A multi-objective linear programming model for scheduling part families and designing a group layout in cellular manufacturing systems

Abstract

Different industries compete to attract customers in different ways. In the field of production, group technology (GT) is defined by identifying and grouping similar parts based on their similarities in design and production. Cellular manufacturing (CM) is an application of GT to reconfigure the factory and job shop design. A manufacturing cell is a group of independent machines with distinct functions put together to produce a family of parts. Designing a cellular manufacturing system involves three major decisions: cell formation (CF), group layout (GL), and group scheduling (GS). Although these decisions are interrelated and can affect each other, they have been considered separately or sequentially in previous research. In this paper, CF, GL, and GS decisions are considered simultaneously. Accordingly, a multi-objective linear programming (MOLP) model is proposed to optimize weighted completion time, transportation cost, and machine idle time for a multi-product system. Finally, the model will be solved using the ε-constraint method, representing different scales solutions for decision-making. The proposed model is NP-hard. Therefore, a multi-objective genetic algorithm (MOGA) has been presented to solve it since GAMS software is unable to find optimal solutions for large-scale problems.

Keywords: group technology, cell formation, group scheduling, group layout, cellular manufacturing system, multi-objective linear programming model

1- Introduction

Group technology (GT) is a production system that allows companies to minimize work in progress, ordering time, and costs while producing a variety of products. By combining different equipment and machines into a group or cell, the entire responsibility for producing a set of parts can be delegated to a particular group. Cell manufacturing (CM) is one of the most important applications of GT used to reconfigure factory and job shop design. Cellular manufacturing systems (CMS) include identifying the family of parts and cells of machines so that intracellular transportation is minimized and the use of machines within a cell is maximized (Chang, Wu, & Wu, 2013). CMS is one of the most important tools to improve flexibility in any production planning (Aryanezhad, Aliabadi, & Tavakkoli-Moghaddam, 2011). It shortens manufacturing lead times, restricts intercellular movements to a significant extent, and reduces setup time.

CM aims to maintain the advantages of other methods using classifications of machines and parts. A production line system is a method in which products move sequentially along the line. Although this method lacks production flexibility as it produces only one product by a product manufacturing system and machines on a specific path, materials scheduling and control are easy. Figure 1 shows an overview of the product line system.

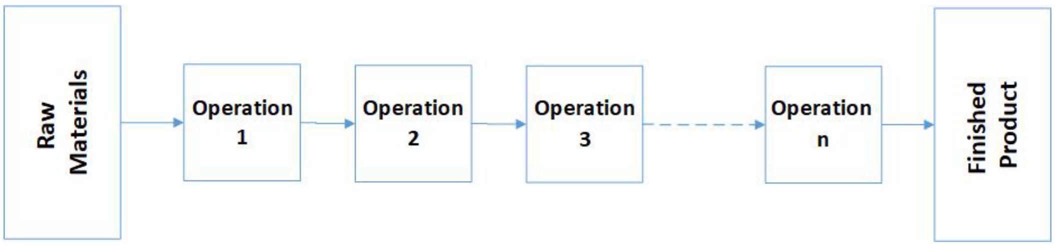

Figure 1: Layout of the product line system

On the other hand, there are job shop production systems with features that are in many ways the opposite of production line systems. The job shop production system has a low production rate and high flexibility. Material control and scheduling are problematic because, after processing on one machine, parts may move a long distance to the next machine. Figure 2 shows a typical layout of the job shop production system.

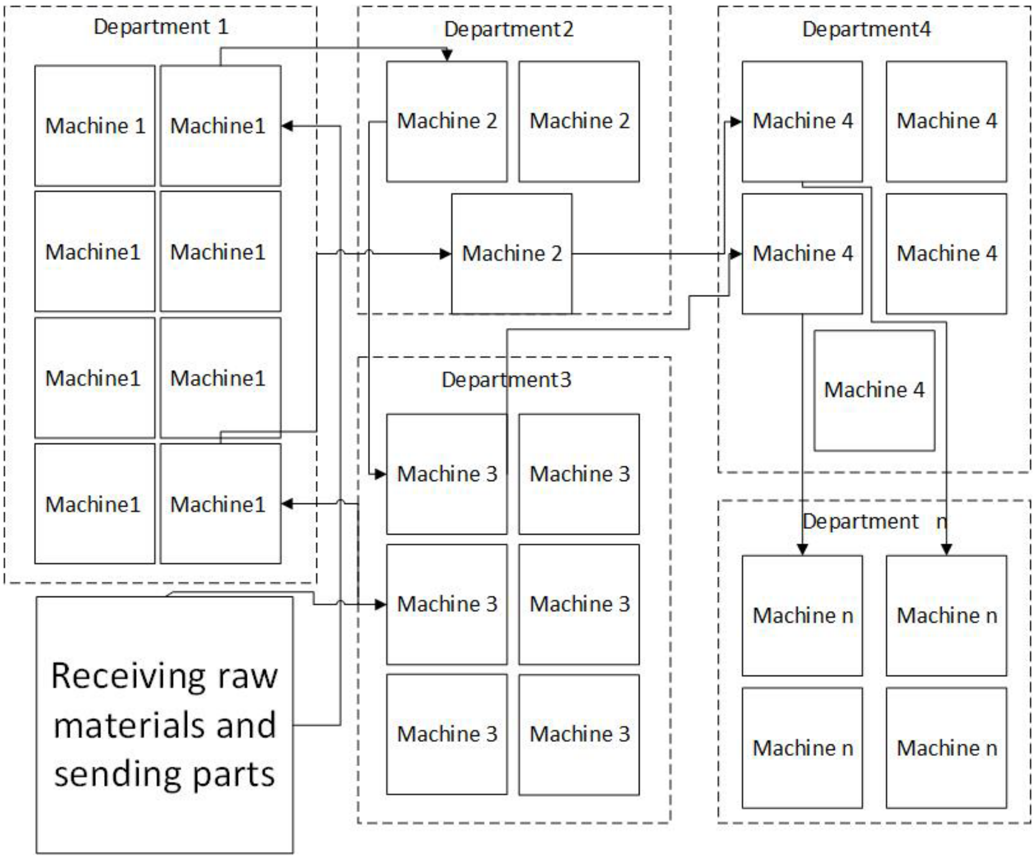

Figure 2: Overview of job shop production system

A third type of production system is cellular manufacturing system. CMS is used when both production volume and product variety are at a medium level. This is more economical than the other two systems. In addition, it improves the material flow and timing of the job shop production system and the flexibility of the production line system. An overview of the cell production system is shown in Figure 3.

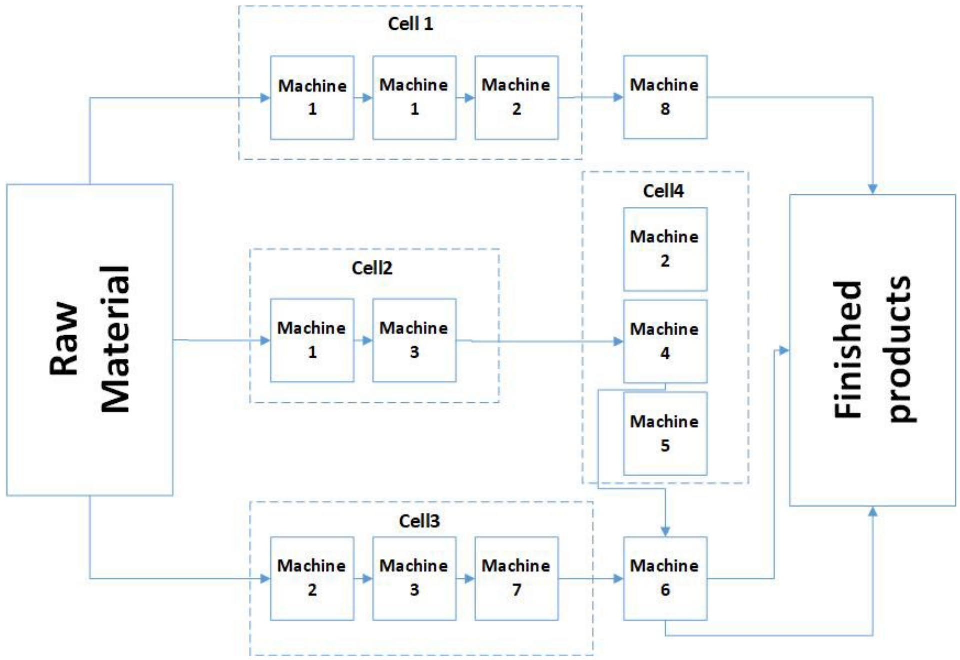

Figure 3: Layout of a cellular manufacturing system

Formation, layout, and scheduling of cells are three essential steps to consider when designing a CMS (Wu, Chu, Wang, & Yue, 2007). Cell formation (CF) comprises recognizing machine cells and part families to minimize the total cost of parts' intercellular movement. Group layout (GL) addresses the issues of determining the cell layout in the floor plan and the machine layout within each cell. The most widespread goal for GL is to minimize transportation costs (Hassan, 1995). Finally, cell management deals with design issues such as group scheduling (GS), which schedules individual parts and families of parts within each cell (Wu, Chu, Wang, & Yue, 2007). The communication between these three decisions plays a vital role in CMS design. CF directly depends on the cost of transporting the parts determined, based on the cell and machine layouts. Additionally, when the cell size becomes smaller, intercellular movements increase, while larger sizes complicate cell management and scheduling. Thus, transportation time is affected by the cells' layout and the machines in the cells. Accordingly, in the recent years, several studies have proposed integration of two or three of these decisions. The integrated problem has usually been formulated as a non-linear model (Alireza Goli, Tirkolaee, & Aydın, 2021; Jufeng Wang, Liu, & Zhou, 2021). Solving these problems is more complicated than linear models.

This paper aims to provide a multi-objective linear programming (MOLP) model to resolve cell formation, group scheduling, and group layout issues, optimizing the objective functions of completion time, machine idle time (efficiency), and transportation time of parts, by considering the maximum and minimum cell constraints and maximum machines per cell.

The remainder of this paper is organized as follows. Section 2 provides the literature review. The problem is defined and formulated in Section 3. Section 4 explains the proposed exact solution method and metaheuristic algorithm. In Section 5, the performance of the ε-constraint and the NSGA-II algorithm are assessed through solving numerical examples from the literature, and the results are presented and discussed. In section 6, important parameters are identified, and sensitivity analysis is performed to validate the model. Finally, Section 7 draws conclusions and future study directions.

2- Review of current literature

The GL decision has been addressed in previous literature, but most researchers have focused on CF and GS problems sequentially and separately. Since the first step in designing a CMS is forming cells, there is no need for a prior solution to the other two decisions (Wei & Gaither, 1990). However, most researchers have supposed that the CF solution can address the GS problems (Elmi, Solimanpur, Topaloglu, & Elmi, 2011; Solimanpur & Elmi, 2013; Zuberek, 1996). Recently, CF, GL, and GS decisions have been considered simultaneously in several studies. The integration of these decisions has been demonstrated to improve CMS design and operational performance.

In some studies, CF has been considered separately. Wei and Gaither (1990) studied the problem of cell formation wherein the goal is minimizing the density in the cells of machines with long waiting lines. Vin, De Lit, and Delchambre (2005) then investigated cell formation with the aim of part traffic using a genetic grouping algorithm (GGA). They designed hybrid cell production systems in two resource-constrained environments. The problem is divided into three stages: first, the identification of part demand and demand diversity using Pareto analysis; second, machinery grouping; and last, labor allocation. They also proposed a goal programming model for this problem. Later, Paydar and Saidi-Mehrabad (2013) investigated cell formation to maximize the effect of grouping. They evaluated a linear programming model of deficit development and the effectiveness of this model, by considering two tests from the literature review. Later still, Mehdizadeh et al. (2020) proposed a nonlinear programming model to integrate CF and production planning problems in a dynamic cellular manufacturing system, where resources for setting up cells are limited. The formation of machine-part families is a major task in CMS.

The issue of integrating CF and GL decisions has been addressed in several studies. In the beginning, researchers focused on solving this problem using mathematical concepts. Rajagopalan and Batra (1975) tackled cell formation and transportation issues between cells using graph theory. Later, the emergence of metaheuristic methods is seen in De Lit, Falkenauer, and Delchambre (2000) and Brown and Sumichrast (2001). In their research, cell formation and part family formation are investigated to minimize the traffic of items. They used the grouping genetic algorithm (GGA) to solve their model. Subsequently, Uddin and Shanker (2002) explained generalized groups wherein each part has one or more process routes. The goal here is to minimize the number of visits of different cells and movement between cells. Jun Wang (2003) then examined cell formation by looking at the location of cells and used a heuristic method to solve the problem.

Since 2003, mathematical models have been presented due to a rise in competitiveness and thus interest in CMS. Cao and Chen (2004) proposed a mixed nonlinear mathematical model that has optimized cell formation and cell setup in a sequential metaheuristic method. Virtual cellular manufacturing can be a solution for CMS and functional layout system problems concurrently.

Therefore, Arora, Haleem, Kumar, and Khan (2020) have more recently discussed various models of the formation of virtual cells and factors that can affect the efficiency of production systems. A fuzzy multi-objective mathematical model for a CMS under dynamic conditions has just been presented by Mohtashami, Alinezhad, and Niknamfar (2020). The optimal layout design in each production period can be determined in their model.

The group scheduling (GS) problem has traditionally been solved in the second step after solving the CF problem. Chandrasekharan and Rajagopalan (1993) targeted exceptional elements and intercellular movement and have used multi-dimensional scaling (MDS) for the solution. Tavakkoli-Moghaddam et al. (2008) presented a GS problem for manufacturing cells, where parts can access other cells. Two evolutionary algorithms, one genetic algorithm (GA), and the other memetic algorithm (MA) have been presented and examined in obtaining the makespan and objective function value. Then, Reza Tavakkoli-Moghaddam, Javadian, Khorrami, and Gholipour-Kanani (2010) dealt with intracellular scheduling to determine the sequence of parts within production cells. However, in intercellular programming, a sequence of cells has been obtained. Their study developed a novel mathematical model for the multi-criteria GS problem in a CMS. Ghezavati and Saidi-Mehrabad (2010) presented a mathematical model for cell manufacturing with group planning in an uncertain space. This model has optimized CF and GS simultaneously and has minimized the total expected cost, including maximum tardiness cost among all parts, subcontracting cost for distinctive elements, and resource cost. A fuzzy Mixed Integer Linear Programming (MILP) model has more recently been presented by A. Goli, Tirkolaee, and Aydin (2021), designed for CF problems, including scheduling the parts within cells in a CMS. J. S. Neufeld, F. F. Teucher, and U. Buscher (2020) found certain characteristics that mark out cellular scheduling distinctiveness. Furthermore, they introduced a novel cell-based objective, namely total cell makespan.

Among the research on cell manufacturing issues, we have carefully examined the research done on cell formation, group layout, and group scheduling. A mathematical model that integrates these three decisions and develops a hierarchical genetic algorithm (HGA) was provided by Wu, Chu, Wang, Yue, and Engineering (2007) to solve the integrated cell design problem. Their model offers better solutions than the sequential one and better performance of the procedures using the proposed hierarchical operators. In their model, they optimize the completion time and consider cell formation and layout of machines in the constraint structure. Arkat, Farahani, and Hosseini (2012) investigated the integrated design of CM systems by presenting two mathematical models. The first combines the cellular design with CF to specify the best cell configuration and cell design to minimize total transportation costs. The second also considers cellular planning to reduce time. Arkat, Farahani, and Ahmadizar (2012) proposed a mathematical model for simultaneously identifying cell formation, cell design, and sequencing operations, to minimize the total cost of transporting parts and minimizing makespan concurrently. The proposed multi-objective genetic algorithm (MOGA) solved the model by finding Pareto optimal solutions. Forghani and Fatemi Ghomi (2020) solved the integration of these three decisions in cellular manufacturing systems, by considering alternative processing routes to minimize cycle time and total cost. A mathematical model has recently been presented to investigate the integrated CF, GL, and GS in a CMS by Arkat, Rahimi, and Farughi (2021). Their objective is to handle the arrival of a new job as a disturbance to the system. Table 1 overviews several papers about CM.

Table 1: Comparison among scholarly papers related to CMS

Papers	Decisions			Objective function			Solving approach		Product number		Method
	CF	GL	GS	Completion time	Transportation cost	Machinery	Sequential	Concurrent	Single	Multi	
(Chandrasekharan & Rajagopalan, 1993)	✓		✓				✓		✓		non-metric Multi-Dimensional Scaling (MDS).
(Tsai, Chu, & Barta, 1997)	✓		✓		✓			✓	✓		Fuzzy Mixed-Integer Programming (FMIP)
(Wu, Chu, Wang, & Yue, 2007)	✓	✓	✓	✓				✓	✓		Hierarchical Genetic Algorithm (HGA)
(R Tavakkoli-Moghaddam, Gholipour-Kanani, & Cheraghalizadeh, 2008)	✓		✓	✓		✓			✓		Genetic Algorithm (GA) and Memetic Algorithm (MA)
(Ghezavati & Saidi-Mehrabad, 2010)	✓		✓	✓		✓		✓	✓		Genetic Algorithm (GA) and Simulated Annealing (SA) algorithm
(Mahdavi, Paydar, Solimanpur, & Saidi-Mehrabad, 2010)	✓	✓		✓		✓		✓	✓		Exact solution by LINGO
)Arkat, Farahani, & Hosseini, 2012(	✓	✓	✓	✓	✓		✓		✓		GA
)Arkat, Farahani, & Ahmadizar, 2012(	✓	✓	✓	✓	✓		✓		✓		Multi-Objective Genetic Algorithm (MOGA)

)Bayram & Şahin, 2016(	✓	✓			✓	✓	✓		✓		Genetic Algorithm (GA) and Simulated Annealing (SA) Algorithm
)Rahimi, Arkat, & Farughi, 2020(	✓	✓	✓	✓		✓	✓		✓		GA and Ant Lion Optimizer (ALO) algorithm
)Shafiee-Gol, Kia, Tavakkoli-Moghaddam, Kazemi, & Kamran, 2021(	✓		✓		✓	✓		✓	✓		CPLEX and Simulated Annealing (SA) algorithm
This paper	✓	✓	✓	✓	✓	✓		✓		✓	ε-constraint and NSGA-II

Investigating the available resources in the field of CMS reveals that works considering the three decisions of CF, GL, and GS are limited. Some of the shortcomings in the existing studies are discussed below:

1. Previous research into CMS has often examined the completion time and transportation of parts (Arkat, Hosseinabadi Farahani, & Hosseini, 2012; Arkat et al., 2021). However, given competition, optimizing machinery costs can mean greater profitability for companies. In this study, the idle time of machines in production systems is examined and optimized.
2. In the reviewed articles, CMS has been considered for the production of parts (Alimian, Ghezavati, & Tavakkoli-Moghaddam, 2020; Shafiee-Gol et al., 2021). However, products can contain one or multiple parts. In our model, the manufacturing system is intended to produce several products, and each product can be composed of one or more parts.
3. Completion time in previous research is often considered to be the sum, average, or maximum completion time of parts, and there was no difference among parts (J. S. Neufeld, F. Teucher, & U. Buscher, 2020). However, the multi-product approach in this research considers the completion time as the weighted average of the products' completion time, which can be determined according to the product share of parts, revenue, and production system costs.
4. In the reviewed papers, the dimensions of machines and floor plans are neglected in modeling. To simplify the problem, researchers have applied two approaches. Some assume that the intercellular layout is pre-specified (Ebrahimi, Kia, & Komijan, 2016; Janis S Neufeld et al., 2020). Others assume the layout of the machine to be a location, with the physical dimensions of the machines not considered (Khamlichi, Oufaska, Zouadi, & Dkiouak, 2020; Shafiee-Gol et al., 2021). However, in this model, the longitudinal and transverse dimensions of machines are considered together with their continuous location within the floor plan of a manufacturing system.

5. In previous research, each part operation is performed on a specific machine (Arkat, Farahani, & Ahmadizar, 2012; Liu, Wang, Leung, & Li, 2016; Shafiee-Gol et al., 2021). Yet in this model, machines can perform specific operations according to their capabilities, and each part is assigned to one of the machines to operate, which is superior in terms of cost and time.

To overcome the abovementioned shortcomings, this paper develops a mathematical model to provide a formulation of CM design that considers cell formation, group layout, and group scheduling. In our model, the manufacturing system must produce several products, whilst minimizing total completion time, transportation cost, and machine idle time between operations. To validate and compare the results, two methods of the exact solution and metaheuristic solution are used to solve this model. GAMS software is used for the exact solution by the ε-constraint method. In addition, a non-dominated sorting genetic algorithm II (NSGA-II) with MATLAB software is used for the metaheuristic solution. Then, we validate our model by comparing the results of the NSGA-II and ε-constraint methods. Since CM is an NP-hard problem, NSGA-II is used to solve the model in large-scale problems.

3- Methodology

3-1 Problem description

Cell manufacturing systems (CMS) are an important application of group technology (GT). Cell manufacturing (CM) involves processing similar parts into a specific machine group or production process. Therefore, it is usually best to assign a cell to a family of parts, each of which is preferably produced completely within that cell, and the cells in a CMS have minimal interaction. The current model aims to aid decisions on cell formation, group layout, and group scheduling, so as to minimize total completion time, transportation cost, and machine idle time between operations.

Assumptions of the model were as listed below:

- A CMS encompasses several products, each of which consists of one or more parts.
- Each part requires one or more operations in a specific sequence.
- A CMS has machines that each can perform one or more operations.
- The dimensions of the general design of the manufacturing system and the dimensions required for each machine are specified.
- The number of cells required is known.
- The maximum and minimum number of machines inside a cell are known (with at least one machine).
- The cost of intracellular and intercellular transportation is a function of their transportation time, which itself is a function of distance.
- Cells have rectangular dimensions.

3-2 Nomenclature

3-2-1 Indices and sets

i: product type i = 1, 2, …, G
j: part type $j = 1, 2, \dots, P_i$
k: cell type k = 1, 2, …, C
l, u, v: machine type l, u, v =1, 2, …, m
o: operation type $o = 1, \dots, N_{ij}$

3-2-2 Parameters

G: number of products

P_i: number of parts of product i

C: number of cells to be performed

m: number of machines

N_{ij}: number of operations of part j of product i

R: maximum number of machines allowed in each cell

H: horizontal dimension of the floor plan

V: vertical dimension of the floor plan

H_l: horizontal dimension required by machine l

V_l: vertical dimension required by machine l

t_{ijo}: processing time of operation o on part j of product i

CA_{ij}: intercellular transfer unit cost for part j of product i

CE_{ij}: intracellular transfer unit cost for part j of product i

A_{ij}: intercellular unit conversion of distance to time for part j of product i

E_{ij}: intracellular unit conversion of distance to time for part j of product i

M: an appropriately large number

α_i: the profitability ratio of product i

$\acute{U}_{ijol}$: 1, if operation o of part j should be processed on machine l for product i, and 0 otherwise

3-2-3 Decision variables

g_{ijo}: completion time of operation o on part j of product i

$g_{(ij)}$: completion time of part j of product i

$g_{(i)}$: completion time of product i

x_l: horizontal distance between the center of machine l and the vertical reference line

y_l: vertical distance between the center of machine l and the horizontal reference line

$C_{ij(o,ó)}$: intra- or intercellular transportation cost for part j of product i between operations o and $ó$

$TA_{ij(oó)}$: intercellular transportation time for part j of product i between operations o and $ó$

$TE_{ij(oó)}$: intracellular transportation time for part j of product i between operations o and $ó$

uw_l: idle time of machine l

U_{ijol}: 1, if operation o on part j of product i should be processed on machine l, and 0 otherwise.

$w_{i\acute{j}\acute{o}}^{ijo}$: 1, if operation o of part j of product i precedes operation $ó$ of part j' of product i′, and 0 otherwise

z_{lk}: 1, if machine l is assigned to cell k, and 0 otherwise

p_{luv}: 1, if $x_l < x_v < x_u$ or $x_u < x_v < x_l$, and 0 otherwise

q_{luv}: 1, if $y_l < y_v < y_u$ or $y_u < y_v < y_l$, and 0 otherwise

3-3 Mathematical model

$$Min \sum_{i=1}^{G} \alpha_i g_{(i)} \quad (1)$$

$$Min \sum_{i=1}^{G} \sum_{j=1}^{P_i} \sum_{o=1}^{N_j-1} C_{ij(o,o+1)} \quad (2)$$

$$Min \sum_{l=1}^{m} uw_l \quad (3)$$

S.t.

$$\sum_{k=1}^{C} z_{lk} = 1 \qquad l = 1, \dots, m \quad (4)$$

$$\sum_{l=1}^{m} z_{lk} \le R \qquad k = 1, \dots, C \quad (5)$$

$$\sum_{l=1}^{m} z_{lk} \ge 1 \qquad k = 1, \dots, C \quad (6)$$

$$|x_l - x_u| \ge \frac{H_l + H_u}{2} \qquad u = l+1, \dots, m \quad l = 1, \dots, m-1 \quad (7)$$

$$|y_l - y_u| \ge \frac{V_l + V_u}{2} \qquad u = l+1, \dots, m \quad l = 1, \dots, m-1 \quad (8)$$

$$(x_v - x_l)(x_u - x_v) < M(p_{luv}) \qquad \forall l, u, v \quad (9)$$

$$-(x_v - x_l)(x_u - x_v) \le M(1 - p_{luv}) \qquad \forall l, u, v \quad (10)$$

$$(y_v - y_l)(y_u - y_v) < M(q_{luv}) \qquad \forall l,u,v \qquad (11)$$

$$-(y_v - y_l)(y_u - y_v) \le M(1 - q_{luv}) \qquad \forall l,u,v \qquad (12)$$

$$z_{lk} + z_{uk} + p_{luv} + q_{luv} - 3 \le z_{vk} \qquad \forall l,u,v,k \qquad (13)$$

$$C_{ij(o\acute{o})} = CA_{ij}TA_{ij(o\acute{o})} + CE_{ij}TE_{ij(o\acute{o})} \qquad \forall i,j \quad o = 1,\dots,N_j - 1 \quad \acute{o} = o + 1 \qquad (14)$$

$$TA_{ij(o\acute{o})} = U_{ijol}U_{ij\acute{o}u}z_{lk}z_{uk}(|x_l - x_u| + |y_l - y_u|)A_{ij} \qquad \forall i,j \quad o = 1,\dots,N_j - 1 \quad \acute{o} = o + 1 \qquad (15)$$

$$TE_{ij(o\acute{o})} = U_{ijol}U_{ij\acute{o}u}z_{lk}(1 - z_{uk})(|x_l - x_u| + |y_l - y_u|)E_{ij} \qquad \forall i,j \quad o = 1,\dots,N_j - 1 \quad \acute{o} = o + 1 \qquad (16)$$

$$g_{ijo} - g_{ij\acute{o}} \ge t_{ij\acute{o}} + TA_{ij(o\acute{o})} + TE_{ij(o\acute{o})} \qquad \forall i,j,o,\acute{o} \qquad (17)$$

$$g_{ijo} - g_{\acute{i}j\acute{o}} + Mw_{\acute{i}j\acute{o}}^{ijo} \ge t_{ijo} \qquad \forall i,j,\acute{i},\acute{j} \quad (o,\acute{o}) \in S_l \qquad (18)$$

$$g_{\acute{i}j\acute{o}} - g_{ijo} + M\left(1 - w_{\acute{i}j\acute{o}}^{ijo}\right) \ge t_{\acute{i}j\acute{o}} \qquad \forall i,j,\acute{i},\acute{j} \quad (o,\acute{o}) \in S_l \qquad (19)$$

$$g_{ijo} \ge t_{ijo} \qquad \forall i,j,o \qquad (20)$$

$$g_{(ij)} \ge g_{ijo} \qquad \forall i,j,o \qquad (21)$$

$$g_{(i)} \ge g_{(ij)} \qquad \forall i,j \qquad (22)$$

$$U_l \ge g_{ijo}U_{ijol} \qquad \forall i,j,o,l \qquad (23)$$

$$x_l \le H - \frac{h_l}{2} \qquad l = 1,\dots,m \qquad (24)$$

$$x_l \ge \frac{h_l}{2} \qquad l = 1,\dots,m \qquad (25)$$

$$y_l \le V - \frac{v_l}{2} \qquad l = 1,\dots,m \qquad (26)$$

$$y_l \ge \frac{v_l}{2} \qquad l = 1,\dots,m \qquad (27)$$

$$\sum_{l=1}^{m} U_{ijol} = 1 \qquad \forall i,j,o \qquad (28)$$

$$U_{ijol} \le \acute{U}_{ijol} \qquad \forall i,j,o,l \qquad (29)$$

$$uw_l = U_l - \sum_i^G \sum_j^{P_i} \sum_o^{N_{ij}} U_{ijol} \times t_{ijo} \qquad (30)$$

$$z_{lk}, p_{luv}, q_{luv}, U_{ijol}, w_{\acute{i}j\acute{o}}^{ijo} = 0 \text{ or } 1 \qquad (31)$$

The purposes of the equations were as follows:

- Equations (1), (2), and (3) were the objective functions, which, respectively, minimized the total product completion times, total transportation cost of parts, and the idle time of machines.

Equation (4) ensures that every machine is assigned to only one cell. Equation (5) ensures that no more than R machines are assigned to each cell. Equation (6) enforced that at least one machine was assigned to each cell. The required space of machines was considered rectangular, and x and y were its centre. Constraints (7) and (8) imposed the required dimensions of the machines. Constraints (9) to (13) stopped cells from overlapping and forced them to be rectangular, and if the vertical and horizontal coordinates of a machine were between two machines of a cell, then the machine must be part of that cell.

Transportation costs were calculated by Equation (14) according to time and type of transportation (intercellular and intracellular). Depending on the coordinates of the machines and the sequence of operations on the machine, Equations (15) and (16) calculated intercellular and intracellular transportation times for part j between operations o and $o^{'}$.

The processing of each part on machines is ensured by constraint (17) according to the priority of the operations. Constraints (18) and (19) imposed that at a time unit only one part could be processed by each machine. Constraint (20) made sure that the processing time of operation o of part j of product i was less than or equal to its completion time and constraint (21) ensured that the completion time of each operation must be less than or equal to the total completion time. The makespan was at least the completion time of each part was ensured by Equation (22). Equation (23) limited the completion time of machine operations. Constraints (24) to (27) forced machines to be located within the boundaries of the floor plan. Equation (28) ensured each operation was performed on only one machine. Equation (29) forced operations to be assigned to machines where such operations were feasible. Equation (30) calculated the idle time of each machine.

- Equation (31) specified that the decision variables are binary.

When $x_l \leq x_v \leq x_u$ or $x_u \leq x_v \leq x_l$, constraints (9) and (10) forced p_{luv} to equal 1. Similarly, when $y_l \leq y_v \leq y_u$ or $y_u \leq y_v \leq y_l$, constraints (11) and (12) forced q_{luv} to equal 1. Using equation (10), if p_{luv} = 1 and q_{luv} = 1 and machines l and u were assigned to cell k, then machine v should be in cell k. Hence, machines were in rectangular spaces of non-overlapping cells.

3-4 Linearization

In this section, some nonlinear equations are linearized. There are some absolute terms in Eqs. 7,8 and 15,16 and the product of decision variables in Eqs. 9-12 and Eq. 15,16 that causes nonlinearity. We can eliminate the absolute terms in Eq. 7 and 8 by changing them to the following four equations:

$$x_l - x_u \geq \frac{H_l + H_u}{2} \qquad u = l+1, \dots, m \quad l = 1, \dots, m-1 \qquad (32)$$

$$x_l - x_u \leq -\left(\frac{H_l + H_u}{2}\right) \qquad u = l+1, \dots, m \quad l = 1, \dots, m-1 \qquad (33)$$

$$y_l - y_u \geq \frac{V_l + V_u}{2} \qquad u = l+1, \dots, m \quad l = 1, \dots, m-1 \qquad (34)$$

$$y_l - y_u \le -\left(\frac{V_l+V_u}{2}\right) \qquad u = l+1,\dots,m \quad l = 1,\dots,m-1 \qquad (35)$$

Since machines have a rectangular dimension, Eqs. 32-35 forced the required dimensions of the machines.

In this section we use the linearization method that Arkat, Hosseinabadi Farahani, et al. (2012) used for their paper. Eqs 9 and 10 forced p_{luv} to equal 1 when terms ($x_v - x_l$) and ($x_u - x_v$) have the same signs (both are negative or positive) and 0 otherwise. b_{vl} is a binary variable which is equal to 1 if ($x_v - x_l$) is positive and 0 otherwise, Eqs. 9 and 10 can be converted into the following equations:

$$(x_v - x_l) < M(b_{vl}) \qquad \forall l, v \qquad (36)$$

$$-(x_v - x_l) \le M(1 - b_{vl}) \qquad \forall l, v \qquad (37)$$

$$(x_u - x_v) < M(b_{uv}) \qquad \forall u, v \qquad (38)$$

$$-(x_u - x_v) \le M(1 - b_{uv}) \qquad \forall u, v \qquad (39)$$

$$b_{vl} + b_{uv} - 1 \le p_{luv} \qquad \forall l, u, v, k \qquad (40)$$

$$-b_{vl} - b_{uv} + 1 \le p_{luv} \qquad \forall l, u, v, k \qquad (41)$$

$$b_{vl} - b_{uv} + 1 \ge p_{luv} \qquad \forall l, u, v, k \qquad (42)$$

$$-b_{vl} + b_{uv} + 1 \ge p_{luv} \qquad \forall l, u, v, k \qquad (43)$$

Equations 40-43 ensured that p_{luv} is equal to 1 when b_{vl} and b_{uv} are either both 1 or both 0 and otherwise is equal to 0.

The linearization procedure of equations 11 and 12 is similar as was done with Eqs. 9 and 10.

At last, Eq. 15 is transformed with the following equation:

$$TA_{ij(o\acute{o})} = U_{ijol} U_{ij\acute{o}u} F_{luv} A_{ij} \qquad \forall i, j \quad o = 1,\dots,N_j - 1 \quad \acute{o} = o + 1 \qquad (44)$$

and following equations are added to the constraints of the model:

$$z_{lk} + z_{uk} - 2 \ge \mathrm{M}(1 - t_{luk}) \qquad \forall l, u, v, k \qquad (45)$$

$$z_{lk} + z_{uk} - 1 \le \mathrm{M} t_{luk} \qquad \forall l, u, v, k \qquad (46)$$

$$F_{luv} \ge (x_l - x_u) + (y_l - y_u) + \mathrm{M}(1 - t_{luk}) \qquad \forall l, u, v, k \qquad (47)$$

$$F_{luv} \ge (x_l - x_u) - (y_l - y_u) + \mathrm{M}(1 - t_{luk}) \qquad \forall l, u, v, k \qquad (48)$$

$$F_{luv} \ge -(x_l - x_u) + (y_l - y_u) + \mathrm{M}(1 - t_{luk}) \qquad \forall l, u, v, k \qquad (49)$$

$$F_{luv} \ge -(x_l - x_u) - (y_l - y_u) + \mathrm{M}(1 - t_{luk}) \qquad \forall l, u, v, k \qquad (50)$$

where t_{luk} is a binary variable that equals to 1 when z_{lk} and z_{uk} both are equal to 1 and otherwise equals to 0. When t_{luk} equals 1, Eqs. 45-50 ensure that $F_{luv} \ge |x_l - x_u| + |y_l - y_u|$.

Similar ways should be performed for Eq. 16.

4- Solution methodology

4-1 NSGA-II algorithm

The non-dominated sorting genetic algorithm (NSGA) method is a common method for several objective function problems based on genetic algorithms. The main criticisms of the NSGA approach are its lack of sophistication and high computational complexity of non-dominated sorting. These limitations have been addressed in an improved version of NSGA, called NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002). This method has attracted a lot of attention in recent years (Akhmet, Hare, & Lucet, 2022; El Yaagoubi, Charhbili, Boukachour, & El Hilali Alaoui, 2022). The implementation steps of the NSGA-II algorithm were as follows:

1- Generate a random initial answer with a population of size i = 1…, pop size, and set k = 1 (k = the number of NSGA-II algorithm repetitions).

2- Arrange the solutions based on their domination and divide them into fronts. A smaller number of fronts means more solutions of that front have dominated a larger number of other solutions. To do this, the following steps were performed for each of the solutions, such as p:

2.1- Consider S_P as a set of solutions that solution p dominates and consider $S_P = \emptyset$

2.2- Consider N_P as the number of solutions that dominate solution p and consider $N_P = \emptyset$

2.3- For each member of the population n=1…, *pop size* such as q and p do the following steps:

2.3.1- If p dominates q, then add q to the set S_P.

2.3.2- If q dominates p, then increment the domination counter of p (N_P).

3- If $N_P = \emptyset$, p belongs to front f_1, in other words, $f_1 = f_1 U(p)$.

4- Continue all the following steps until the number of solutions in front i is not zero ($f_i \neq \emptyset$).
4.1- Define Q as the set of solutions in front $i + 1$ and consider it equal to zero ($Q = \emptyset$). Then do the following steps for each solution p in front f_i.
4.2- For each solution such as q in the set S_P in front f_i, follow these steps. (It should be noted that, S_P is a set of solutions wherein solution p dominates in the previous step).
4.2.1- N_q= N_q-1. This indicates how many times solution q has been dominated.
4.2.2- If $N_q = \emptyset$, q belongs to front f_{i+1}, and then $Q = QU(q)$.
4.3- Add a unit to i ($i = i + 1$).

5- After fronting the solutions based on the different domination levels, to create the next generation, several of them should be selected. In this paper, the binary method was used to determine the solutions. For this purpose, the first two solutions were randomly selected and compared; the better of the two was added to the store. The following two criteria determined the better answers:

5.1- Rank priority: In this priority, the answers with a lower rank or lower front were selected because the solutions in these fronts can dominate most of the other solutions.
5.2- Otherwise, if both solutions were in the same front, then use the criterion crowding distance (CD), which is explained as:

5.2.1- For each front f_i, consider n_i as the number of solutions of that front.

5.2.2- Define d_i as the distance between the solutions of the fronts and set the distance between all solutions as zero ($f_i(d_i) = 0$).

5.2.3- For each solution, such as j in front f_i, consider each of the objective functions of the problem such as m, and perform the following steps:

5.2.3.1- On the front f_i, sort all the solutions based on m.

5.2.3.2- After sorting the solutions, set $I(d_1) = I(d_n) = \infty$ where $I(di)$ is the CD of front i. This is because there is no other solution next to the solutions to cover it. For solutions 2 to n-1, $I(d_k)$ is determined based on the following equations:

$$CD_K = I(d_k)_1 + \cdots + I(d_k)_m \tag{32}$$

$$I(d_k).m = \frac{I(k+1).m - I(k-1).m}{f_m^{max} - f_m^{min}} \tag{33}$$

In Equation (32), the $I(d_k)_m$ refers to mth objective function, and to calculate the total CD, $I(d_k)$ must be calculated separately, which is specified in Equation (33). Figure 4 shows how to determine the CD in f_i.

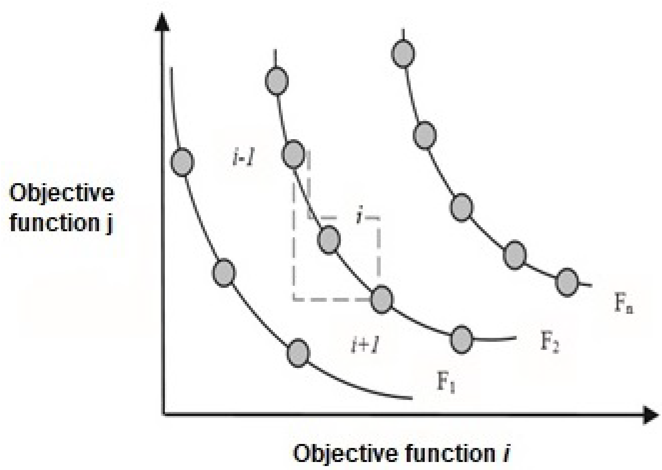

Figure 4: Calculating crowding distance (CD)

5.2.3.3- After calculating the CD, the solution with the greater CD is selected.

6- After the previous step, a pool was created called the selected population. Then genetic operators were utilized to develop the offspring population. The genetic operators used in this paper were crossover and mutation operators. See Figure 5.

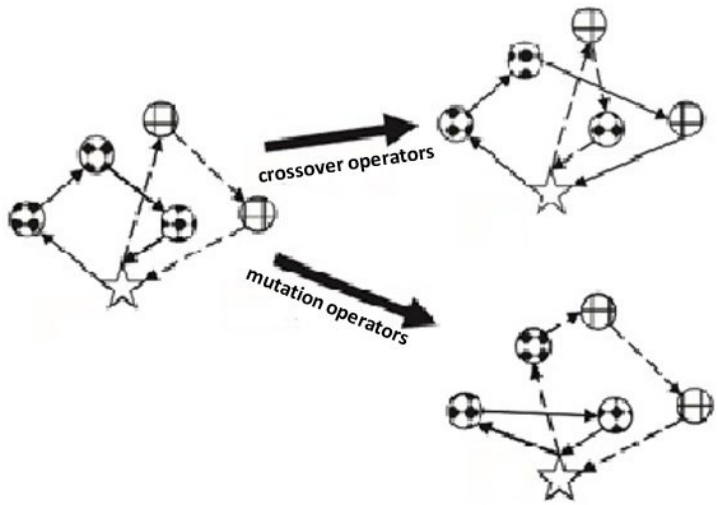

Figure 5: Crossover and mutation operators

7- After determining the offspring population of P_t, this population combined with the parent population (Q_t) (P_t U Q_t). Each pool had a capacity of *n,* and some of the solutions that had combined had to be removed. This was done using the following steps to reach size n.

7.1- First, front the solutions according to the method described in step 2.

7.2- Determine the crowding distance of each solution in its front.

7.3- Start from the f_i front and select its solutions according to the CD and drop them into the new population pool (K + 1). Continue this step until population pool (K + 1) reaches n. see Figure 6.

8- After creating the population (K + 1), the next step was 2, the specified steps were repeated.

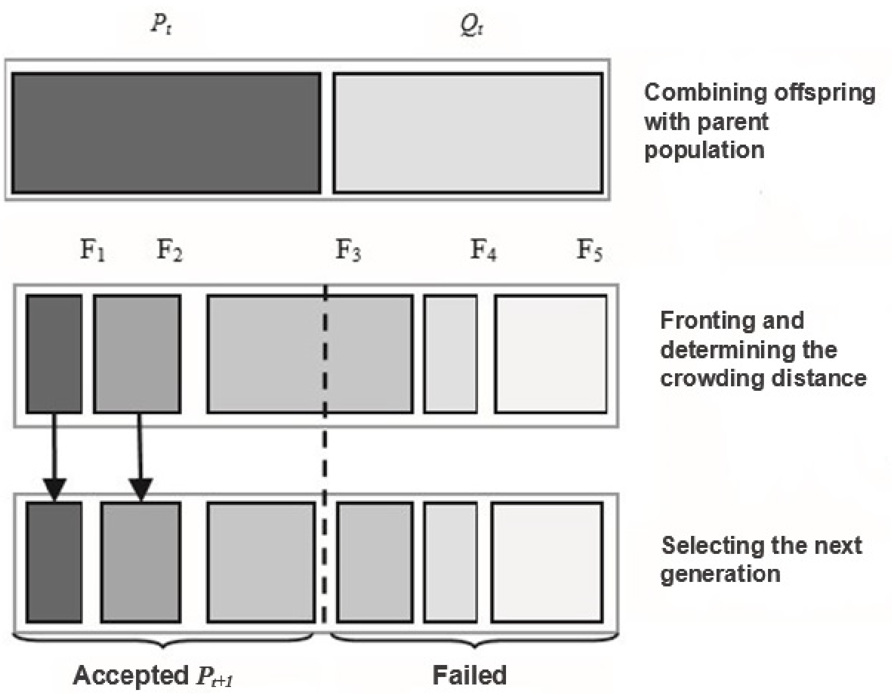

Figure 6: NSGA-II procedure.

4-2 ε-constraint method

The ε-constraint method (K. J. B. Miettinen, Massachusetts, 1999; Vira, Haimes, & Engineering, 1983) is the best-known approach for solving multi-objective problems, according to (Ehrgott & Gandibleux, 2003). This method reformulates the multi-objective optimisation problem by keeping one of the objectives while the rest of the objectives are restricted within user-specified values. In practice, because of the high number of subproblems and the difficulty to establish an efficient variation scheme, this approach has mostly been integrated within heuristic and interactive schemes. The Pareto optimal solutions can be created by the ε-constraint method (Bérubé, Gendreau, & Potvin, 2009). In brief, Pareto optimal solution is defined as a set of 'non-inferior' solutions in the objective space defining a boundary beyond which none of the objectives can be improved without sacrificing at least one of the other objectives (K. Miettinen, 2012).

$$\min f_1(x) \qquad \forall x \in X$$

$$f_2(x) \leq \varepsilon_2$$

$$\dots$$

$$f_n(x) \leq \varepsilon_n$$

The ε-constraint method consists of 5 steps:

1- Consider one of the objective functions (T) as the main objective function.
2- Solve the model based on T and obtain the optimal values.

3- Obtain several values for $\varepsilon_2, \ldots, \varepsilon_n$ and divide the interval between the two optimal values into a determined number.
4- Each time, solve the problem with the primary objective function with any of the values $\varepsilon_2, \ldots, \varepsilon_n$.
5- Report the Pareto answers found.

5- Numerical solution

The model is solved using GAMS and MATLAB in a personal computer with an Intel Core i5 2.40 GHz CPU and 4 GB RAM to find a nondominated solution (Pareto optimal solution). In multi-objective optimization, a solution is nondominated if it outperforms any other solution on at least one criterion (Cheng, Janiak, & Kovalyov, 1998). For instance, solution S1 dominates solution S2 if all of S1's objective values are better than the corresponding objective values of solution S2 and solution S1 is dominated by solution S3 if all of S3's objective values are better than the corresponding objective values of S1 (Guschinsky, Kovalyov, Rozin, Brauner, & Research, 2021; Mahesh, Nallagownden, & Elamvazuthi, 2016). This dominance rule could be formulated as follows:

$$f_i(y) \leq f_i(z) \qquad \forall i \in \{1,2, \ldots, n\}$$

$$f_i(y) < f_i(z) \qquad \exists i \in \{1,2, \ldots, n\}$$

The data used in this study were hypothetical data to solve the problem and also evaluate the performance of the proposed NSGA-II. The main parameters of the model were the number of cells, machines, products, parts, and operations, which determine the scales of the model. These parameters for different scales were as specified in Table 2.

Table 2: Main parameters of the model

Parameter	Small-scale	Medium-scale	Large-scale
Number of cells	2	2	3
Number of machines	3	4	6
Number of products	2	3	4
Number of parts	4	6	8
Number of operations	8	13	20

5-1 Small scale solution using the ε-constraint method

The general parameters of the model were the maximum number of machines allowed in each cell and the dimensions of the floor plan. These parameters were shown in Table 3.

Table 3: General parameters of the model

Parameter	Value
Maximum number of machines allowed in each cell	3
Horizontal dimension of the floor plan	8
Vertical dimension of the floor plan	8

In the model, three machines were considered. Each had a horizontal and vertical dimension to locate it on the floor plan. The parameters were as set out in Table 4.

Table 4: Dimensions of each machine

Parameter	Machine 1	Machine 2	Machine 3
Horizontal dimension of machine	1.2	1.6	1.5
Vertical dimension of machine	1.5	2	1.5

The importance factor of a product is a factor that is determined by the policies and value-added of the product for the company. This coefficient can be determined according to demand, profit, and utility, for the production system. Since the policies of each company can be different, the type of this parameter is general and is not precisely defined. Table 5 sets out the products importance factors.

Table 5: Products importance factor

Parameter	Product 1	Product 2
Importance factor	1.2	1.5

The parts in the cell production system are processed on different machines. After finishing the operation on one machine, a part is moved to the next. Moving parts between machines is costly and time-consuming, depending on the type and size of the part and different movement types according to the number of machines in each cell. As a result, a coefficient for the time and cost of intercellular transfer for each part can be determined. The parameters used in solving the model are shown in Table 6.

Table 6: Coefficient for the time and cost of intracellular and intercellular transfer for each part

Parameter	Part 1	Part 2	Part 3	Part 4
Intercellular transfer cost coefficient	1.5	2	1	1.2
Intracellular transfer cost coefficient	3	4	2	1.2
Intercellular transfer time coefficient	5	7	7	6
Intracellular transfer time coefficient	12	14	14	10

In this problem, some specific operations are carried out on each machine. More specifically, it is assumed that operations 3, and 4 are done on machine 1, operations 2,6, and 7 are performed on machine 2 and operations 1,4, and 8 are done on machine 3. Moreover, it is also assumed that two operations are performed on each part. The processing time of each operation is shown in Table 7.

Table 7: Time of operations

Operati on	Operati on 1	Operati on 2	Operati on 3	Operati on 4	Operati on 5	Operati on 6	Operati on 7	Operati on 8
Part 1	10	12	0	0	0	0	0	0
Part 2	0	0	8	15	0	0	0	0
Part 3	0	0	0	0	11	20	0	0
Part 4	0	0	0	0	0	0	7	10

In this section, the input parameters of the model were specified. Accordingly, the problem was solved by the Epsilon-constraint method using GAMS software. Table 8 illustrates the results. In a multi-objective optimization problem, there is no specific solution that optimizes simultaneously all the objectives. Without any subjective preference, there may be an infinite number of Pareto optimal solutions that can be equally good. Decision makers (DM) can have different goals when studying these problems. The objective may be a priority that satisfies one objective or the trade-offs in satisfying the different objectives (Grimme et al., 2021). For example, in the small scale solution, if the decision maker's priority is minimizing the machine idle time, solution numbers 1, 6, 11, and 16 should be selected (see Table 8). For selecting one of these solutions, the next priority between completion time and transportation cost leads to their choice. Since there are 3 objective functions in our case there are six ways to choose the ideal Pareto optimal solution.

Table 8: Values of objective functions for small scale with the exact solution method

Solution number	Completion time	Transportation cost	Machine idle time
1	224.7	300.4358	81.3
2	224.7	300.4358	82.675
3	212.7	300.4358	84.05
4	212.7	300.4358	85.425
5	212.7	300.4358	86.8
6	207.9	319.9115	81.3
7	207.9	319.9115	82.675
8	204.42	319.9115	84.05
9	198.9	319.9115	85.425
10	198.9	319.9115	86.8
11	207.9	339.3873	81.3
12	207.9	339.3873	82.675
13	204.42	339.3873	84.05
14	198.9	339.3873	85.425
15	198.9	339.3873	86.8
16	194.34	358.863	81.3
17	194.34	358.863	82.675
18	194.34	358.863	84.05
19	191.4	358.863	85.425

20	188.6	358.863	86.8

5-2 Medium-scale solution from the ε-constraint method

The values of the objective functions obtained from the ε-constraint method are shown in Table 9.

Table 9: Values of objective functions for medium scale with the exact solution method

Solution number	Completion time	Transportation cost	Machine idle time
1	167	482	95
2	167	482	101.5
3	167	482	108
4	167	512.5	95
5	167	512.5	101.5
6	167	512.5	108
7	167	543	95
8	167	543	101.5
9	164	543	108
10	167	573.5	95
11	161	573.5	101.5
12	161	573.5	108
13	183	604	88.5
14	167	604	95
15	157	604	101.5
16	154	604	108

5-3 Different scales solution using meta-heuristic method (NSGA-II)

Some parameters need to be set while running the algorithms. The parameters used in this method were:

1- Maximum repetition = 800
2- Population = 80
3- Percentage of crossover operator = 0.8
4- Percentage of mutation operator = 0.3

The problem was solved with the above parameters on various scales in MATLAB software. The results were as shown in Figures 7, 8, and 9.

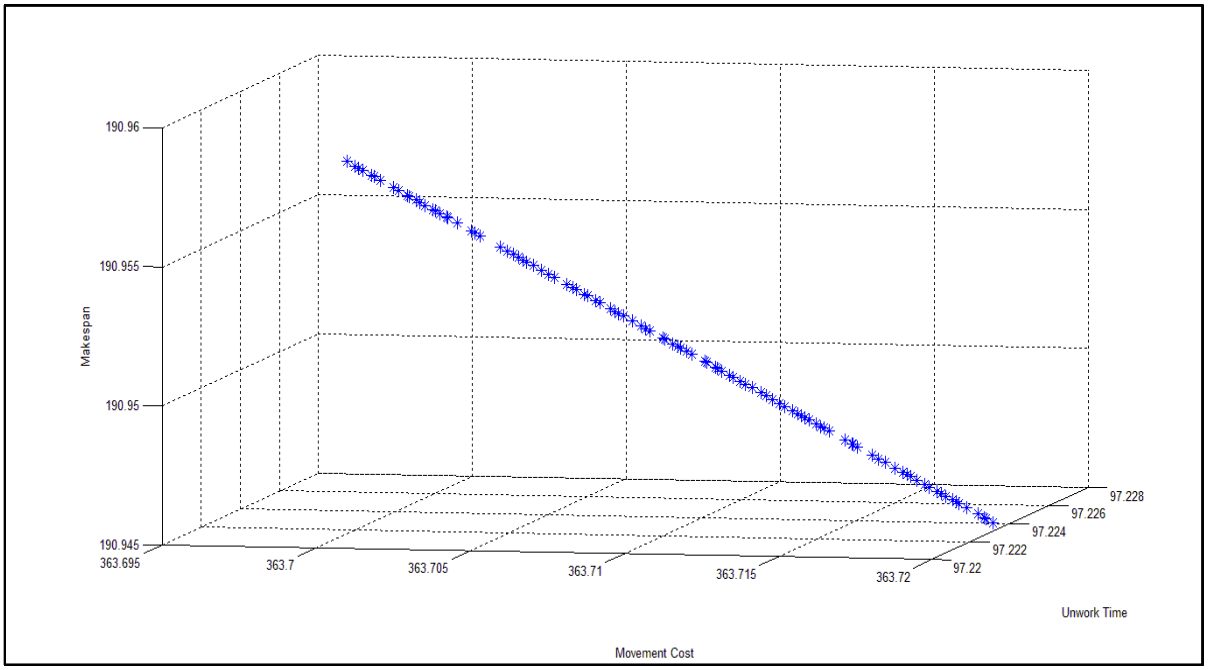

Figure 7: Small-scale meta-heuristic solution

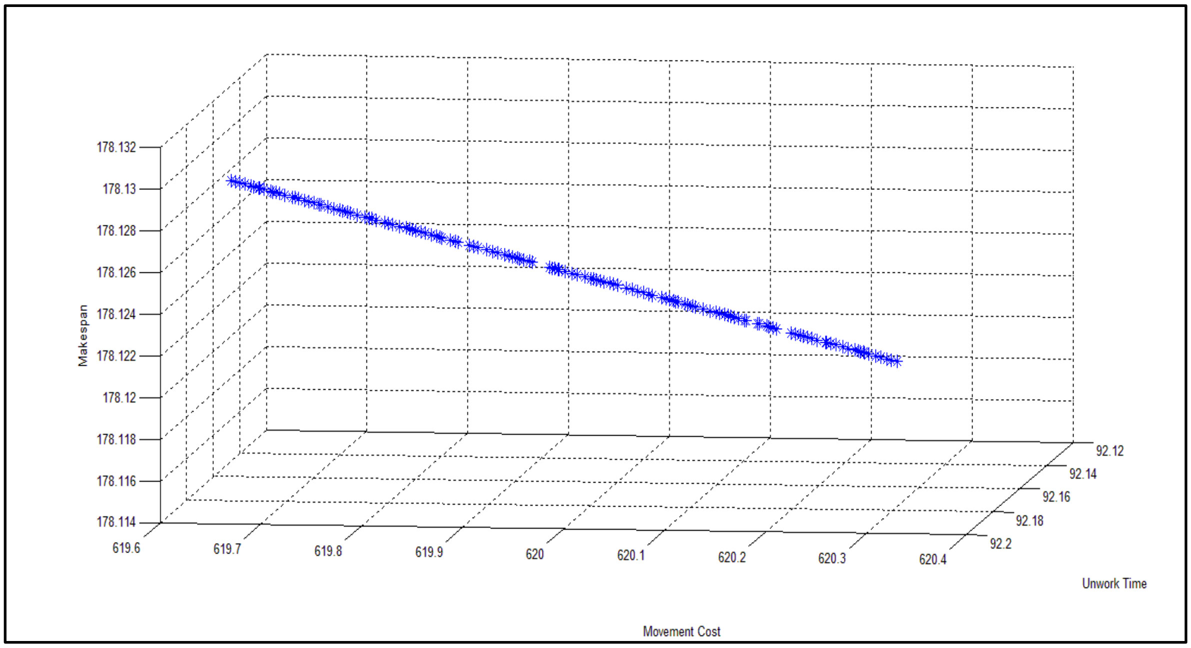

Figure 8: Medium-scale meta-heuristic solution

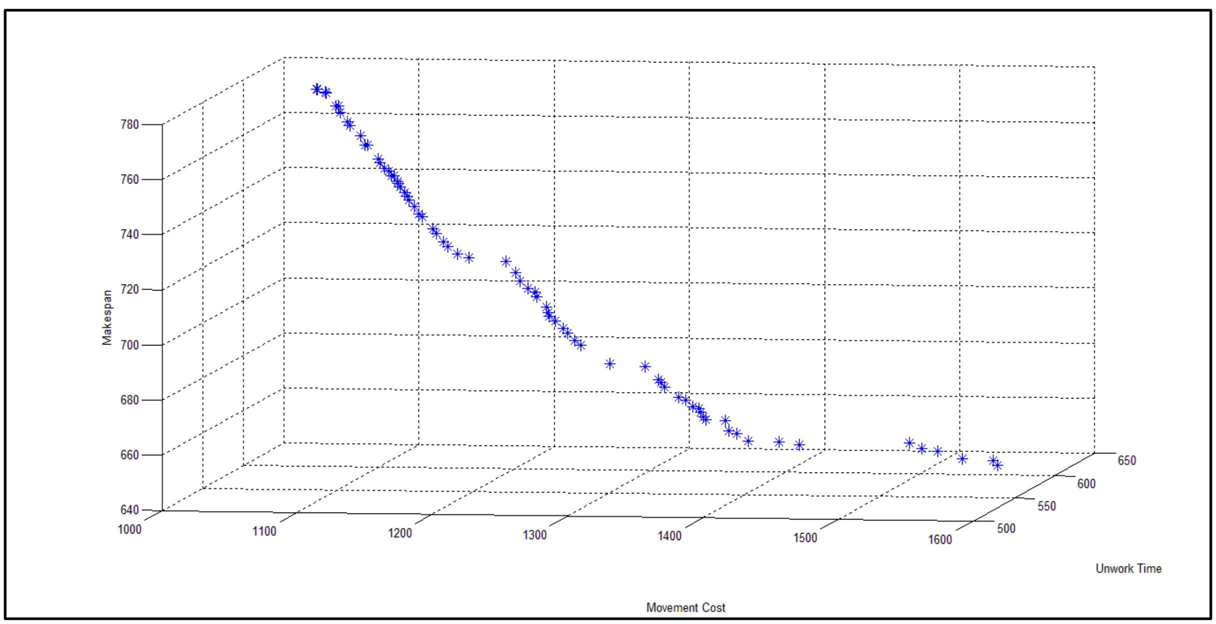

Figure 9: Large-scale meta-heuristic solution

Moreover, the performance of the proposed NSGA-II should be evaluated. For this reason, the small-scale problem is solved and compared with three other metaheuristic algorithms. These algorithms are Scatter search (SS), Tabu search (TS), and Simulated annealing (SA). Firstly, a brief description of these algorithm is presented and then the results are compared.

5-4 Scatter search (SS)

This algorithm was first presented by Glover (1977) as an extension of mathematical formulation for combinatorial optimization problems. Scatter search originated from earlier strategies for combining constraints and decision rules. Its objective is enabling a solution procedure based on the combined elements to achieve better solutions than one based only on the original elements (Glover, Laguna, & Martí, 2003). Its framework is flexible, allowing the development of implementations with varying degrees of sophistication (Martí, Corberán, & Peiró, 2016).

A detailed description of SS is presented in Algorithm 2. The main parameters are the P (main population), PR (reference set), ND (non-dominated set), n_e (evaluations number), initial temperature (T_i), cooling rate (T_{cr}), and temperature threshold (T_{stop}). The solution process starts with generating the initial population of *P*, and the external archive of non-dominated solutions (*ND*). These solutions are then sorted according to Pareto-dominance ranking, in a way that $P[1]$ is the solution dominating more solutions in the population, while $P[P_{size}]$ is the worst regarding this criterion. After initializing the population set, the *PR* should be created, which is comprised of the best $PR_{size}/2$ solutions of *P* in *PR* and $PR_{size}/2$ solutions of *PR* with the greatest Euclidean distance from other solutions in *PR*. Then, the non-dominated solutions (*ND*) should be sorted out from the solutions included in *PR*, obtaining the first Pareto frontier. Next, a certain number of solutions of *PR*, *Z*, are selected and passed to crossover function introduced in NSGAII algorithm to generate new solutions. Each combination gives a trial solution as a result. The domination of each new combination is checked with the solutions in *ND*. If the trial solution is not dominated by those previously included in *ND*, the none dominated solutions list is updated. The process is repeated until no improvement has been achieved for any new trail solution or the number of iterations/evaluations ($eval\_counter$) exceeds n_e.

Algorithm 2 Scatter search

Inputs: $P_{size}, PR_{size}, ND_{size}, T_i, T_{cr}, T_{stop}, n_e$;
$P \leftarrow \emptyset$; $PR \leftarrow \emptyset$; $ND \leftarrow \emptyset$; $eval\_counter \leftarrow 0$;
For ($P_{count} = 1\ to\ P_{size}$)
    $P[P_{count}] \leftarrow$ Generate a random solution;
    $ND \leftarrow$ Determine the non-dominated solutions;
Sort the solutions of *P* according the objective function;
For ($PR_{count} =$ 1to $PR_{size}/2$)
    $PR[PR_{count}] \leftarrow P[P_{count}]$;
Sort the solutions of *P* according their Euclidean distance to the solutions included in *PR*;
For (PR_{count}= ($PR_{size}/2$) + 1 to PR_{size})
    $PR[PR_{count}] \leftarrow P[PR_{count} - (PR_{size}/2)]$;
do
    new_solutions ←FALSE;
    NewSubSets ← Randomly chose the pairs of solutions ;

```
  do
    Select the next subset Z in NewSubSets;
    trial ← Crossover(Z);
    trial2 ← Mutation(trial);
    ND ← Determine the non-dominated solutions;
    If (trial2 dominates the solutions in ND) then
      Update ND according trial2;
      new_solutions ←TRUE;
    Delete Z from NewSubSets;
    For (PR_count = 1 to PR_size)
      If (trial2 ≺ PR[PR_count]) then
        PR[PR_count] ← trial2;
  while (NewSubSets ≠ Ø);
while ((new_solutions = TRUE) AND (eval_counter <n_e));
```

5-5 Tabu search (TS)

Tabu search (TS) is a metaheuristic local search method which is utilized for mathematical optimization. TS was developed by Glover (1989) as a motivation for mechanics of human memory (Malczewski, 2018). Tabu search is an effective method for overcoming local optimality trapping. It redirects the search to keep exploring the feasible region even after a local optimum is found, and it seeks to avoid returning to the same local optimum by employing a search memorizing mechanism. The algorithm selects the best objective value for creating the seed for the next generation (Banerjee, Singh, Sahana, & Nath, 2022).

TS has been used in many single objective optimization problems. Nevertheless, the multi objective variants of TS are simple, general and tractable which show a good performance at identifying a wide Pareto front for different problems. The multi-objective tabu search algorithm used for solving the problem is described below.

Step 0. Initialization

Set the tabu list and the *ND* solutions as empty. Generate the initial population randomly with *pop* number of solutions, and determine *ND* list and the current solution.

Step 1. Select the objective

Active a single objective to become primary using a multinomial probability mass function.

Step 2. Search the neighborhood

Based on the current solution generate a neighboring population of size pop. The non-tabu that has the best objective function value, or the tabu solution which dominates any solution in the *ND* should be chosen as the best candidate solution.

Step 3. Update the ND solutions list

Compare each candidate solution with the current *ND* solutions list and replace the old dominated solutions from *ND* with new dominating candidates. Add a candidate solution to the *ND* solutions list, if it is not dominated by any current *ND* solution.

Step 4. Update the tabu list

Update the tabu list according to the best candidate solution determined in *Step 2*.

Step 5. Diversification

Check the last moves in step 4, and if the ND list has not been updated, one of the ND solutions found should be randomly selected as the new solution. Afterwards, the tabu list should be reset and the search restarts.

5-6 Simulated annealing (SA)

Simulated annealing is a method for solving optimization problems, which models the physical process of heating a material and then slowly decreasing the temperature to minimize defects and the system energy. Particles may freely move at high temperatures, while by decreasing the temperature, they are more restricted due to the excessive required energy. Throughout the simulation according to the real practice of the annealing process, the primary goal is to minimize the *energy function*, $E(x)$, of the state x, introducing a controlling parameter T as the computational temperature. At each iteration of the simulated annealing algorithm a new sample point is randomly generated, considering the equilibrium distribution π_T (x) ∝ exp$\{E(x)/T\}$. As T become smaller the probability mass of π_T exclusively focuses on the neighboring region of the global minimum of E, so that any new point from π_T is more likely to place at the minimum of E.

Generally, moving from the current state x is examined by Metropolis-Hastings criteria, which considers an acceptance chance for the new state x' as:

$$A = \min(1, \exp\{-\frac{\delta E(x', x)}{T}\})$$

where

$$\delta E(x', x) = E(x') - E(x)$$

The perturbations from x to x′ which decrease the energy are always accepted. However, at high values of T the perturbations increasing the energy are likely to be accepted regarding the value of A. With the proceeding of the algorithm, the temperature systematically decreases, imposing the algorithm to reduce the extent of its search for converging to a minimum.

In single objective problems the energy function $E(x)$, is meant to measure of the quality of any solution x. However, the desired solutions in the multi-objective problems are only defined in relation to each other. In this paper a multi-objective SA algorithm is presented to deal the group layout problem. At each iteration of the algorithm, one epoch consisting of L_k solutions is drawn with a fixed temperature T_k. The computational temperature decreases by a proportion of α in the next epoch. Each new solution is a mutation of the current state and is accepted with probability given by (5), as shown in lines 4-8. The algorithm terminates once the temperature is decreased enough.

Algorithm 3 Simulated annealing

Inputs:

L_k *Sequence of epoch durations*

x *Initial solution*

```
for k := 1, …, pop
    for i := 1, …, L_k
        for x' := mutate(x)
        δE := E(x') − E(x)
        u := rand(0,1)
        if u < min(1, exp{− δE/T_k})
            x := x'
        end
    end
    T_{k+1} := T_k × α
end
```

During the optimization procedure, the true Pareto front is surely unavailable. Therefore, an estimate of the Pareto front is proposed using the set of mutually non-dominating solutions found thus far in the annealing. Let $\tilde{F}_x$ be the elements of $\tilde{F}$ that dominate x:

$$\tilde{F}_x = \{y \in \tilde{F} \mid y \prec x\}$$

so that an energy difference between x and x' would be:

$$\delta E(x', x) = \frac{1}{|\tilde{F}|}\left(|\tilde{F}_x| - |\tilde{F}_{x'}|\right)$$

Division by $|\tilde{F}|$ secures δE against fluctuations in the number of non-dominating solutions at each iteration. Thus, the energy difference between any two of non-dominating solutions is zero.

The problem is solved in all scales with implementing four algorithms in MATLAB and run on a PC with Intel Core i5 2.40 GHz CPU and 4 GB RAM. One of the Pareto optimal solutions and the solving time are shown in Table 10.

TABLE

Evaluating performance of the proposed NSGA-II

Since there is not a specific optimal solution in multi objective problems, three performance measures are used to compare the proposed NSGA II with three other algorithms over 10 times run. The performance measures are the average distance between the pareto optimal solutions, average number of Pareto solutions and the solving time. The first measure is obtained using the following equation:

$$\delta_j = \operatorname{argmin}_{i\in H}\left(\sqrt{\sum_k \left(\frac{f_i^k - f_j^{*k}}{f_j^{*k}}\right)^2}\right)$$

$$ED = \max_{k\in O, j\in E}\left(\left|\frac{f_{\delta_j}^k - f_j^{*k}}{f_j^{*1}}\right|\right)$$

where E is the solutions of optimal pareto reported by ϵ-constrint method, H is the solutions of best pareto, reported by a given algorithm, O is the set of all objective functions, f_j^{*k} is the k-th objective value of solution j in the optimal pareto front, f_i^k is the k-th objective value of solution i in the pareto front presented by the algorithm, δ_j is the index of the solution in H with the minimum Euclidean distance from solution $j \in E$, and ED is the external distance of the pareto front provided by the algorithm from the optimal pareto. It can be concluded that an algorithm with the lowest percentage of ED is the best option for solving the multi-objective problem.

TABLE

5-4 Computational results

While the exact solution of the mathematical model was done on a small and medium scale, the multi-objective genetic algorithm (NSGA-II) was used to validate and solve the model on a large scale because, in this case, the CM was an NP-hard problem. Comparison of the two solutions showed that the exact solution method performed better on small and medium scales. In addition, solving by the ε-constraint method increased the solution time. Table 10 shows a side-by-side comparison of the results. We should note that the Pareto-optimal solutions of the ε-constraint method in this table are based on the minimum solving time amongst other solutions in order to demonstrate that the meta-heuristic method took less time for solving than the best time of another method.

Table 10: Comparison of solution methods

Method	Exact solution (ε-constraint method)				Meta-heuristic method (NSGA-II)			
	Completion time	Transportation cost	Machine idle time	Solving time (Second)	Completion time	Transportation cost	Machine idle time	Solving time (Second)

Small-scale	188.970	387.700	84.900	3108	190.25	363.71	97.224	2748
Medium-scale	165.625	546.81	100.69	5872	178.12	620	92.16	3135
Large-scale (NP-hard problem)	-	-	-	-	710	1300	575	3852

To validate the model, the results from GAMS software were compared with the results from MATLAB software. Since the two were very close to one another, it can be concluded that the solution of the model is valid.

In this section the details of the data on small scale problem are reset to test the proposed approach for solving the model. As presented in the Table 11, there are nine different instances according to dimension of machines and floor plan, importance factor, intercellular transfer cost coefficient, intracellular transfer time coefficient, and operation time. In general, the results that are shown in the table 12 are one of the Pareto optimal solutions of the search. The results of the e-constraint method and NSGA-II method could be compared and it can be seen that they are close enough to each other, therefore, it can be concluded that our model is valid.

Instance	Parameters of each instance																									
no	Horizontal dimension of machine			Vertical dimension of machine			Horizontal dimension of the floor plan	Vertical dimension of the floor plan	Importance factor		Intercellular transfer cost coefficient				Intracellular transfer time coefficient				Operation time							
	Machine			Machine					Product		Part				Part				Operation							
	1	2	3	1	2	3			1	2	1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8
1	10	12	14	15	20	17	100	100	1.5	2.5	1.2	1.4	2	2.4	20	25	30	40	110	140	124	173	182	135	179	193
2	20	23	15	20	19	25	200	200	2.3	1.7	2.5	2.7	4	1.9	14	11	13	17	210	280	271	236	286	294	271	250
3	20	21	27	21	22	23	500	500	1.4	1.7	3.2	2.7	1.9	4.8	19	14	30	48	320	314	398	376	372	325	317	367
4	17	20	16	19	21	18	400	400	1.7	1.8	1.7	2.1	2.7	3.5	47	46	30	25	410	467	482	431	429	460	471	453
5	11	10	14	14	12	14	600	600	1.4	1.5	1.4	1.5	1.7	4.7	30	47	25	37	501	539	597	541	520	573	581	564
6	17	20	24	18	22	25	900	900	2.2	2.5	1.5	1.8	2.1	1.7	30	27	25	35	630	671	692	683	691	648	627	661
7	25	30	34	30	34	40	200	200	3.2	4	4.7	3.9	2.3	1.6	25	27	29	36	715	763	781	796	774	741	725	792
8	60	40	30	40	45	60	500	500	4.1	5	2.5	3.1	2.8	3.6	11	30	47	41	820	837	891	863	814	873	886	864
9	15	20	17	36	21	19	900	900	4.5	5.5	2.5	3.1	2.8	3.6	11	30	47	41	914	963	974	980	931	947	982	907

Table 11: The input data for test instances.

Table 12: Results of test instances

Solution	Exact solution (ε-constraint method)				Meta-heuristic method (NSGA-II)			
no	Completion time	Transportation cost	Machine idle time	Solving time (Second)	Completion time	Transportation cost	Machine idle time	Solving time (Second)
1	4403.000	5654.400	1252.000	3280	6364	5460	1717	2556.06
2	5086.100	5326.600	1357.000	3805	5242	5697	1584	2793.63

3	6176.700	12687.200	1954.000	3057	9468	14200	1798	2528.15
4	9458.500	10802.855	2502.241	4124	9636	10270	3414	3027.87
5	6193.900	6088.100	1339.000	3693	8561	8502	14430	2789.05
6	13901.794	10859.513	2732.477	2956	14590	8487	4041	2452.89
7	54130.400	16544.900	8231.000	2871	49660	14860	10630	2398.87
8	46281.400	25254.000	4803.000	3169	50720	23310	6692	2508.79
9	35228.500	9861.600	1890.000	3275	36870	8249	3121	2464.08

To evaluate solving time of the exact solution method, we could reduce some variables and constraints and see the positive or negative effect on solving time. As we mentioned in explaining the e-constraints method, first we solve the model with the first objective function. Afterward, in the second step, solve the model by the second objective function by moving the first objective to constraints, and in the third step, move the first and the second objective function to constraints and solve the model with the third objective function. Table 13 shows the time of these 3 steps by running the model on the small scale. It should be noted that percentages that mentioned in the last column show the effect (increase or decrease) of changes on solution time.

Table 13: Comparison of solution time

Method	Time of first step (Second)	Time of second step (Second)	Time of third step (Second)	Total (second)
Current model	569	2180	359	3108
Without product (i = 0)	-	2180	1704	3884 (+24%)
Removing Assigning machines to cells (z_lk = 0)	124	290	85	499 (-83%)
Removing capacity of each cell (constraints 5)	571	2180	377	3128 (+0.6 %)
Removing limitation for each cell (constraint 6)	192	195	229	616 (-80%)
Removing Operation on	141	49	33	223 (-92%)

machines (constraint 29)				
Excluding shape of cells (constraints 9-12)	23	54	28	105 (-96%)
Excluding floor plan dimensions (constraints 24-27)	582	1430	322	2334 (-25%)

We can see removing a variable or constraint can have a negative effect on solution time. It can be justified that the variable or constraint can make the model simpler and eliminating them make the solving process more complicated.

6- Sensitivity analysis

To perform sensitivity analysis on the important parameters, first the parameters were identified, and then the analysis was undertaken. It was also examined whether, in case of any change of an input parameter, the model showed the expected performance or not. Sensitivity analysis is performed on several Pareto points and the percentages obtained are the average percentage change.

6-1 Horizontal dimension parameter of the floor plan

- Increasing this parameter did not change the value of the objective function.
- Reducing this parameter from 8 to 4.3 did not change the objective function but falling below 4.3 rendered the problem unjustified.

6-2 Parameter of vertical dimension of machines

The average percentage of the objective functions varied when the vertical dimension of machines was changed, as shown in Table 14.

Table 14: Average change of objective function from changing the vertical dimension of machines

Change in the vertical dimension of machines	Change in completion time	Change in transportation cost	Change in idle time of the machine
20%	5.2%	12.9%	Unjustified
50%	9.7%	24.1%	Unjustified

100%	10.3%	26%	Unjustified
-20%	-5.2%	-12.9%	-26.6%
-50%	-9.7%	-24.1%	-48.3%
-100%	-10.3%	-26%	-54%

6-3 The maximum number of allowed machines in the cell

- Increasing this parameter did not change the value of the objective function.
- Reducing this parameter from 3 to 2 did not change the objective function but falling below 2 rendered the problem unjustified.

6-4 Product importance factor

The average percentage of the objective functions varied when the product importance factor was changed, as shown in Table 15.

Table 15: Average change of objective functions from changing the importance factor of the products

Change in the product importance factor	Change in completion time	Change in transportation cost	Change in idle time of the machine
20%	20%	0	0
50%	50%	0	0
100%	100%	0	0
-20%	-20%	0	0
-50%	-50%	0	0
-100%	-100%	0	0

6-5 Intercellular transfer cost coefficient

The average percentage of the objective functions varied when the intercellular transfer cost coefficient was changed, as shown in Table 16.

Table 16: Average change of objective functions from changing intercellular transfer cost coefficient

Change in the intercellular transfer cost coefficient	Change in completion time	Change in transportation cost	Change in idle time of the machine
20%	0	15.3%	0
50%	0	39.4%	0
100%	0	88.9%	0
-20%	0	-17.7%	0
-50%	0	-43.8	0
-100%	0	-92.4	0

The results derived from the sensitivity analysis revealed that, by changing different parameters, the model shows the expected behavior. Therefore, it can be concluded that the model works appropriately and accurately.

7- Conclusion

Due to the importance and interdependence of CF, GS, and GL problems in a CMS, a mathematical model has been presented to simultaneously approach these three decisions. The objective is to minimize total completion time, transportation cost, and machine idle time between operations. We have extended Arkat et al. (2012a) by adding machinery to the objective function and considering a multi-machine system that makes our model more practical.

Two methods have been developed to solve the model: the exact solution by the ε-constraint method using GAMS software and a non-dominated sorting genetic algorithm II (NSGA-II) with MATLAB software for the metaheuristic solution. To examine the effectiveness of the integrated model, we have compared the results of the two methods. Although the comparison has indicated that the proposed exact solution method is superior to the NSGA-II, the metaheuristic algorithm can provide a near-optimal solution in a much shorter computational time and solve large-scale problems. In addition, by utilizing sensitivity analysis, the impacts of various model parameters on the system's behavior are investigated. This helps the manufacturer to detect the more sensitive parameters.

The proposed model may give rise to several potential research improvements:

- In this study, moving parts are considered individually, but in the real world, small parts can usually be done in batches, which reduces transportation costs in this case.

- We consider that cells consist of different machines performing only production operations. In many companies, the manufactured parts are assembled inside the company and become the final product after completing the manufacturing process. In future studies, cells could also be considered for inspection and maintenance, which can also affect the movements and performance of the system.
- In this research, machines are conceived to operate one piece at a time. However, with technological advances, companies and production units are increasingly having machines perform operations on several parts simultaneously. Considering the capacity of such machines may be worthy of exploration in the future.
- The operation time on the parts in the model is considered definite. However, parts operation times on machines are usually uncertain. This may also provide for a fresh investigation.

Uncategorized References

Akhmet, A., Hare, W., & Lucet, Y. (2022). Bi-objective optimization for road vertical alignment design. *Computers & Operations Research, 143*, 105764. doi:https://doi.org/10.1016/j.cor.2022.105764

Alimian, M., Ghezavati, V., & Tavakkoli-Moghaddam, R. (2020). New integration of preventive maintenance and production planning with cell formation and group scheduling for dynamic cellular manufacturing systems. *Journal of Manufacturing Systems, 56*, 341-358.

Arkat, J., Farahani, M. H., & Ahmadizar, F. (2012). Multi-objective genetic algorithm for cell formation problem considering cellular layout and operations scheduling. *International Journal of Computer Integrated Manufacturing, 25*(7), 625-635.

Arkat, J., Farahani, M. H., & Hosseini, L. (2012). Integrating cell formation with cellular layout and operations scheduling. *The International Journal of Advanced Manufacturing Technology, 61*(5-8), 637-647.

Arkat, J., Hosseinabadi Farahani, M., & Hosseini, L. (2012). Integrating cell formation with cellular layout and operations scheduling. *The International Journal of Advanced Manufacturing Technology, 61*(5), 637-647. doi:10.1007/s00170-011-3733-4

Arkat, J., Rahimi, V., & Farughi, H. (2021). Reactive Scheduling Addressing Unexpected Disturbance in Cellular Manufacturing Systems. *International Journal of Engineering, 34*(1), 162-170.

Arora, P., Haleem, A., Kumar, H., & Khan, S. A. (2020). Recent Development in Virtual Cellular Manufacturing System. In *Recent Advances in Mechanical Engineering* (pp. 1-7): Springer.

Aryanezhad, M., Aliabadi, J., & Tavakkoli-Moghaddam, R. (2011). A new approach for cell formation and scheduling with assembly operations and product structure. *International Journal of Industrial Engineering Computations, 2*(3), 533-546.

Banerjee, A., Singh, D., Sahana, S., & Nath, I. (2022). Chapter 3 - Impacts of metaheuristic and swarm intelligence approach in optimization. In S. Mishra, H. K. Tripathy, P. K. Mallick, A. K. Sangaiah, & G.-S. Chae (Eds.), *Cognitive Big Data Intelligence with a Metaheuristic Approach* (pp. 71-99): Academic Press.

Bayram, H., & Şahin, R. (2016). A comprehensive mathematical model for dynamic cellular manufacturing system design and Linear Programming embedded hybrid solution techniques. *Computers & Industrial Engineering, 91*, 10-29.

Bérubé, J.-F., Gendreau, M., & Potvin, J.-Y. (2009). An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits. *European journal of operational research, 194*(1), 39-50.

Brown, E. C., & Sumichrast, R. T. (2001). CF-GGA: a grouping genetic algorithm for the cell formation problem. *International Journal of Production Research, 39*(16), 3651-3669.

Cao, D., & Chen, M. (2004). Using penalty function and Tabu search to solve cell formation problems with fixed cell cost. *Computers & Operations Research, 31*(1), 21-37.
Chandrasekharan, M., & Rajagopalan, R. (1993). A multidimensional scaling algorithm for group layout in cellular manufacturing. *International journal of production economics, 32*(1), 65-76.
Chang, C.-C., Wu, T.-H., & Wu, C.-W. (2013). An efficient approach to determine cell formation, cell layout and intracellular machine sequence in cellular manufacturing systems. *Computers & Industrial Engineering, 66*(2), 438-450.
Cheng, T. E., Janiak, A., & Kovalyov, M. Y. J. S. J. o. O. (1998). Bicriterion single machine scheduling with resource dependent processing times. *8*(2), 617-630.
De Lit, P., Falkenauer, E., & Delchambre, A. (2000). Grouping genetic algorithms: an efficient method to solve the cell formation problem. *Mathematics and Computers in simulation, 51*(3-4), 257-271.
Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation, 6*(2), 182-197.
Ebrahimi, A., Kia, R., & Komijan, A. R. (2016). Solving a mathematical model integrating unequal-area facilities layout and part scheduling in a cellular manufacturing system by a genetic algorithm. *SpringerPlus, 5*(1), 1-29.
Ehrgott, M., & Gandibleux, X. (2003). Multiobjective combinatorial optimization—theory, methodology, and applications. In *Multiple criteria optimization: State of the art annotated bibliographic surveys* (pp. 369-444): Springer.
El Yaagoubi, A., Charhbili, M., Boukachour, J., & El Hilali Alaoui, A. (2022). Multi-objective optimization of the 3D container stowage planning problem in a barge convoy system. *Computers & Operations Research*, 105796. doi:https://doi.org/10.1016/j.cor.2022.105796
Elmi, A., Solimanpur, M., Topaloglu, S., & Elmi, A. (2011). A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. *Computers & Industrial Engineering, 61*(1), 171-178.
Forghani, K., & Fatemi Ghomi, S. M. T. (2020). Joint cell formation, cell scheduling, and group layout problem in virtual and classical cellular manufacturing systems. *Applied Soft Computing, 97*, 106719. doi:https://doi.org/10.1016/j.asoc.2020.106719
Ghezavati, V., & Saidi-Mehrabad, M. (2010). Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time. *The International Journal of Advanced Manufacturing Technology, 48*(5-8), 701-717.
Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences, 8*(1), 156-166.
Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing, 1*(3), 190-206.
Glover, F., Laguna, M., & Martí, R. (2003). Scatter search. In *Advances in evolutionary computing* (pp. 519-537): Springer.
Goli, A., Tirkolaee, E. B., & Aydin, N. S. (2021). Fuzzy Integrated Cell Formation and Production Scheduling considering Automated Guided Vehicles and Human Factors. *IEEE Transactions on Fuzzy Systems*, 1-1. doi:10.1109/TFUZZ.2021.3053838
Goli, A., Tirkolaee, E. B., & Aydın, N. S. J. I. T. o. F. S. (2021). Fuzzy integrated cell formation and production scheduling considering automated guided vehicles and human factors. *29*(12), 3686-3695.
Grimme, C., Kerschke, P., Aspar, P., Trautmann, H., Preuss, M., Deutz, A. H., . . . Emmerich, M. (2021). Peeking beyond peaks: Challenges and research potentials of continuous multimodal multi-objective optimization. *Computers & Operations Research, 136*, 105489. doi:https://doi.org/10.1016/j.cor.2021.105489
Guschinsky, N., Kovalyov, M. Y., Rozin, B., Brauner, N. J. C., & Research, O. (2021). Fleet and charging infrastructure decisions for fast-charging city electric bus service. *135*, 105449.

Hassan, M. M. (1995). Layout design in group technology manufacturing. *International journal of production economics, 38*(2-3), 173-188.

Khamlichi, H., Oufaska, K., Zouadi, T., & Dkiouak, R. (2020). A Hybrid GRASP Algorithm for an Integrated Production Planning and a Group Layout Design in a Dynamic Cellular Manufacturing System. *IEEE Access, 8*, 162809-162818.

Kirkpatric, S. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671-680.

Liu, C., Wang, J., Leung, J. Y.-T., & Li, K. (2016). Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm. *International Journal of Production Research, 54*(3), 923-944.

Mahdavi, I., Paydar, M., Solimanpur, M., & Saidi-Mehrabad, M. (2010). A mathematical model for integrating cell formation problem with machine layout. *International Journal of Industrial Engineering & Production Research, 21*(2), 61-70.

Mahesh, K., Nallagownden, P., & Elamvazuthi, I. J. E. (2016). Advanced Pareto front non-dominated sorting multi-objective particle swarm optimization for optimal placement and sizing of distributed generation. *9*(12), 982.

Malczewski, J. (2018). 1.15 - Multicriteria Analysis. In B. Huang (Ed.), *Comprehensive Geographic Information Systems* (pp. 197-217). Oxford: Elsevier.

Martí, R., Corberán, Á., & Peiró, J. (2016). Scatter Search. In R. Martí, P. Panos, & M. G. C. Resende (Eds.), *Handbook of Heuristics* (pp. 1-24). Cham: Springer International Publishing.

Miettinen, K. (2012). *Nonlinear multiobjective optimization* (Vol. 12): Springer Science & Business Media.

Miettinen, K. J. B., Massachusetts. (1999). Nonlinear multiobjective optimization kluwer academic publishers.

Mohtashami, A., Alinezhad, A., & Niknamfar, A. H. (2020). A fuzzy multi-objective model for a cellular manufacturing system with layout designing in a dynamic condition. *International Journal of Industrial and Systems Engineering, 34*(4), 514-543.

Neufeld, J. S., Teucher, F., & Buscher, U. (2020). Scheduling flowline manufacturing cells with inter-cellular moves: non-permutation schedules and material flows in the cell scheduling problem. *International Journal of Production Research, 58*(21), 6568-6584.

Neufeld, J. S., Teucher, F. F., & Buscher, U. (2020). Scheduling flowline manufacturing cells with inter-cellular moves: non-permutation schedules and material flows in the cell scheduling problem. *International Journal of Production Research, 58*(21), 6568-6584. doi:10.1080/00207543.2019.1683251

Paydar, M. M., & Saidi-Mehrabad, M. (2013). A hybrid genetic-variable neighborhood search algorithm for the cell formation problem based on grouping efficacy. *Computers & Operations Research, 40*(4), 980-990.

Rahimi, V., Arkat, J., & Farughi, H. (2020). A vibration damping optimization algorithm for the integrated problem of cell formation, cellular scheduling, and intercellular layout. *Computers & Industrial Engineering, 143*, 106439.

Rajagopalan, R., & Batra, J. (1975). Design of cellular production systems a graph-theoretic approach. *The International Journal of Production Research, 13*(6), 567-579.

Shafiee-Gol, S., Kia, R., Tavakkoli-Moghaddam, R., Kazemi, M., & Kamran, M. A. (2021). Integration of parts scheduling, MRP, production planning and generalized fixed-charge transportation planning in the design of a dynamic cellular manufacturing system. *RAIRO: Recherche Opérationnelle, 55*, 1875.

Solimanpur, M., & Elmi, A. (2013). A tabu search approach for cell scheduling problem with makespan criterion. *International journal of production economics, 141*(2), 639-645.

Tavakkoli-Moghaddam, R., Gholipour-Kanani, Y., & Cheraghalizadeh, R. (2008). A genetic algorithm and memetic algorithm to sequencing and scheduling of cellular manufacturing systems. *International Journal of Management Science and Engineering Management, 3*(2), 119-130.

Tavakkoli-Moghaddam, R., Javadian, N., Khorrami, A., & Gholipour-Kanani, Y. (2010). Design of a scatter search method for a novel multi-criteria group scheduling problem in a cellular manufacturing system. *Expert Systems with Applications, 37*(3), 2661-2669.

Tsai, C.-C., Chu, C.-H., & Barta, T. A. (1997). Modeling and analysis of a manufacturing cell formation problem with fuzzy mixed-integer programming. *IIE transactions, 29*(7), 533-547.

Uddin, M. K., & Shanker, K. (2002). Grouping of parts and machines in presence of alternative process routes by genetic algorithm. *International journal of production economics, 76*(3), 219-228.

Vin, E., De Lit, P., & Delchambre, A. (2005). A multiple-objective grouping genetic algorithm for the cell formation problem with alternative routings. *Journal of Intelligent Manufacturing, 16*(2), 189-205.

Vira, C., Haimes, Y. Y. J. N.-H. S. i. S. S., & Engineering. (1983). Multiobjective decision making: theory and methodology. 62-109.

Wang, J. (2003). Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm. *Automatica, 39*(9), 1607-1615.

Wang, J., Liu, C., & Zhou, M. (2021). *Multi-factory Cellular Manufacturing Cell Formation and Product Scheduling via Genetic Algorithm.* Paper presented at the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE).

Wei, J. C., & Gaither, N. (1990). An optimal model for cell formation decisions. *Decision Sciences, 21*(2), 416-433.

Wu, X., Chu, C.-H., Wang, Y., & Yue, D. (2007). Genetic algorithms for integrating cell formation with machine layout and scheduling. *Computers & Industrial Engineering, 53*(2), 277-289.

Wu, X., Chu, C.-H., Wang, Y., Yue, D. J. C., & Engineering, I. (2007). Genetic algorithms for integrating cell formation with machine layout and scheduling. *53*(2), 277-289.

Zuberek, W. (1996). *Optimal schedules of manufacturing cells-modeling and analysis using timed Petri nets.* Paper presented at the Proceedings of IEEE International Symposium on Industrial Electronics.