problem studied can be

reduced to a problem very similar to SUE, both in terms of formulation and of algorithms and ultimately that they can be reduced to a problem in terms of link flows. Moreover, the formulations proposed allowed a better understanding of the programmes by the analysis of their Hessian matrix which showed that the programmes for the extended problems keep the properties of the corresponding SUE ones.

Objective functions for other extensions of the SUE problem with the same cost structure (separable link costs and additive path costs) could be devised and analysed similarly by starting from the idea of a gradient reducing to the difference between present and auxiliary (balanced) link flows. Given the results obtained in chapter four and five, it could be expected that algorithms similar to those explored in this thesis applied to further extensions of the SUE problem formulated as those examined in chapter five, would perform similarly.

Further work to support the practical application of the probit solution to elastic demand assignment could involve the satisfaction function, or more generally of the distribution of the minimum of the perceived costs. The methods suggested at the end of chapter three to determine the satisfaction include a way to obtain it consistently with the Mendell-Elston approximation but a better understanding of the possible distribution of the minimum of the perceived costs would be of interest.

The work in **chapter six** considered the problem of calibrating a multinomial probit model solved analytically by focussing on the use of the Mendell-Elston approximation. To develop an efficient calibration procedure, the chapter presented an algorithm for obtaining analytical derivatives exploiting the structure of the Mendell-Elston approximation calculations. The method was coded in a program that can alternatively use numerical derivatives and the feasibility and efficiency of calculating analytical derivatives with the method developed was shown by comparing the calculation times required by the two differentiation techniques in a number of test cases. It was also remarked that a differentiation procedure similar to the one used in the Mendell-Elston case can be applied with other approximations.

The calibration results from the same test cases, built by generating artificial data from known models, showed that the maximum likelihood program developed can capture the real parameters and structure of the models but the quality of the results depends, as expected, on the size of the sample of data used.

Further work on multinomial probit calibration can be suggested building on that presented in chapter six: for instance, the implementation of other multinomial probit solution methods and the comparison of the results they would give on a series of test cases, to extend to the calibration problem the accuracy tests reported in chapter three. Similar numerical investigations extended to a large number of model structures and sample sizes could be used to develop a database of information on parameter retrievability which would show merits and limitations of the model and could provide helpful information for survey design. The possible multimodality of the multinomial probit likelihood function, an issue often disregarded in the literature and difficult to investigate analytically, could also be explored with an extensive series of numerical tests carried out calibrating a number of models from different starting points that would provide guidance on what model structures may bring about this issue.

REFERENCES

Abramowitz M. and Stegun I.A. (1972) *Handbook of Mathematical Functions*. National Bureau of Standards.

Akamatsu T. (1996) Cyclic flows, Markov process and stochastic traffic assignment. *Transportation Research B*, **30**(5), 369-386.

Alvarez M. and Nagler J. (1994) *Correlated Disturbances in Discrete Choice Models: a Comparison of Multinomial Probit Models and Logit Models*. Political Methodology Working Paper. University of California at Riverside.

Alvarez, M., Nagler J. and Willette J. (1999) *Measuring the Relative Impact of Issues and the Economy in Democratic Elections*. Social Science Working Paper 1052. California Institute of Technology.

Andrews R.D. and Langdon M.G. (1976) *An Individual Cost Minimising Method of Determining Modal Split Between Three Travel Modes*. TRRL Laboratory Report 698, Transport and Road Research Laboratory, Crowthorne, Berkshire.

Beckmann M., McGuire C.B. and Winsten C.B. (1956) *Studies in the Economics of Transportation*. Cowles Commission Monograph, Yale University Press, New Haven.

Bekhor S. and Prashker J.N. (1999) Formulation of extended logit stochastic user equilibrium assignment. In: *Proceedings of the 14th International Symposium on Transportation and Traffic Theory* (ed. A. Ceder), Pergamon, Oxford, 351-372.

Bell M.G.H. (1995) Alternatives to Dial's logit assignment algorithm. *Transportation Research B*, **29**(4), 287-295.

Bell M.G.H. and Cassir C. (1998) The use of the path flow estimator in multimodal networks. In: *Proceedings of the ICTTS98 Conference (International Conference on Traffic and Transportation Studies)*, Beijing.

Bell M.G.H. and Grosso S. (1998) The path flow estimator as a network observer. *Traffic Engineering and Control*, **39**(10), 540-549.

Bell M.G.H. and Iida Y. (1997) *Transportation Network Analysis*. Wiley, Chichester, England.

Bell M.G.H., Cassir C., Grosso S., Clement S.J. (1997) Path flow estimation in traffic system management. In: *Proceedings of the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems*, Chania, Greece, 1316-1321.

Bell M.G.H., Lam W.H.K. and Iida Y. (1996) A time-dependent path flow estimator. In: *Proceedings of 13th International Symposium on Transportation and Traffic Theory* (ed. J.B. Lesort), Elsevier Press, New York, 173-193.

Bell M.G.H., Lam W.H.K., Ploss G. and Inaudi D. (1993) Stochastic user equilibrium assignment and iterative balancing. *Proceedings of 12th International Symposium on Transportation and Traffic Theory* (ed. C.F. Daganzo), Elsevier Press, New York, 427-439.

Bell M.G.H., Shield C.M., Busch F. and Kruse G. (1997) A stochastic user equilibrium path flow estimator. *Transportation Research C*, **2**(3/4), 197-210.

Ben-Akiva M. and Bierlaire M. (1999). Discrete choice methods and their applications to short-term travel decisions. In: *Handbook of Transportation Science* (ed. R. Hall). Kluwer, Dordrecht, The Netherlands, 5-33.

Ben-Akiva M. and Lerman S.R. (1985) *Discrete Choice Analysis*. MIT Press, Cambridge, Massachusetts.

Ben-Akiva M., Bergman M.J., Daly A.J and Ramaswamy R. (1984) Modelling inter urban route choice behaviour, In: *Proceedings of the 9th International Symposium on Transportation and Traffic Theory*, VNU Science Press, 299-330.

Berntsen J., Espelid T.O. and Genz A. (1991) Algorithm 698: DCHURE - An adaptive multidimensional integration routine for a vector of integrals. *ACM Transactions on Mathematical Software*, **17**, 452-456.

Boffey T.B. (1982) *Graph Theory in Operations Research*. Macmillan. London.

Bolduc D. (1992) Generalized autoregressive errors in the multinomial probit model. *Transportation Research B*, **26**(2), 155-170.

Bolduc D. (1999) A practical technique to estimate multinomial probit models in transportation. *Transportation Research B*, **33**(1), 63-79.

Bolduc D., Lacroix G. and Muller C. (1996) The choice of medical providers in rural Benin: a comparison of discrete choice models. *Journal of Health Economics*, **15**, 477-498.

Borsch-Supan, A. and Hajivassiliou, V.A. (1993) Smooth unbiased multivariate probability simulators for maximum likelihood estimators of limited dependent variable models. *Journal of Econometrics*, **58**(3), 347-368.

Bouthelier F. and Daganzo C.F. (1979) Aggregation with multinomial probit and estimation of disaggregate models with aggregate data: a new methodological approach. *Transportation Research B*, **13**(2), 133-146.

Boyd J. and Melman R. (1980) The effect of fuel economy standards on the U.S. automotive market: an hedonic demand analysis. *Transportation Research A*, **14**(5-6), 367-378.

Brownstone D. and Train K. (1999) Forecasting new product penetration with flexible substitution patterns. *Journal of Econometrics*, **89**, 109-129.

Bunch D.S. (1991) Estimability in the multinomial probit model. *Transportation Research B*, **25**(1), 1-12.

Burrell J.E. (1968) Multiple route assignment and its application to capacity restraint. *4th International Symposium on the Theory of Traffic Flow*, Karlsruhe.

Cantarella G.E. (1997) A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. *Transportation Science*, **31**(2), 107-128.

Cantarella G.E. and Binetti M. (1998) Stochastic equilibrium traffic assignment with value-of-time distributed among users. *International Transactions in Operational Research*, **5**(6), 541-553.

Cantarella G.E., Cascetta E. Adamo V. and Astarita V. (1999) A doubly dynamic traffic assignment model for planning applications. *Proceedings of the 14th International Symposium on Transportation and Traffic Theory* (ed. A. Ceder), Pergamon, Oxford, 373-395.

Cardell N. and Dunbar F. (1980) Measuring the societal impacts of automobile downsizing. *Transportation Research A*, **14**(5-6), 423-434.

Carrese S., Gori S. and Picano T. (1997) A parking equilibrium model for park pricing and park&ride planning. In: *Proceedings of the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems*, Chania, Greece, 1305-1309.

Cascetta E., Nuzzolo A., Russo F. and Vitetta A. (1996) A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. In: *Proceedings of the 13th International Symposium on Transportation and Traffic Theory* (ed. J.B. Lesort), Elsevier Press, New York, 697-711.

Cascetta E., Russo F., Vitetta A. (1997) Stochastic User Equilibrium assignment with explicit path enumeration: comparison of models and algorithms. In:

Proceedings of the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems, Chania, Greece, 1078-1084.

Chen M. and Alfa A.S. (1991a) A network design algorithm using a stochastic incremental traffic assignment approach. *Transportation Science*, **25**(3), 215-224.

Chen M. and Alfa A.S. (1991b) Algorithms for solving Fisk's stochastic traffic assignment model. *Transportation Research B*, **25**(6), 405-412.

Chu C. (1981) A paired combinatorial logit model for travel demand analysis. In: *Proceedings of the Fifth World Conference on Transportation Research*, **4**, 295-309.

Clark C.E. (1961) The greatest of a finite set of random variables. *Operations Research*, **9**, 145-162.

Clark S.D. and Watling D.P. (2002) Sensitivity analysis of the probit-based stochastic user equilibrium model. *Transportation Research B*, **36**(7), 41-59.

Cox D.R. and Wermuth N. (1991) A simple approximation for bivariate and trivariate Normal integrals. *International Statistical Review*, **59**(2), 263-269.

Dafermos, S. (1980) Traffic equilibrium and variational inequalities. *Transportation Science*, **14**, 42-54.

Daganzo C.F. (1979) *Multinomial Probit. The Theory and its Application to Demand Forecasting*. Academic Press, New York.

Daganzo C.F. (1980) *Equilibrium Analysis in Transportation: the State of the Art and Some New Results*. Research Report UCB-ITS-RR-80-1, University of California, Berkeley, California.

Daganzo C.F. (1982) Unconstrained extremal formulation of some transportation equilibrium problems. *Transportation Science*, **16**(3), 332-360.

- Daganzo C.F. (1983) Stochastic network equilibrium with multiple vehicle types and asymmetric, indefinite link cost jacobians. *Transportation Science*, **17**(3), 282-300.
- Daganzo C.F. and Kusnic M. (1993) Two properties of the nested logit model. *Transportation Science*, **27**(4), 395-400.
- Daganzo C.F. and Sheffi Y. (1977) On stochastic models of traffic assignment. *Transportation Science*, **11**(3), 253-274.
- Daganzo C.F., Bouthelie F. and Sheffi Y. (1977) Multinomial probit and qualitative choice: a computationally efficient algorithm. *Transportation Science*, **11**(4), 338-358.
- Damberg O., Lundgren J.T., Patriksson M. (1996) An algorithm for the stochastic user equilibrium problem. *Transportation Research B*, **30**(2), 115-131.
- Dansie B.R. (1985) Parameter estimability in the multinomial probit model. *Transportation Research B*, **19**(6), 526-528.
- De La Barra T., Pérez B., Añez J. (1993) Multidimensional path search and assignment. In: *Proceedings of the 21st PTRC Summer Annual Meeting*, 307-319.
- Deák I. (1980) Three digit accurate multiple Normal probabilities. *Numerische Mathematik*, **35**, 369-380.
- Deák I. (1986) Computing probabilities of rectangles in case of multinormal distributions. *Journal of Statistical Computation and Simulation*, **26**, 101-114.
- Department of Transport Standing Advisory Committee on Trunk Road Appraisal (SACTRA) (1994) *Trunk Roads and the Generation of Traffic*, HMSO, London.
- Dial R.B. (1971) A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, **5**, 83-111.

Dial R.B. (1996) Bicriterion traffic assignment: basic theory and elementary algorithms. *Transportation Science*, **30**(2), 93-111.

Dijkstra E.W. (1959) Note on two problems in connection with graphs (spanning tree, shortest path). *Numerical Mathematics*, **1**(3), 269-271.

Donnelly T.G. (1973) Algorithm 462: bivariate Normal distribution. *Communications of the Association for Computing Machinery*, **16**, 638.

Drezner Z. (1992) Computation of the multivariate Normal integral. *ACM Transactions on Mathematical Software*, **18**(4), 470-480.

Emmerson P. (1992), Using elastic equilibrium for road schemes. In: *Proceedings of the 20th PTRC Summer Annual Meeting*, 77-90.

European Commission (1996) *Transport Research - APAS - Modelling of Urban Transport*. Office for Official Publications of the European Communities, Luxembourg.

Fisk C. (1980) Some developments in equilibrium traffic assignment methodology. *Transportation Research B*, **14**, 243-256.

Fletcher R. and Reeves C.M. (1964) Function minimisation by conjugate gradients. *Computer Journal*, **7**, 149-154.

Florian M. and Fox B. (1976) On the probabilistic origin of Dial's multipath traffic assignment model. *Transportation Research*, **10**(5), 339-341.

Frank M. and Wolfe P. (1956) An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3**(1-2), 95-110.

Frölich M., Heshmati A. and Lechner M. (2000) A microeconomic evaluation of long-term sickness in Sweden. Presented at the *European Evaluation Society Annual*

Conference, Lausanne. Available at: http://www.europeanevaluation.org/pdf/7-2_froelich.pdf.

Gabriel, S.A. and Bernstein D. (1997) The traffic equilibrium problem with nonadditive path costs. *Transportation Science*, **31**(4), 337-348.

Garrido R.A. and Mahamassani H.S. (2000) Forecasting freight transportation with the space-time multinomial probit model. *Transportation Research B*, **34**(5), 403-418.

Genz A. (1992) Numerical Computation of multivariate Normal probabilities. *Journal of Computational and Graphical Statistics*, **1**, 141-149.

Genz A. (1993) Comparisons of methods for the computation of multivariate normal probabilities. *Computing Science and Statistics*, **25**, 400-405.

Genz A. (2000) Personal communication.

Genz A. (2001) *Numerical Computation of Bivariate and Trivariate Normal Probabilities*. Department of Mathematics. Washington State University. USA.

Genz A. and Kwong K-S (2000) Numerical evaluation of singular multivariate Normal distributions. *Journal of Statistical Computation and Simulation*, **68**, 1-21.

Genz A. and Malik A.A. (1980) An adaptive algorithm for numerical integration over an n-dimensional rectangular region. *Journal of Computational and Applied Mathematics*, **6**(4), 295-302.

Geweke J., Keane M. and Runkle D. (1994) *Alternative Computational Approaches to Inference in the Multinomial Probit Model*. Research Department, Federal Reserve Bank of Minneapolis.

Gill P.E., Murray W. and Wright M.H. (1981) *Practical Optimization*. Academic Press, London.

Guo Q.L., Maher M.J. and Wamuziri S. (2001) Risk analysis in construction networks using a modified stochastic assignment model. *Journal of Civil Engineering and Environmental Systems*, **18**(3), 215-241.

Hall M.D., Fashole-Luke T., Van Vliet D. and Watling D.P. (1992) Demand-responsive assignment in SATURN. In: *Proceeding of the PTRC 20th Summer Annual Meeting, Seminar E*, 25-39.

Hart, Cheney E. W., Lawson C. L., Maehly H. J., Mesztenyi C. K., Rice J. R., Thacher H. C. Jr. and C. Witzgall (1968) *Computer Approximations*. Wiley. New York.

Hausman J.A. and Wise D.A. (1978) A conditional probit model for qualitative choice: discrete decisions recognising interdependence and heterogeneous preferences. *Econometrica*, **46**, 403-426.

Healy M.J.R. (1968) Algorithm AS6: triangular decomposition of a symmetric matrix. *Applied Statistics*, **17**, 195-196.

Hicks J.E. and Ham H. (1997) Disaggregate simplicial decomposition algorithm for large scale network equilibrium modeling. Presented at the *INFORMS 1997 Meeting*, Dallas.

Hill I. D. (1973) Algorithm AS66: The normal integral. *Applied Statistics*, **22**, 424-427.

Hohenbichler M. (1981) Approximate evaluation of the multinormal distribution function. In: *Studies on the Reliability of Redundant Structural Systems*, Technical University of Munich, Heft 58, 55-66.

Hohenbichler M. and Rackwitz R. (1983) First order concepts in system reliability. *Structural Safety*, **1**(3), 177-188.

Horn R.A. and Johnson C.R. (1990) *Matrix Analysis*. Cambridge University Press.

Horowitz J.L. (1991) Reconsidering the multinomial probit model. *Transportation Research B*, **25**(6), 433-438.

Horowitz J.L., Sparmann J.M. and Daganzo C.F. (1982) An investigation of the accuracy of the Clark approximation for the multinomial probit model. *Transportation Science*, **16**(3), 382-401.

Huang H.J. (1995) A combined algorithm for solving and calibrating the stochastic traffic assignment model. *Journal of the Operational Research Society*, **46**, 977-987.

Huang H.J. and Bell M.G.H. (1998) A study on logit assignment which excludes all cyclic flows. *Transportation Research B*, **32**(6), 401-412.

Hughes P.C. (1998) Link-based methods for stochastic user equilibrium traffic assignment. PhD Thesis. Napier University, Edinburgh.

Hughes P.C. and Maher M.J. (1997) SUE and junction modelling - the extension of stochastic user equilibrium assignment to model realistic junction delays. In: *Proceedings of the 25th European Transport Forum*, Brunel University, 65-75.

Hughes P.C. and Maher M.J. (1998) New methods for solving the SUE assignment problem. Presented at the 3rd IMA Conference on Mathematics in Transport Planning and Control, Cardiff.

Hunt D.T. and Kornhauser A.L. (1996) Assigning traffic over essentially-least-cost paths. *Transportation Research Record*, **1556**, 1-7.

Jayakrishnan R., Tsai W.K., Phrasher J.N., Rajadhyasha S. (1994), Faster path-based algorithm for traffic assignment. *Transportation Research Record*, **1443**, 75-82.

Joe H. (1995) Approximations to multivariate rectangle probabilities based on conditional expectations. *Journal of the American Statistical Association*, **431**, 957-964 (with corrections from <http://hajek.stat.ubc.ca/~harry/table1a7.txt>).

Johnson N.L. and Kotz S. (1972) *Distributions in Statistics, Vol. 4: Continuous Multivariate Distributions*. Wiley. New York.

Kamakura W.A. (1989) The estimation of multinomial probit models: a new calibration algorithm. *Transportation Science*, **23**(4), 253-265.

Keane M.P. (1992) A note on the identification in the multinomial probit model. *Journal of Business and Economic Statistics*, **10**(2), 193-200.

Kontolemakis J. (1998) Fitting modal split models using ALOGIT. Unpublished MSc Dissertation, Napier University, Edinburgh.

Koppelman F.S. and Sethi V. (2000) Closed form logit models. In: *Handbook of Transport Modeling* (eds: D.A. Hensher and K.J. Button), Pergamon Press, Oxford, 209-227.

Koppelman F.S. and Wen C.-H. (2000) The paired combinatorial logit model: properties, estimation and application. *Transportation Research B*, **34**(2), 75-89.

Kotz S. and Johnson N.L. (1982) *Encyclopaedia of Statistical Sciences*. Wiley. New York.

Kupiszewska D. and Van Vliet D. (1998) Incremental traffic assignment: a perturbation approach. Presented at the 3rd *IMA Conference on Mathematics in Transport Planning and Control*, Cardiff.

Kupiszewska D. and Van Vliet D. (1999) 101 uses for path-based assignment. In: *Proceedings of the European Transport Conference 1999*, Cambridge.

Lam S.H. (1991) Multinomial probit model estimation: computational procedures and applications. PhD dissertation. Civil Engineering Department, The University of Texas at Austin.

Langdon M.G. (1981) A theoretical study of the problem of multiple choice in transport models. *Assessment Division Working Paper (81)5*. TRRL, Crowthorne.

Langdon M.G. (1984a) Improved algorithms for estimating choice probabilities in the multinomial probit model. *Transportation Science*, **18**(3), 267-299.

Langdon M.G. (1984b) Methods of determining choice probability in utility maximising multiple alternative models. *Transportation Research B*, **18**(3), 209-234.

Larsson T. and Patriksson M. (1992) Simplicial Decomposition with disaggregate representation for the traffic assignment problem. *Transportation Science*, **26**(1), 4-17.

LeBlanc L.J. (1975) An algorithm for the discrete network design problem. *Transportation Science*, **9**, 183-199.

Lerman S.R. and Manski C.F. (1981) On the use of simulated frequencies to approximate choice probabilities. In: *Structural Analysis of Discrete Data with Econometric Applications* (eds. C.F. Manski and D. McFadden), MIT Press, Cambridge MA.

Leurent F.M. (1994) Cost versus time equilibrium over a network. *Transportation Research Record*, **1443**, 84-91.

Leurent F.M. (1995) The practice of a dual criteria assignment model with a continuously distributed value-of-time. In: *Proceedings of the 23rd European Transport Forum, Seminar E*, PTRC, London, 117-128.

Leurent F.M. (1996a) Path-storing equilibration algorithms for several traffic assignment models. In: *Applications of Advanced Technologies in Transportation Engineering* (eds: Y.J. Stephanedes and F. Filippi), American Society of Civil Engineers, 633-638.

Leurent F.M. (1996b) The theory and practice of a dual criteria assignment model with a continuously distributed value-of-time. In: *Proceedings of 13th International Symposium on Transportation and Traffic Theory* (ed. J.B. Lesort), Elsevier Press, New York, 455-477.

Leurent F.M. (1997a) Contribution to logit assignment model. *Transportation Research Record*, **1493**, 207-212.

Leurent F.M. (1997b) Curbing the computational difficulty of the logit equilibrium assignment model. *Transportation Research B*, **31**(4), 315-326.

Liu Y.-H and Mahamassani H.S. (2000) Global maximum likelihood estimation procedure for multinomial probit (MNP) model parameters. *Transportation Research B*, **34**(5), 419-449.

Maher M.J. (1992) SAM - A stochastic assignment model. In: *Mathematics in Transport Planning and Control* (ed. J.D. Griffiths), Oxford University Press, 121-132.

Maher M.J. (1998) Algorithms for logit-based stochastic user equilibrium assignment. *Transportation Research B*, **32**(8), 539-550.

Maher M.J. and Hughes P.C. (1995) The attractions of SUE - a practicable approach to stochastic user equilibrium assignment. In: *Proceedings of the 23rd European Transport Forum, Seminar E*, 103-115.

Maher M.J. and Hughes P.C. (1996a) Estimation of the potential benefit from an ATT system using a multiple user class stochastic user equilibrium assignment model. In: *Applications of Advanced Technologies in Transportation Engineering* (eds: Y.J Stephanedes. and F. Filippi), American Society of Civil Engineers, 700-704.

Maher M.J. and Hughes P.C. (1996b) Manifestations of SUE - a comparison and evaluation of different methods of stochastic user equilibrium assignment. In: *Proceedings 24th European Transport Forum*.

Maher M.J. and Hughes P.C. (1997a) A probit-based stochastic user equilibrium assignment model. *Transportation Research B*, **31**(4), 341-355.

Maher M.J. and Hughes P.C. (1997b) An algorithm for SUEED - Stochastic User Equilibrium with Elastic Demand. In: *Proceedings of the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems*, Chania, Greece.

Maher M.J. and Hughes P.C. (1998a) Recent developments in stochastic assignment modelling. *Traffic Engineering and Control*, **39**(3), 174-179.

Maher M.J. and Hughes P.C. (1998b) New algorithms for the solution of the stochastic user equilibrium problem with elastic demand. In: *Proceedings of the 8th World Conference on Transport Research*.

Maher M.J. and Hughes P.C. (1998c) *New Methods for Solving the SUE Assignment Problem*. Unpublished Internal Report, Department of Civil and Transportation Engineering, Napier University, Edinburgh.

Maher M.J. and Zhang X. (2000) Formulation and algorithms for the problem of stochastic user equilibrium assignment with elastic demand. In: *Proceedings of the 8th Meeting of the Euro Working Group on Transportation*, Rome, Italy.

Maher M.J., Hughes P.C. and Kim K.-S. (1999) New algorithms for the solution of the stochastic user equilibrium assignment problem with elastic demand. In: *Proceedings of the 14th International Symposium on Transportation and Traffic Theory* (ed. A. Ceder).

Maher M.J., Rosa A. and Kontolemakis J. (1999) Bias in discrete choice models estimates. In: *Proceedings of the European Transport Conference 1999*, Cambridge.

McFadden D. (1973) Conditional logit analysis of quantitative choice behavior. In: *Frontier of Econometrics* (ed. P. Zarembka). Academic Press, New York.

McFadden D. (1975) *On Independence, Structure, and Simultaneity in Transportation Demand Analysis*. Working Paper No. 7511, Urban Travel Demand Forecasting Project, Institute of Transportation and Traffic Engineering, University of California, Berkeley.

McFadden D. (1978) Modelling the choice of residential location. In: *Spatial Interaction Theory and Planning Models* (eds.: A. Karlqvist, L. Lundqvist, F. Snickars, and J. Weibull), North Holland, Amsterdam. 75-96.

McFadden D. (1989) A method of simulated moments for estimation of discrete choice response models with numerical integration. *Econometrica*, **57**(5), 995-1026.

McFadden D. and Train K. (1996) *Mixed MNL Models for Discrete Response*. Working Paper. Department of Economics. University of California. Berkeley.

Melchers R.E. (1999) *Structural Reliability Analysis and Prediction (Second Edition)*. Wiley. Chichester. England.

Mendell N.R. and Elston R.C. (1974) Multifactorial qualitative traits: genetic analysis and prediction of recurrence risks. *Biometrics*, **30**, 41-57.

Miller H.J. (1997) *Towards Consistent Travel Demand Estimation in Transportation Planning: A Guide to the Theory and Practice of Equilibrium Travel Demand Modeling*. Research Report, Bureau of Transportation Statistics, U. S. Department of Transportation. Available at <http://www.bts.gov/tmip/papers/feedback/miller/toc.htm>

Milton R.C. (1972) Computer evaluation of the multivariate Normal integral. *Technometrics*, **14**(4), 881-889

Munizaga M.A. and Ortúzar J. de D. (1997) On the applicability of the multinomial probit model. In: *Proceedings 25th European Transport Forum, Seminar F*, Brunel University, UK.

Munizaga M.A., Heydecker B.J. and Ortúzar J. de D. (2000) Representation of heteroskedasticity in discrete choice models. *Transportation Research B*, **34**, 219-240.

Munizaga M.A., Heydecker B.G. and Ortúzar J. de D. (1997) On the error structure of discrete choice models. *Traffic Engineering and Control*, **38**(11), 593-597.

MVA (1994) *TRIPS version 7.0. User Manual*. MVA.

MVA (1997) *Impact of Highway Capacity Reductions. Report on Modelling*. MVA.

Nielsen O.A. (1996) Do stochastic traffic assignment models consider differences in road users utility functions? In: *Proceedings of the 24th European Transport Forum*, PTRC.

Nielsen O.A. (1997) On the distribution of the stochastic components in SUE traffic assignment models. In: *Proceedings of the 25th European Transport Forum*, PTRC, 77-93.

Nielsen O.A., Frederiksen R.D. and Simonsen R. (1998) Stochastic user equilibrium traffic assignment with turn delays in intersections. *International Transactions in Operational Research*, 5(6), 555-568.

Nuzzolo A., Russo F., Crisalli U. (1997) A pseudo dynamic assignment to extraurban road networks using a C-logit route choice model. In: *Proceedings of the 25th European Transport Forum*, 95-105.

Olson J.M. and Weissfeld L.A. (1991) Approximation of certain multivariate Normal integrals. *Statistics and Probability Letters*, 11, 309-318.

Ortúzar J. de D. (2001) On the development of the nested logit model. *Transportation Research B*, 35(2), 213-216.

Ortúzar J. de D. and Willumsen L.G. (1994) *Modelling Transport, Second Edition*, Wiley, Chichester, England.

Pandey M.D. (1998a) An effective approximation to evaluate multinormal integrals. *Structural Safety*, 20, 51-67.

Pandey M.D. (1998b) A simple approach to multinormal integration with applications to the system reliability computation. *Civil Engineering and Environmental Systems*, **15**, 89-105.

Park D. and Rilett L.R. (1997) Identifying multiple and reasonable paths in transportation networks A heuristic approach. *Transportation Research Record*, **1607**, 31-37.

Pattison T. and Gossink D. (1999) Misclassification probability bounds for multivariate gaussian classes. *Digital Signal Processing*, **9**, 280-296.

Polak E. and Ribière G. (1969) Note sur la convergence des methodes de directions conjuguées. *Revue Française d'Informatique et de Recherche Operationelle, Serie Rouge*, **16**, 35-43.

Prashker J.N. and Bekhor S. (1998) Investigation of stochastic network loading procedures. *Transportation Research Record*, **1645**, 94-102.

Press W.H., Teukolsky S.A., Vetterling, W.T. and Flannery B.P. (1992) *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Second edition. Cambridge University Press.

Rice J., Reich T. and Cloninger R.C. (1979) An approximation to the multivariate Normal integral: its application to multifactorial qualitative traits. *Biometrics*, **35**, 451-459.

Rosa A. (2001) *Path-based traffic assignment with probit analytical methods*. Unpublished paper for the 33rd UTSG Conference. St. Anne College, Oxford.

Rosa A. (2002) *Stochastic user equilibrium traffic assignment with multiple user classes and elastic demand*. Unpublished paper for the 34th UTSG Conference. Edinburgh City Chambers, Edinburgh.

Rosa A. and Maher M.J. (2002a) Stochastic user equilibrium traffic assignment with multiple user classes and elastic demand. In: *Proceedings of the Euro Working Group on Transportation 2002 Conference*. Bari, Italy, 392-397.

Rosa A. and Maher M.J. (2002b) Algorithms for solving the probit path-based stochastic user equilibrium traffic assignment problem with one or more user classes. In: *Transportation and Traffic Theory in the 21st Century. Proceedings of the 15th International Symposium on Transportation and Traffic Theory* (ed. M.A.P. Taylor), Pergamon Press. 371-392.

Saleh W.S., Berghout E.A., Cassir C., Bell M.G.H. (1998) Modelling the effect of TDM packages in urban areas using the Path Flow Estimator. In: *Proceedings of the 8th World Conference on Transport Research*, Antwerp, Belgium.

Scales L.E. (1985) *Introduction to Non-Linear Optimisation*. Macmillan. London.

Schervish M.J. (1984) Multivariate Normal probabilities with error bound. *Applied Statistics*, **33**, 81-94.

Schittenhelm H. (1990) On the integration of an effective assignment algorithm with path and path-flow management in a combined trip distribution and traffic assignment algorithm. In: *Proceedings of the 18th PTRC Summer Annual Meeting*, 203-214.

Scott K. and Bernstein D. (1997) Solving a best path problem when the value of time function is nonlinear. Submitted to the Transportation Research Board.

Sheffi Y. (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, New Jersey.

Sheffi Y. and Daganzo C.F. (1977) Hypernetworks and supply-demand equilibrium obtained with disaggregate demand models. *Transportation Research Records*, **673**, 113-121.

Sheffi Y. and Daganzo C.F. (1980) Computation of equilibrium over transportation networks: the case of disaggregate demand models. *Transportation Science*, **14**(2), 155-173.

Sheffi Y. and Powell W. (1981) A comparison of stochastic and deterministic traffic assignment over congested networks. *Transportation Research B*, **15**(1), 53-64.

Sheffi Y. and Powell W. (1982) An algorithm for the equilibrium assignment problem with random link times. *Networks*, **12**, 191-207.

Sheffi Y., Hall R. and Daganzo C. (1982) On the estimation of the multinomial probit model. *Transportation Research A*, **16**(5-6), 447-456.

Shewchuk J.R. (1994) *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. School of Computer Science. Carnegie Mellon University.

Smith M.J. (1979) The existence, uniqueness and stability of traffic equilibria. *Transportation Research B*, **13**, 293-304.

Solow A.R. (1990) A method for approximating multivariate orthant normal probabilities. *Journal of Statistical Computation and Simulation*, **37**, 225-229.

Somerville P.N. (1998) Numerical computation of multivariate Normal and multivariate-t probabilities over convex regions. *Journal of Computational and Graphical Statistics*, **7**(4), 529-544.

Somerville P.N. and Wang M.C. (1994) Computation of multivariate Normal probabilities over convex regions. In: *Proceedings of the 26th Symposium on the Interface, Computing Science and Statistics*, Fairfax, Virginia. Interface Foundation. 229-231.

Sparmann J. and Daganzo C.F. (1982) *TROMP: User's Manual*. ITS Research Report. Institute of Transportation Studies, University of California, Berkeley.

- Sparmann J.M., Daganzo C.F. and Soheily M. (1983) Linear probit models: statistical properties and improved estimation methods. *Transportation Research B*, **17**(1), 67-86.
- Sun C., Jayakrishnan R. and Tsai W.K. (1996) Computational study of a path-based algorithm and its variants for static traffic assignment. *Transportation Research Record*, **1537**, 106-115.
- Tang L.K. and Melchers R.E. (1987) Improved approximation for multinormal integral. *Structural Safety*, **4**, 81-93.
- Terada S. and Takahashi T. (1988) Failure conditioned reliability index. *ASCE Journal of Structural Engineering*, **114**(4), 943-952.
- Thomas R. (1991) *Traffic Assignment Techniques*, Avebury Technical, Aldershot.
- Thompson A.S. (1974) The multivariate Normal distribution as applied to EMC problems. *IEEE Transactions on Electromagnetic Compatibility*, **16** (2), 115-125.
- Thurstone L. (1927) A law of comparative judgement. *Psychological Review*, **34**, 273-286.
- Tong Y.L. (1990) *The Multivariate Normal Distribution*. Springer-Verlag. New York.
- Train K. (1998) Recreation demand models with taste differences over people. *Land Economics*, **74** (2), 230-239.
- Train K. (2002) *Discrete Choice Methods with Simulation*. Pre-print version available on: <http://elsa.berkeley.edu/~train>. Also published in 2003 by Cambridge University Press.
- Tronrud D.E. (1992) Conjugate-direction minimization - an improved method for the refinement of macromolecules. *Acta Crystallographica A*, **48**, 912-916.

US Bureau of Public Roads (1964) *Traffic Assignment Manual*. US Department of Commerce, Washington DC.

Van Vliet D. (1981) Selected node-pair analysis in Dial's assignment algorithm. *Transportation Research B*, **15**, 65-68.

Van Vliet D. and Hall M. (1993) *SATURN Version 8.4 User Manual*. ITS, University of Leeds.

Van Vuren T. (1994) The trouble with SUE. Stochastic assignment options in practice. In: *Proceedings of the PTRC Summer Annual Meeting 1994, Seminar H*, 41-52.

Van Vuren T. and Watling D. (1991) Multiple user class assignment model for route guidance. *Transportation Research Record*, **1306**, 22-32.

Vovsha P. (1997) Application of cross-nested logit model to mode choice in Tel Aviv, Israel, Metropolitan area. *Transportation Research Record*, **1607**, 6-15.

Vovsha P. and Bekhor S. (1998) Link nested logit model of route choice: overcoming route overlapping problem. *Transportation Research Record*, **1645**, 133-142.

Vythoulkas, P.C. (1990) A dynamic stochastic assignment model for the analysis of general networks. *Transportation Research B*, **24**(6), 453-469.

Wardrop J.G. (1952) Some theoretical aspects of road traffic research. *Proceedings, Institution of Civil Engineers II*, **1**, 325-378.

Weeks M. (1997) The multinomial probit model revisited: a discussion of parameter estimability, identification and specification testing. *Journal of Economic Surveys*, **11**(3), 297-320.

Wen C.-H. and Koppelman F.S. (2001) The generalised nested logit model. *Transportation Research B*, **35**(7), 627-641.

Williams H.W.C.L. (1977) On the formation of travel demand models and economic evaluation measures of users benefits. *Environment and Planning A*, **9**, 285-344.

Yai T. and Shimizu T. (1998) Multinomial probit with structured covariance for choice situations with similar alternatives. *Transportation Research Record*, **1645**, 69-75.

Yai T., Iwakura S. and Morichi S. (1997) Multinomial probit with structured covariance for route choice behaviour. *Transportation Research B*, **31**(3), 195-207.

APPENDIX A

A1 Gradient and Hessian of the MUC SUE Objective Function in Terms of Standardised Link Flows

The MUC SUE objective function in terms of standardised link flows (obtained from the formulation of Daganzo (1982)) is:

$$z_{MUCSUE}(\mathbf{v}) = -\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} S_{rs}^{(k)}(\mathbf{v}) + \sum_I v_i b_i(v_i) - \sum_I \int_0^{v_i} b_i(\omega) d\omega \quad (\text{A1.1})$$

A typical term of its gradient can be derived by using the chain rule starting from the gradient w.r.t. the common component of the link costs, reported in 5.2.3.1. Alternatively (A1.1) can be differentiated directly to write:

$$\frac{\partial z_{MUCSUE}(\mathbf{v})}{\partial v_i} = -\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} \frac{\partial S_{rs}^{(k)}}{\partial v_i} + b_i(v_i) + v_i \frac{db_i}{dv_i} - b_i(v_i) \quad (\text{A1.2})$$

Recalling that, from the properties of the satisfaction (see, Sheffi, 1985), it is:

$$\frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} = P_{rsp}^{(k)} \quad (\text{A1.3})$$

that the path costs by user class are additive:

$$e_p^{(k)} = \sum_I c_i^{(k)} \delta_{ip}^{(k)} \quad (\text{A1.4})$$

and that link costs by user class are separable and defined as:

$$c_i^{(k)} = c_{0i}^{(k)} + \beta^{(k)} b_i(v_i) \quad (\text{A1.5})$$

it is:

$$\frac{\partial S_{rs}^{(k)}}{\partial v_i} = \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial v_i} = \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial c_i} \frac{dc_i}{dv_i} = \sum_P P_{rsp}^{(k)} \beta^{(k)} \delta_{ip} \frac{db_i}{dv_i} \quad (\text{A1.6})$$

Substituting (A1.6) in (A1.2), it results in:

$$\frac{\partial z_{MUCSUE}(\mathbf{v})}{\partial v_i} = \left(- \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + v_i \right) \frac{db_i(v_i)}{dv_i} \quad (\text{A1.7})$$

The general term of the Hessian is obtained by differentiating (A1.7) once more w.r.t. a generic standardised link flow:

$$\frac{\partial z_{MUCSUE}(\mathbf{v})}{\partial v_i \partial v_j} = \frac{\partial}{\partial v_j} \left(\left(- \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + v_i \right) \frac{db_i(v_i)}{dv_i} \right) \quad (\text{A1.8})$$

This results in:

$$\begin{aligned} \frac{\partial z_{MUCSUE}(\mathbf{v})}{\partial v_i \partial v_j} &= \frac{db_i(v_i)}{dv_i} \delta_{ij} + \left(- \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \frac{d^2 b_i(v_i)}{dv_i^2} \delta_{ij} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \frac{\partial P_{rsp}^{(k)}}{\partial v_j} \delta_{pi} \frac{db_i(v_i)}{dv_i} \end{aligned} \quad (\text{A1.9})$$

where δ_{ij} is 1 when $j=i$ and is zero otherwise, since the link costs are separable. The second term can be rewritten by considering that the part in brackets is the difference between the auxiliary and current solution, respectively w_i and v_i , and the third term can be developed by considering that the choice probability of a path is a function of the path costs $e_p^{(k)}$. Thus, considering these relationships and tidying up, it gives:

$$\frac{\partial z_{MUCSUE}(\mathbf{v})}{\partial v_i \partial v_j} = \frac{db_i(v_i)}{dv_i} \delta_{ij} + (-w_i + v_i) \frac{d^2 b_i(v_i)}{dv_i^2} \delta_{ij} + \quad (\text{A1.10})$$

$$- \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \delta_{pi} \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \frac{\partial e_m^{(k)}}{\partial v_j} \frac{db_i(v_i)}{dv_i}$$

and recalling (A1.4) and (A1.5) it is possible to write:

$$\frac{\partial e_m^{(k)}}{\partial v_j} = \frac{\partial e_m^{(k)}}{\partial c_j^{(k)}} \frac{\partial c_j^{(k)}}{\partial v_j} = \delta_{mj} \beta^{(k)} \frac{db_j}{dv_j} \quad (\text{A1.11})$$

Thus, the general term of the Hessian of (A1.1) can finally be expressed as:

$$\frac{\partial z_{MUCSUE}(\mathbf{v})}{\partial v_i \partial v_j} = \frac{db_i}{dv_i} \delta_{ij} + (-w_i + v_i) \frac{d^2 b_i}{dv_i^2} \delta_{ij} + \quad (A1.12)$$

$$- \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \delta_{pi} \frac{db_i}{dv_i} \delta_{mj} \frac{db_j}{dv_j}$$

In matrix form $\nabla^2 z_{MUCSUE}(\mathbf{v})$ can be written as:

$$\nabla^2 z_{MUCSUE}(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbf{b} + \mathbf{R} \nabla_{\mathbf{v}}^2 \mathbf{b} + \quad (A1.13)$$

$$+ \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs} \left(\nabla_{\mathbf{v}} b \Delta_{rs}^{(k)} \right) \left(- \nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} b \Delta_{rs}^{(k)} \right)$$

where \mathbf{R} is the vector of differences between the current and the auxiliary link flows and $\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)}$ is the Jacobian of the path choice probabilities for class k and for the paths between r and s .

A2 Gradient and Hessian of the MUC SUE Objective Function in Terms of Common Link Costs

The MUC SUE objective function expressed in terms of common link costs (obtained from the formulation of Daganzo (1982)) is:

$$z_{MUCSUE}(\mathbf{b}) = -\sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} S_{rs}^{(k)} + \sum_I \int_0^{\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}}} b_i^{-1}(\omega) d\omega \quad (\text{A2.1})$$

The gradient of (A2.1) is obtained in detail in 5.2.3.1 and its i th component is:

$$\frac{\partial z_{MUCSUE}(\mathbf{b})}{\partial b_i} = -\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \quad (\text{A2.2})$$

A general term of the Hessian of the MUC SUE objective function in terms of common link costs can be written by differentiating (A2.2) w.r.t. a component of the vector of common parts of the link costs, thus writing:

$$\frac{\partial^2 z_{MUCSUE}(\mathbf{b})}{\partial b_i \partial b_j} = \frac{\partial}{\partial b_j} \left(-\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} + b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \quad (\text{A2.3})$$

Recalling that the path costs by user class are additive:

$$e_p^{(k)} = \sum_I c_i^{(k)} \delta_{ip} \quad (\text{A2.4})$$

and that the link costs are defined by:

$$c_i^{(k)} = c_{0i}^{(k)} + \beta^{(k)} b_i(v_i) \quad (\text{A2.5})$$

the derivative of the first term is:

$$\frac{\partial}{\partial b_j} \left(-\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{pi} \right) = -\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \frac{\partial P_{rsp}^{(k)}}{\partial b_j} \delta_{pi} \quad (\text{A2.6})$$

$$-\sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \frac{\partial e_m^k}{\partial b_j} \delta_{pi} = -\sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \delta_{mi} \delta_{pi}$$

The second term is simply:

$$\frac{\partial}{\partial b_j} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) = \frac{\partial}{\partial b_i} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \delta_{ij} \quad (\text{A2.7})$$

where δ_{ij} is 1 when $j=i$ and is zero otherwise, since the costs are separable.

The general term of the Hessian is thus:

$$\begin{aligned} \frac{\partial^2 z_{MUCSUE}(\mathbf{b})}{\partial b_i \partial b_j} = & -\sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \delta_{mi} \delta_{pi} + \\ & + \frac{\partial}{\partial b_i} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \delta_{ij} \end{aligned} \quad (\text{A2.8})$$

Equivalently $\nabla^2 z_{MUCSUE}(\mathbf{b})$, the Hessian of (A2.1), can be written in matrix form as:

$$\nabla^2 z_{MUCSUE}(\mathbf{b}) = \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \Delta_{rs}^{(k)} \left(-\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \mathbf{\Delta}_{rs}^{(k)T} + \nabla_{\mathbf{b}} \mathbf{b}^{-1} \quad (\text{A2.9})$$

where $\Delta_{rs}^{(k)}$ is the link-path incidence matrix for class k and $\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)}$ is the Jacobian of the path choice probabilities for class k and for the paths between r and s .

A3 Gradient and Hessian of the SUE ED Objective Function in Terms of Link Flows

A typical term of the gradient of the SUE ED function in terms of link flows:

$$z_{SUEED}(\mathbf{x}) = -\sum_I \int_0^{x_i} c_i(\omega) d\omega + \sum_I x_i c_i(x_i) - \sum_{RS} \int_{S_{0rs}}^{S_{rs}(x)} D_{rs}(S_{rs}) dS_{rs} \quad (\text{A3.1})$$

can be obtained by writing:

$$\frac{\partial z_{SUEED}(\mathbf{x})}{\partial x_i} = -c_i(x_i) + c_i(x_i) + x_i \frac{dc_i}{dx_i} - \sum_{RS} \frac{d}{dS_{rs}} \left(\int_{S_{0rs}}^{S_{rs}(x)} D_{rs}(S_{rs}) dS_{rs} \right) \frac{\partial S_{rs}(x)}{\partial x_i} \quad (\text{A3.2})$$

The $\frac{\partial S_{rs}(x)}{\partial x_i}$ can be written as:

$$\frac{\partial S_{rs}}{\partial x_i} = \sum_P \frac{\partial S_{rs}}{\partial e_p} \frac{\partial e_p}{\partial c_i} \frac{dc_i}{dx_i} \quad (\text{A3.3})$$

Thus substituting (A3.3) in (A3.2) it results in:

$$\frac{\partial z_{SUEED}(\mathbf{x})}{\partial x_i} = x_i \frac{dc_i}{dx_i} - \sum_{RS} D_{rs}(S_{rs}) \sum_P \frac{\partial S_{rs}}{\partial e_p} \frac{\partial e_p}{\partial c_i} \frac{dc_i}{dx_i} \quad (\text{A3.4})$$

Recalling that, by the properties of the satisfaction (see Sheffi, 1985), it is:

$$\frac{\partial S_{rs}}{\partial e_p} = P_{rsp} \quad (\text{A3.5})$$

and that path costs are additive:

$$e_p = \sum_I c_i \delta_{ip} \quad (\text{A3.6})$$

and link costs are separable, (A3.4) can be elaborated as:

$$\frac{\partial Z_{SUEED}(\mathbf{x})}{\partial x_i} = x_i \frac{dc_i}{dx_i} - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{ip} \frac{dc_i}{dx_i} \quad (\text{A3.7})$$

which can be tidied up to give:

$$\frac{\partial Z_{SUEED}(\mathbf{x})}{\partial x_i} = \left(x_i - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{ip} \right) \frac{dc_i}{dx_i} \quad (\text{A3.8})$$

The general term of the Hessian of the new SUE ED objective function in terms of flows is obtained by differentiating the general term of the gradient (A3.8) once more w.r.t. a link flow:

$$\begin{aligned} \frac{\partial^2 Z_{SUEED}(\mathbf{x})}{\partial x_i \partial x_j} &= \frac{\partial}{\partial x_j} \left(x_i \frac{dc_i}{dx_i} - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{ip} \frac{dc_i}{dx_i} \right) \\ &= \left(\frac{dc_i}{dx_i} + x_i \frac{d^2 c_i}{dx_i^2} \right) \delta_{ij} - \left(\sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{ip} \frac{d^2 c_i}{dx_i^2} \right) \delta_{ij} + \\ &\quad - \sum_{RS} \frac{\partial D_{rs}(S_{rs})}{\partial x_j} \sum_P P_{rsp} \delta_{ip} \frac{dc_i}{dx_i} - \sum_{RS} D_{rs}(S_{rs}) \sum_P \frac{\partial P_{rsp}}{\partial x_j} \delta_{ip} \frac{dc_i}{dx_i} \end{aligned} \quad (\text{A3.9})$$

where δ_{ij} is 1 if $i=j$ and 0 otherwise. And, finally,

$$\begin{aligned} \frac{\partial^2 Z_{SUEED}(\mathbf{x})}{\partial x_i \partial x_j} &= \left(\frac{dc_i}{dx_i} + (x_i - y_i) \frac{d^2 c_i}{dc_i^2} \right) \delta_{ij} - \sum_{RS} \frac{dD_{rs}(S_{rs})}{dS_{rs}} \sum_P P_{rsp} \delta_{ip} \frac{dc_i}{\partial x_i} \sum_M P_{rsm} \delta_{jm} \frac{dc_j}{\partial x_j} + \\ &\quad - \sum_{RS} D_{rs}(S_{rs}) \sum_P \sum_M \frac{\partial P_{rsp}}{\partial c_m} \delta_{jm} \frac{dc_j}{dx_j} \delta_{ip} \frac{dc_i}{dx_i} \end{aligned} \quad (\text{A3.10})$$

This can be written in matrix form as:

$$\begin{aligned}
\nabla^2 z(\mathbf{x})_{SUEED} &= \nabla_x \mathbf{c} + \nabla_x^2 \mathbf{c} \mathbf{R} + \\
&+ \sum_{RS} \left(-\frac{dD_{rs}(S_{rs})}{dS_{rs}} \right) (\nabla_x \mathbf{c} \Delta_{rs} \mathbf{P}_{rs}) (\nabla_x \mathbf{c} \Delta_{rs} \mathbf{P}_{rs})^T + \\
&+ \sum_{RS} D_{rs}(S_{rs}) (\nabla_x \mathbf{c} \Delta_{rs}) (-\nabla_e \mathbf{P}_{rs}) (\nabla_x \mathbf{c} \Delta_{rs})^T
\end{aligned} \tag{A3.11}$$

Where \mathbf{R} is the vector of differences between the current and the auxiliary link flows and $\nabla_e \mathbf{P}_{rs}$ is the Jacobian of the path choice probabilities for the paths between r and s .

A4 Gradient and Hessian of the SUE ED Objective Function in Terms of Link Costs

The gradient of the SUEED function in terms of link costs:

$$z_{SUEED}(\mathbf{c}) = \sum_I \int_{c_{0i}}^{c_i} c_a^{-1}(\omega) d\omega - \sum_{RS} \int_{S_{0rs}}^{S_{rs}(\mathbf{c})} D_{rs}(S_{rs}) dS_{rs} \quad (\text{A4.1})$$

has a general term that can be derived as:

$$\frac{\partial z_{SUEED}(\mathbf{c})}{\partial c_i} = c_i^{-1}(c_i) - \sum_{RS} \frac{d}{dS_{rs}} \left(\int_{S_{0rs}}^{S_{rs}(\mathbf{c})} D_{rs}(S_{rs}) dS_{rs} \right) \frac{\partial S_{rs}(\mathbf{c})}{\partial c_i} \quad (\text{A4.2})$$

The $\frac{\partial S_{rs}(\mathbf{c})}{\partial c_i}$ can be written as:

$$\frac{\partial S_{rs}}{\partial c_i} = \sum_P \frac{\partial S_{rs}}{\partial e_p} \frac{\partial e_p}{\partial c_i} \quad (\text{A4.3})$$

which, substituted in (A4.2), gives:

$$\frac{\partial z_{SUEED}(\mathbf{c})}{\partial c_i} = c_i^{-1}(c_i) - \sum_{RS} D_{rs}(S_{rs}) \sum_P \frac{\partial S_{rs}}{\partial e_p} \frac{\partial e_p}{\partial c_i} \quad (\text{A4.4})$$

Recalling that, from the properties of the satisfaction (see Sheffi, 1985), it is:

$$\frac{\partial S_{rs}}{\partial e_p} = P_{rsp} \quad (\text{A4.5})$$

and that path costs are additive:

$$e_p = \sum_I c_i \delta_{ip} \quad (\text{A4.6})$$

and that link costs are separable (A4.4), can be written as:

$$\frac{\partial z_{SUEED}(\mathbf{c})}{\partial c_i} = c_i^{-1}(c_i) - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{ip} \quad (\text{A4.7})$$

Finally, substituting the current flow for the inverse of the link cost function, (A4.7) can also be written as:

$$\frac{\partial z_{SUEED}(\mathbf{c})}{\partial c_i} = x_i - \sum_{RS} D_{rs}(S_{rs}) \sum_P P_{rsp} \delta_{ip} \quad (\text{A4.8})$$

Differentiating (A4.7) once more w.r.t. a link cost yields the general term of the Hessian:

$$\frac{\partial^2 z_{SUEED}(\mathbf{c})}{\partial c_i \partial c_j} = \frac{\partial}{\partial c_j} \left(c_i^{-1}(c_i) - \sum_{RS} D_{rs}(S_{rs}) \sum_K P_{rsk} \delta_{ik} \right) \quad (\text{A4.9})$$

The first term of (A4.9) is simply:

$$\frac{\partial c_i^{-1}(c_i)}{\partial c_j} = \frac{dc_i^{-1}(c_i)}{dc_j} \delta_{ij} \quad (\text{A4.10})$$

where δ_{ij} is 1 if $i=j$ and is 0 otherwise.

Differentiating the second term, gives:

$$- \sum_{RS} \frac{\partial D_{rs}(S_{rs})}{\partial c_j} \sum_P P_{rsp} \delta_{ip} - \sum_{RS} D_{rs}(S_{rs}) \sum_P \frac{\partial P_{rsp}}{\partial c_j} \delta_{ip} \quad (\text{A4.11})$$

Recalling (A4.5), (A4.6) and that the demand between on OD pair is a function of the satisfaction between the same OD pair only, the general term of the Hessian results:

$$\frac{\partial^2 z_{SUEED}(\mathbf{c})}{\partial c_i \partial c_j} = \frac{dc_i^{-1}(c_i)}{dc_i} \delta_{ij} - \sum_{RS} \frac{dD_{rs}(S_{rs})}{dS_{rs}} \sum_M P_{rsm} \delta_{jm} \sum_P P_{rsp} \delta_{ip} + \quad (\text{A4.12})$$

$$-\sum_{RS} D_{rs}(S_{rs}) \sum_P \sum_M \frac{\partial P_{rsp}}{\partial e_m} \delta_{jm} \delta_{ik}$$

Thus, the Hessian of (A4.1) can be written in matrix form as:

$$\begin{aligned} \nabla^2 Z_{SUEED}(\mathbf{c}) = & \nabla_c \mathbf{c}^{-1} + \sum_{RS} \left(-\frac{dD_{rs}(S_{rs})}{dS_{rs}} \right) (\Delta_{rs} \mathbf{P}_{rs}) (\Delta_{rs} \mathbf{P}_{rs})^T + \\ & + \sum_{RS} D_{rs}(S_{rs}) (\Delta_{rs}) (-\nabla_e \mathbf{P}_{rs}) (\Delta_{rs})^T \end{aligned} \quad (\text{A4.13})$$

where \mathbf{P}_{rs} is the vector of the path choice probabilities for the paths between r and s and $\nabla_e \mathbf{P}_{rs}$ is its Jacobian matrix.

A5 Gradient and Hessian of the MUC SUE ED Objective Function Derived from the Formulation of Maher *et al.* (1999)

The MUC SUE ED objective function derived from the SUE ED one of Maher *et al.* (1999) is:

$$\begin{aligned}
 z_{MUCSUEED}(\mathbf{v}, \mathbf{q}) = & -\sum_I \int_0^{v_i} b_i(\omega) d\omega + \sum_I v_i b_i(v_i) - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} S_{rs}^{(k)}(\mathbf{v}) D_{rs}^{(k)}(S_{rs}^{(k)}(\mathbf{v})) + \\
 & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)-1}(q_{rs}^{(k)}) D_{rs}^{(k)}(S_{rs}^{(k)}(\mathbf{v})) + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \int_{q_{rs0}^{(k)}}^{q_{rs}^{(k)}} D_{rs}^{(k)-1}(\sigma) d\sigma + \\
 & - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} D_{rs}^{(k)-1}(q_{rs}^{(k)})
 \end{aligned} \tag{A5.1}$$

The gradient of (A5.1) has two typical terms that can be obtained, respectively, differentiating the objective function w.r.t. the standardised link flows and w.r.t. the OD flows by user class.

Differentiating (A5.1) w.r.t. a standardised link flow it results:

$$\begin{aligned}
 \frac{\partial z_{MUCSUEED}}{\partial v_i} = & -b_i(v_i) + b_i(v_i) + v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_i} D_{rs}^{(k)}(S_{rs}^{(k)}) + \\
 & - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} S_{rs}^{(k)} \frac{\partial D_{rs}^{(k)}}{\partial v_i} + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)-1}(q_{rs}^{(k)}) \frac{\partial D_{rs}^{(k)}}{\partial v_i}
 \end{aligned} \tag{A5.2}$$

This expression can be elaborated recalling that path costs by user class $e_p^{(k)}$ are additive:

$$e_p^{(k)} = \sum_I c_i^{(k)} \delta_{ip}^{(k)} \tag{A5.3}$$

and link costs by user class are separable and given by:

$$c_i^{(k)} = c_{0i}^{(k)} + \beta^{(k)} b_i(v_i) \quad (\text{A5.4})$$

Moreover, from the properties of the satisfaction (see Sheffi, 1985), it is:

$$\frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} = P_p^{(k)} \quad (\text{A5.5})$$

Thus, it is possible to write (A5.2) as:

$$\begin{aligned} \frac{\partial Z_{MUCSUEED}}{\partial v_i} = & v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial c_p^{(k)}} \frac{dc_p^{(k)}}{dv_i} + \\ & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1}(q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial D_{rs}^{(k)}}{\partial v_i} \end{aligned} \quad (\text{A5.6})$$

and finally:

$$\begin{aligned} \frac{\partial Z_{MUCSUEED}}{\partial v_i} = & v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rs}^{(k)} \delta_{ip} \beta^{(k)} \frac{db_i}{dv_i} + \\ & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1}(q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial D_{rs}^{(k)}}{\partial v_i} \end{aligned} \quad (\text{A5.7})$$

which can be tidied up to give:

$$\begin{aligned} \frac{\partial Z_{MUCSUEED}}{\partial v_i} = & \left(v_i - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rs}^{(k)} \delta_{ip} \right) \frac{db_i}{dv_i} + \\ & + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1}(q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial D_{rs}^{(k)}}{\partial v_i} \end{aligned} \quad (\text{A5.8})$$

Differentiating (A5.1) w.r.t. the OD flows by user class it gives:

$$\frac{\partial Z_{MUCSUEED}}{\partial q_{rs}^{(k)}} = \frac{\alpha^{(k)}}{\beta^{(k)}} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} D_{rs}^{(k)}(S_{rs}^{(k)}) + \frac{\alpha^{(k)}}{\beta^{(k)}} D_{rs}^{(k)-1}(q_{rs}^{(k)}) + \quad (\text{A5.9})$$

$$-\frac{\alpha^{(k)}}{\beta^{(k)}} D_{rs}^{(k)-1}(q_{rs}^{(k)}) - \frac{\alpha^{(k)}}{\beta^{(k)}} q_{rs}^{(k)} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}}$$

which, tidying up, results:

$$\frac{\partial z_{MUCSUEED}}{\partial q_{rs}^{(k)}} = \frac{\alpha^{(k)}}{\beta^{(k)}} (D_{rs}^{(k)}(S_{rs}^{(k)}) - q_{rs}^{(k)}) \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} \quad (\text{A5.10})$$

The derivation of the Hessian of (A5.1) at the equilibrium point, can be organised as in Maher and Zhang (2000) writing the Hessian of the objective function partitioned in four sub-matrices:

$$\begin{aligned} \nabla^2 z_{MUCSUEED} &= \begin{bmatrix} \left[\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial q_{tu}^{(l)}} \right] & \left[\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial q_{rs}^{(k)}} \right] \\ \left[\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial v_i} \right] & \left[\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial v_j} \right] \end{bmatrix} \\ &= \begin{bmatrix} \nabla_{\mathbf{q}}^2 z_{MUCSUEED} & \nabla_{\mathbf{qv}}^2 z_{MUCSUEED} \\ \nabla_{\mathbf{vq}}^2 z_{MUCSUEED} & \nabla_{\mathbf{v}}^2 z_{MUCSUEED} \end{bmatrix} \end{aligned} \quad (\text{A5.11})$$

The North-East submatrix can be written deriving once more (A5.10) w.r.t. an OD flow. Its general entry $\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial q_{tu}^{(l)}}$ is different from 0 only if $rs=tu$ and $l=k$, that is the submatrix is diagonal with non zero entries equal to:

$$\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)2}} = -\frac{\alpha^{(k)}}{\beta^{(k)}} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} + \frac{\alpha^{(k)}}{\beta^{(k)}} (D_{rs}^{(k)}(S_{rs}^{(k)}) - q_{rs}^{(k)}) \frac{d^2 D_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)2}} \quad (\text{A5.12})$$

At the MUC SUE ED point it will be:

$$D_{rs}^{(k)}(S_{rs}^{(k)}) - q_{rs}^{(k)} = 0 \quad \forall r, s, k \quad (\text{A5.13})$$

thus, the diagonal entries of this submatrix will reduce to:

$$\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)2}} = -\frac{\alpha^{(k)}}{\beta^{(k)}} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} \quad (\text{A5.14})$$

The typical entries of the South-East and North-West submatrices of the Hessian (which are symmetric) are obtained by differentiating (A5.10) once more but w.r.t. a standardised link flow:

$$\begin{aligned} \frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial v_i} &= \frac{\alpha^{(k)}}{\beta^{(k)}} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} \frac{\partial D_{rs}^{(k)}(S_{rs}^{(k)})}{\partial v_i} \\ &= \frac{\alpha^{(k)}}{\beta^{(k)}} \frac{dD_{rs}^{(k)-1}(q_{rs}^{(k)})}{dq_{rs}^{(k)}} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \end{aligned} \quad (\text{A5.15})$$

Considering that at MUC SUE ED it is:

$$\frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} = \left(\frac{dD_{rs}^{(k)-1}}{dq_{rs}^{(k)}} \right) \quad (\text{A5.16})$$

the (A5.15) can be rewritten as:

$$\frac{\partial^2 z_{MUCSUEED}}{\partial q_{rs}^{(k)} \partial v_i} = \frac{\alpha^{(k)}}{\beta^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \quad (\text{A5.17})$$

Finally, the South-East submatrix of the Hessian can be obtained by differentiating (A5.8) once more w.r.t. a standardised link flow:

$$\begin{aligned} \frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial v_j} &= \left(\frac{db_i}{dv_i} + v_i \frac{d^2 b_i}{dv_i^2} \right) \delta_{ij} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \frac{\partial^2 S_{rs}^{(k)}}{\partial v_i \partial v_j} \\ &\quad - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial v_j} + \end{aligned} \quad (\text{A5.18})$$

$$\begin{aligned}
& + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1} (q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial^2 D_{rs}^{(k)} (S_{rs}^{(k)})}{\partial v_i \partial v_j} + \\
& - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_j} \frac{\partial D_{rs}^{(k)} (S_{rs}^{(k)})}{\partial v_i}
\end{aligned}$$

Rearranging the third and the last term, that are equivalent, it gives:

$$\begin{aligned}
\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial v_j} & = \left(\frac{db_i}{dv_i} + v_i \frac{d^2 b_i}{dv_i^2} \right) \delta_{ij} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)} (S_{rs}^{(k)}) \frac{\partial^2 S_{rs}^{(k)}}{\partial v_i \partial v_j} \\
& - 2 \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \frac{dD_{rs}^{(k)} (S_{rs}^{(k)})}{dS_{rs}^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial v_j} + \\
& + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1} (q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial^2 D_{rs}^{(k)} (S_{rs}^{(k)})}{\partial v_i \partial v_j}
\end{aligned} \tag{A5.19}$$

The last term vanishes at MUC SUE ED, so it can be disregarded. The $\frac{\partial^2 S_{rs}^{(k)}}{\partial v_i \partial v_j}$ can be elaborated recalling (A5.3), (A5.4) and (A5.5). It results in:

$$\begin{aligned}
\frac{\partial^2 S_{rs}^{(k)}}{\partial v_i \partial v_j} & = \frac{\partial}{\partial v_j} \left(\sum_P P_{rsp}^{(k)} \delta_{ip} \beta^{(k)} \frac{db_i}{dv_i} \right) \\
& = \frac{d^2 b_i}{dv_i^2} \delta_{ij} \sum_P P_{rsp}^{(k)} \delta_{ip} \beta^{(k)} + \sum_P \frac{\partial P_{rsp}^{(k)}}{\partial v_j} \delta_{ip} \beta^{(k)} \frac{db_i}{dv_i} \\
& = \frac{d^2 b_i}{dv_i^2} \delta_{ij} \sum_P P_{rsp}^{(k)} \delta_{ip} \beta^{(k)} + \sum_P \sum_L \frac{\partial P_{rsp}^{(k)}}{\partial e_l^{(k)}} \delta_{jl} \beta^{(k)} \frac{db_j}{dv_j} \delta_{ip} \beta^{(k)} \frac{db_i}{dv_i}
\end{aligned} \tag{A5.20}$$

Considering that at MUC SUE ED $D_{rs}^{(k)} (S_{rs}^{(k)})$ can be replaced by $q_{rs}^{(k)}$ and substituting (A5.20) in (A5.19), it results in:

$$\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial v_j} = \left(\frac{db_i}{dv_i} + v_i \frac{d^2 b_i}{dv_i^2} \right) \delta_{ij} +$$

$$\begin{aligned}
& - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} q_{rs}^{(k)} \left(\frac{d^2 b_i}{dv_i^2} \delta_{ij} \sum_P P_{rsp}^{(k)} \delta_{ip} \beta^{(k)} + \right. \\
& \quad \left. + \sum_P \sum_L \frac{\partial P_{rsp}^{(k)}}{\partial e_l^{(k)}} \delta_{jl} \beta^{(k)} \frac{db_j}{dv_j} \delta_{ip} \beta^{(k)} \frac{db_i}{dv_i} \right) + \\
& - 2 \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial v_j} + \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \left(D_{rs}^{(k)-1}(q_{rs}^{(k)}) - S_{rs}^{(k)} \right) \frac{\partial^2 D_{rs}^{(k)}(S_{rs}^{(k)})}{\partial v_i \partial v_j}
\end{aligned} \tag{A5.21}$$

Tidying up, (A5.21) finally results in:

$$\begin{aligned}
\frac{\partial^2 z_{MUCSUEED}}{\partial v_i \partial v_j} &= \frac{db_i}{dv_i} \delta_{ij} + \left(v_i - \sum_K \alpha^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{ip} \right) \frac{d^2 b_i}{dv_i^2} \delta_{ij} + \\
& - \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \sum_P \delta_{ip} \frac{db_i}{dv_i} \sum_L \frac{\partial P_{rsp}^{(k)}}{\partial e_l^{(k)}} \delta_{jl} \frac{db_j}{dv_j} + \\
& - 2 \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \frac{\partial S_{rs}^{(k)}}{\partial v_i} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial v_j}
\end{aligned} \tag{A5.22}$$

In matrix form (A5.22) can be written as:

$$\begin{aligned}
\nabla_{\mathbf{v}}^2 z_{MUCSUEED} &= \nabla_{\mathbf{v}} \mathbf{b} + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)} \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \right) \left(- \nabla_{e^{(k)}} P_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \right)^T + \\
& + 2 \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \left(\nabla_{\mathbf{v}} S_{rs}^{(k)} \right) \left[\text{diag} \left(- \frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} \right) \right] \left(\nabla_{\mathbf{v}} S_{rs}^{(k)} \right)^T
\end{aligned} \tag{A5.23}$$

A6 Gradient and Hessian of the Short MUC SUE ED Objective Function in Terms of Standardised Flows

The short MUC SUE ED objective function in terms of link flows is:

$$z_{MUCSUEED}(\mathbf{v}) = -\sum_A \int_0^{v_a} b_a(\omega) d\omega + \sum_A v_a b_a(v_a) - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \int_{S_{rs}^{(k)}}^{S_{rs}^{(k)}(\mathbf{v})} D_{rs}^{(k)}(\sigma) d\sigma \quad (\text{A6.1})$$

The gradient of (A6.1) w.r.t. the standard link flows is obtained by recalling that paths costs by user class $e_p^{(k)}$ are additive:

$$e_p^{(k)} = \sum_I c_i^{(k)} \delta_{ip}^{(k)} \quad (\text{A6.2})$$

and link costs are separable and given by:

$$c_i^{(k)} = c_{0i}^{(k)} + \beta^{(k)} b_i(v_i) \quad (\text{A6.3})$$

and that, by the properties of the satisfaction (see Sheffi, 1985), it is:

$$\frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} = P_{rsp}^{(k)} \quad (\text{A6.4})$$

Differentiating (A6.1) once w.r.t. the standardised flows gives:

$$\begin{aligned} \frac{\partial z_{MUCSUEED}(\mathbf{v})}{\partial v_i} &= -b_i(v_i) + b_i(v_i) + v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \frac{\partial S_{rs}^{(k)}}{\partial v_i} \\ &= v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial v_i} \\ &= v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial c_i^{(k)}} \frac{dc_i^{(k)}}{dv_i} \end{aligned} \quad (\text{A6.5})$$

$$= v_i \frac{db_i}{dv_i} - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \beta^{(k)} \frac{db_i}{dv_i}$$

Finally, the general term of the gradient of (A6.1) is:

$$\frac{\partial z_{MUCSUEED}(\mathbf{v})}{\partial v_i} = \left(v_i - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \right) \frac{db_i}{dv_i} \quad (\text{A6.6})$$

A typical term of the Hessian of (A6.1) can be derived by writing:

$$\begin{aligned} \frac{\partial^2 z_{MUCSUEED}(\mathbf{v})}{\partial v_i \partial v_j} &= \left(\frac{db_i}{dv_j} + (v_i - w_i) \frac{d^2 b_i}{dv_i} \right) \delta_{ij} - \sum_K \alpha^{(k)} \sum_{RS} \frac{\partial D(S_{rs}^{(k)})}{\partial v_j} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \frac{db_i}{dv_i} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial P_{rsp}^{(k)}}{\partial v_j} \delta_{ip}^{(k)} \frac{db_i}{dv_i} \end{aligned} \quad (\text{A6.7})$$

where δ_{ij} is 1 if $i=j$ and 0 otherwise. Considering again (A6.2), (A6.3), and (A6.4), (A6.7) can be rewritten as:

$$\begin{aligned} \frac{\partial^2 z_{MUCSUEED}(\mathbf{v})}{\partial v_i \partial v_j} &= \left(\frac{db_i}{dv_i} + (v_i - w_i) \frac{d^2 b_i}{dv_i} \right) \delta_{ij} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} \frac{\partial D_{rs}^{(k)}(S_{rs}^{(k)})}{\partial S_{rs}^{(k)}} \sum_M \frac{\partial S_{rs}^{(k)}}{\partial e_m^{(k)}} \frac{\partial e_m^{(k)}}{\partial v_b} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \frac{db_i}{dv_i} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \delta_{ip}^{(k)} \frac{db_i}{dv_i} \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \frac{\partial e_m^{(k)}}{\partial v_j} \\ &= \left(\frac{db_i}{dv_i} + (v_i - w_i) \frac{d^2 b_i}{dv_i} \right) \delta_{ij} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} \frac{\partial D_{rs}^{(k)}(S_{rs}^{(k)})}{\partial S_{rs}^{(k)}} \sum_M \frac{\partial S_{rs}^{(k)}}{\partial e_m^{(k)}} \delta_{jm}^{(k)} \beta^{(k)} \frac{db_j}{dv_j} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \frac{db_i}{dv_i} + \end{aligned} \quad (\text{A6.8})$$

$$-\sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} D_{rs}^{(k)} (S_{rs}^{(k)}) \sum_P \left(\sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \delta_{mj}^{(k)} \frac{db_j}{dv_j} \right) \delta_{ip}^{(k)} \frac{db_i}{dv_i}$$

More succinctly, the Hessian of (A6.1) can be written in matrix form as:

$$\begin{aligned} \nabla^2 Z_{MUCSUEED}(\mathbf{v}) = & \nabla_{\mathbf{v}} \mathbf{b} + \mathbf{R} \nabla_{\mathbf{v}}^2 \mathbf{b} + \\ & + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} \left(-\frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)} \right)^T + \\ & + \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} D_{rs}^{(k)} (S_{rs}^{(k)}) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \right) \left(-\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)} \right) \left(\nabla_{\mathbf{v}} \mathbf{b} \Delta_{rs}^{(k)} \right)^T \end{aligned} \quad (\text{A6.9})$$

where \mathbf{R} is the diagonal matrix of the differences between the present and the auxiliary standard link flows, $\mathbf{P}_{rs}^{(k)}$ is the vector of the path choice probabilities for the paths between r and s and for class k , and $\nabla_{\mathbf{e}^{(k)}} \mathbf{P}_{rs}^{(k)}$ is its Jacobian matrix.

A7 Gradient and Hessian of the Short MUC SUE ED Objective Function in Terms of Common Link Costs

The short MUC SUE ED objective function in terms of common parts of the link costs is:

$$Z_{MUCSUEED}(\mathbf{b}) = -\sum_I \frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \int_0^{c_i^{(1)} - c_{0i}^{(1)}} b_i^{-1}(\omega) d\omega - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} \int_{S_{0rs}^{(k)}}^{S_{rs}^{(k)}(\mathbf{b})} D_{rs}^{(k)}(\sigma) d\sigma \quad (\text{A7.1})$$

A typical term of its gradient is obtained by recalling that path costs by user class $e_p^{(k)}$ are additive:

$$e_p^{(k)} = \sum_I c_i^{(k)} \delta_{ip}^{(k)} \quad (\text{A7.2})$$

and link costs are separable and given by:

$$c_i^{(k)} = c_{0i}^{(k)} + \beta^{(k)} b_i(v_i) \quad (\text{A7.3})$$

and that, by the properties of the satisfaction (see Sheffi, 1985), it is:

$$\frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} = P_{rsp}^{(k)} \quad (\text{A7.4})$$

Differentiating once w.r.t. the common part of the link costs of a link, (A7.1) results in:

$$\begin{aligned} \frac{\partial Z_{MUCSUEED}(\mathbf{b})}{\partial b_i} &= b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \frac{\partial S_{rs}^{(k)}}{\partial b_i} \\ &= v_i - \sum_K \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial b_i} \\ &= v_i - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial S_{rs}^{(k)}}{\partial e_p^{(k)}} \frac{\partial e_p^{(k)}}{\partial c_i^{(k)}} \frac{\partial c_i^{(k)}}{\partial b_i} \end{aligned} \quad (\text{A7.5})$$

$$= v_i - \sum_K \frac{\alpha^{(k)}}{\beta^{(k)}} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \beta^{(k)}$$

which, tidying up, becomes:

$$\frac{\partial Z_{MUCSUEED}(\mathbf{b})}{\partial b_i} = v_i - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} \quad (\text{A7.6})$$

The Hessian can be obtained by differentiating (A7.6) once more w.r.t. the common part of a generic link cost:

$$\begin{aligned} \frac{\partial^2 Z_{MUCSUEED}(\mathbf{b})}{\partial b_i \partial b_j} &= \frac{\partial}{\partial b_j} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \delta_{ij} - \sum_K \alpha^{(k)} \sum_{RS} \frac{\partial D_{rs}^{(k)}(S_{rs}^{(k)})}{\partial b_j} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \frac{\partial P_{rsp}^{(k)}}{\partial b_j} \delta_{ip}^{(k)} \end{aligned} \quad (\text{A7.7})$$

where δ_{ij} is 1 if $i=j$ and 0 otherwise. Considering again (A7.1), (A7.2) and (A7.3), (A7.7) can be rewritten as:

$$\begin{aligned} \frac{\partial^2 Z_{MUCSUEED}(\mathbf{b})}{\partial b_i \partial b_j} &= \frac{\partial}{\partial b_j} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \delta_{ij} - \sum_K \alpha^{(k)} \sum_{RS} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \frac{\partial S_{rs}^{(k)}}{\partial b_j} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \delta_{ip}^{(k)} \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \frac{\partial e_m^{(k)}}{\partial b_j} \end{aligned} \quad (\text{A7.8})$$

$$\begin{aligned} &= \frac{\partial}{\partial b_j} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \delta_{ij} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} \frac{dD_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \sum_M \frac{\partial S_{rs}^{(k)}}{\partial e_m^{(k)}} \frac{\partial e_m^{(k)}}{\partial b_j} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} + \\ &\quad - \sum_K \alpha^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \delta_{ip}^{(k)} \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \beta^{(k)} \delta_{jm} \end{aligned}$$

and, finally:

$$\begin{aligned}
\frac{\partial Z^2_{MUCSUEED}(\mathbf{b})}{\partial b_i \partial b_j} &= \frac{\partial}{\partial b_j} \left(b_i^{-1} \left(\frac{c_i^{(1)} - c_{0i}^{(1)}}{\beta^{(1)}} \right) \right) \delta_{ij} + \\
&- \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} \frac{dq_{rs}^{(k)}(S_{rs}^{(k)})}{dS_{rs}^{(k)}} \sum_M P_{rsm}^{(k)} \delta_{jm}^{(k)} \sum_P P_{rsp}^{(k)} \delta_{ip}^{(k)} + \\
&- \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} q_{rs}^{(k)}(S_{rs}^{(k)}) \sum_P \delta_{ip}^{(k)} \sum_M \frac{\partial P_{rsp}^{(k)}}{\partial e_m^{(k)}} \delta_{mj}^{(k)}
\end{aligned} \tag{A7.9}$$

which can also be rewritten in matrix terms as:

$$\begin{aligned}
\nabla^2 z(\mathbf{b})_{MUCSUEED} &= \nabla_b \mathbf{b}^{-1} + \sum_K \alpha^{(k)} \sum_{RS} \left(-\frac{dD_{rs}^{(k)}}{dS_{rs}^{(k)}} \right) (\Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)}) (\Delta_{rs}^{(k)} \mathbf{P}_{rs}^{(k)})^T + \\
&+ \sum_K \alpha^{(k)} \beta^{(k)} \sum_{RS} D_{rs}^{(k)}(S_{rs}^{(k)}) (\Delta_{rs}^{(k)} (-\nabla_{e^{(k)}} \mathbf{P}_{rs}^{(k)}) \Delta_{rs}^{(k)T})
\end{aligned} \tag{A7.10}$$

where $\mathbf{P}_{rs}^{(k)}$ is the vector of the path choice probabilities for the paths between r and s and for class k , and $\nabla_{e^{(k)}} \mathbf{P}_{rs}^{(k)}$ is its Jacobian matrix.

**PUBLISHED
PAPER NOT
INCLUDED**