Time-dependent Hyperstar algorithm for robust vehicle navigation in time-dependent stochastic road networks

Michael G H Bell[1], Valentina Trozzi[1], Solmaz Haji Hosseinloo[1], Guido Gentile[2] and Achille Fonzone[1]

Abstract

The vehicle navigation problem studied in Bell (2009) is revisited and a time-dependent reverse Hyperstar algorithm is presented. This minimises the expected time of arrival at the destination, and all intermediate nodes, where expectation is based on a pessimistic (or risk-averse) view of unknown link delays. This may also be regarded as a hyperpath version of the Chabini and Lan (2002) algorithm, which itself is a time-dependent A* algorithm. Links are assigned undelayed travel times and maximum delays, both of which are potentially functions of the time of arrival at the respective link. The driver seeks probabilities for link use that minimise his/her maximum exposure to delay on the approach to each node, leading to the determination of the pessimistic expected time of arrival. Since the context considered is vehicle navigation where the driver is not making repeated trips, the probability of link use may be interpreted as a measure of link attractiveness, so a link with a zero probability of use is unattractive while a link with a probability of use equal to one will have no attractive alternatives. A solution algorithm is presented and proven to solve the problem provided the node potentials are feasible and a FIFO condition applies for undelayed link travel times. The paper concludes with a numerical example.

Key words: Robust route guidance; Vehicle navigation; Uncertain networks

Introduction

Vehicle navigation systems have made great advances in recent years. Not only has the quality of the electronic maps incorporated in them become more accurate, but also their user friendliness has improved. Current systems allow users to vary the criteria used for building routes (for example, fastest or shortest) or to avoid certain links, route sections or areas (like the congestion charging zone in London). Recent systems also allow the inclusion of on-line traffic information, broadcast with the TMC or more recently the TPEG protocol, enabling drivers to be routed away from congestion once this has been detected.

In addition to such improvements, there has been a move to make navigation available via GPS-enabled cell phones, such as the iPhone from Apple. The rapid spread of such devices has created the possibility of "crowdsourcing" maps and traffic data by tracking the users of such cell phones. Although "crowdsourcing" is in its infancy, the trend is clearly toward the more widespread use of navigation systems by cell phone, both in and out of vehicles. In addition, access to the internet facilitates the provision of a wide range of location-based services to accompany navigation, ranging from locating friends to making reservations.

[1] Centre for Transport Studies, Imperial College London
[2] Dipartimento di Idraulica Trasporti e Strade, Sapienza Università di Roma

Experience suggests that on-line congestion information alone may lead to unsatisfactory results as not all roads are monitored, the knock-on effects of mass rerouting is not taken into account, and there is often a lag between the onset of congestion and the broadcast of a warning message. As a consequence, systems have been developed to predict congestion (see, for example, the Inrix system). An alternative approach is to seek routes that avoid links with a history of congestion on the grounds that they are likely to prove unreliable in terms of travel time.

Market research has revealed that expected arrival time reliability is a major concern for users of navigation systems (see Park 2009). Recent work designed to generate more reliable routes is reviewed in Bell (2009). In general, most approaches so far proposed involve measuring link travel time reliability and then finding paths which tend to avoid links that are unreliable (see for example, Kaparias 2008, Chen et al. 2007 or Park et al 2007). This may be achieved by penalising unreliable links or by the use of a *k*-shortest path method. It should be noted that a path that excludes unreliable links may still be unreliable in total, particularly where the resulting path is tortuous.

More recently, Bell (2009) has noted that a reinterpretation of the Spiess and Florian (1989) algorithm enables it to be used to generate the set of paths, referred to as the *hyperpath*, that minimises expected travel time if at each node offering a choice of route drivers assign themselves probabilistically to the exit links so as to minimise their exposure to maximum delay on these exit links. This may be regarded as a *robust* approach to path finding, as it requires that each link be given a maximum delay but makes no assumption about the distribution of actual link delay, other than it lie between zero and the given maximum.

An important feature of road networks is that link travel times are subject to both regular time-of-day variation, for example peak and off-peak periods, and irregular incidents. There is therefore a strong case, particularly in urban areas, for time-dependent path finding. Because of the large number of links in road networks, efficient path finding algorithms are required. For static networks, the Dijkstra (1959) algorithm, which can be regarded as the mother of all efficient shortest path algorithms, may be sped up by the addition of node potentials (see Wagner and Willhalm, 2006) leading to the well known A* algorithm first proposed by Hart et al (1968). Although all vehicle navigation systems appear to use some form of the A* algorithm, the efficiency in practice depends on the choice of node potentials, and this remains a closely guarded secret. Bell (2009) showed that node potentials may also be applied to the reinterpreted Spiess and Florian algorithm speeding up the generation of hyperpaths. The resulting algorithm is referred to there and here as the Hyperstar algorithm.

Chabini and Lan (2002) have shown that the A* algorithm may be applied to time-dependent networks in a straight forward way provided links conform to a FIFO condition. FIFO implies that it should not be possible for link travel time as experienced by drivers to fall faster than real time, otherwise the sequence of drivers entering a link may differ from the sequence leaving it, implying overtaking or queue jumping within the link. However, a problem in applying the Hyperstar algorithm to time-dependent networks is that it builds the hyperpath from the destination forward to the origin. It is conventionally assumed that link travel time is a function of the time of arrival at a link and not the time of departure from a link. Moreover, Hyperstar provides a pessimistic estimate of the time of departure from the origin given the time of arrival at the destination, whereas a navigation system would normally require a pessimistic estimate of the time of arrival at the destination given the time of departure from the origin.

This paper explores the implications and consequences of reversing the Hyperstar algorithm to generate a hyperpath from the origin to the destination in the context of time-dependent networks. It shows that this reversal implies that drivers minimise their exposure to arrival delays at the destination and all intermediate nodes in the hyperpath, arguably a more rational strategy than basing route choices on departure delays from the origin and all intermediate nodes. It is proved that the reversed Hyperstar algorithm solves the underlying time-dependent problem, provided each link conforms to FIFO operation and all node potentials are feasible. The reversed Hyperstar algorithm is therefore the hyperpath counterpart to the Chabini and Lan (2002) time-dependent A* algorithm. An 8 time 8 node grid network with randomly generated link lengths and maximum link delays is used to illustrate some of the properties of the reversed Hyperstar algorithm.

Hyperpaths

A technical as well as a historical introduction to hypergraphs and hyperpaths may be found in Gallo et al. (1993), cited in Nielsen et al. (2005). A directed hypergraph consists of a set of nodes (vertices) and a set of hyperlinks (hyperarcs). A hyperlink consists of a set of head nodes and a set of tail nodes, either of which (but not both) may be empty. In contrast to a normal link, a hyperlink may have multiple head and tail nodes. Special hypergraphs arise when either the set of head nodes or the set of tail nodes is singular. A forward hypergraph, or F-graph, arises when the hyperlinks have singular head nodes.

Hyperpaths in hypergraphs are an extension to paths in graphs that allow graphs to represent more complex situations (Nielsen et al. 2005). A hypernode can be used to represent a transport interchange and hyperlinks the connections between interchanges, for example. Hall (1986) formulated the minimum expected travel time path problem for random time-dependent networks. He noted that finding the best route requires a strategy that assigns optimal successors to a node as a function of time. Pretolani (2000) shows that directed hypergraphs may be used to model discrete random time-dependent networks and that the minimum expected travel time problem is equivalent to solving the shortest hyperpath problem. Moreover, optimal strategies under different objectives, such as minmax travel time, can also be found by choosing appropriate weights. The shortest hyperpath approach has been proposed for routing problems in discrete random time-dependent networks, where the travel time through a link is a random variable whose distribution depends on the departure time (Miller-Hooks 2001). Problems of this type arise with hazardous materials transportation (Miller-Hooks and Mahmassani 1998) or packet routing in congested communication networks where it is important to allow for real-time routing decisions in response to on-trip events (Miller-Hooks 2001). The bicriterion shortest hyperpath problem has also been investigated. One such application relates to hazmat transportation where both expected travel time and expected risk are minimised.

F-graphs have been studied in the context of urban transit problems (see Nguyen and Pallottino 1988). Transit networks can be represented by a set of nodes corresponding to the stops and links corresponding to in-vehicle line segments, which in turn represent feasible combinations of boarding and alighting stops. A hyperpath represents an *attractive* set of in-vehicle line segments connecting an origin to a destination. The set of attractive in-vehicle line segments leaving a stop can be grouped to form a hyperlink, with one head node (the boarding stop) and potentially multiple tail nodes (the alighting stops, where the passenger interchanges if the stop is not the destination).

Behind the concept of attractiveness in transit networks are two assumptions. The first is the rule that passengers choose whichever attractive line arrives at the stop first. As a consequence, the attractive in-vehicle line segments are used in proportion to their service frequencies. This means that for each hyperlink the tail node is chosen by each passenger according to the “whichever line arrives first” rule and is therefore used in proportion to the corresponding service frequency. The second is that expected path travel time is minimised. An in-vehicle line segment is therefore only attractive if by excluding it from the choice set (and thus from the hyperlink) the expected travel time would be increased. Spiess and Florian (1989) have shown that, subject to these two assumptions, hyperpaths may be found by solving a linear programming problem.

Hypergraphs and hyperpaths have found applications in many fields, identified in Nielsen et al. (2005).

Notation

$c_a(t)$	Undelayed travel time on link *a* for trips arriving at link *a* at time *t* (h)
l_a	Length of link *a* (km)
$v_a(i)$	Speed on link *a* in interval *i* (kph)
$d_a(t)$	Maximum delay for link *a* for trips arriving at link *a* at time *t* (h)
p_a	Probability of using link *a*, interpreted as link attractiveness (primal variable)
q_a	Pessimistic expectation of maximum delay on link *a* (dual variable)
b_i	1 if node *i* is a destination, -1 if node *i* is an origin, and 0 otherwise
w_i	Pessimistic expectation of delay in arriving at node *i*, based on the probability of using the links entering that node (primal variable) (h)
u_i	Pessimistically expected arrival time at node *i* (dual variable)
f_i	Sum of inverse maximum link delays for attractive links entering node *i* (see Step 2 of Algorithm A_0)
y_i	Probability of using node *i*
h_i	Potential for node *i*, used to speed up the search for the destination
A	Set of links
H	Set of links in the hyperpath from origin *r* to destination *s*
I	Set of nodes
A_i^+	Set of links entering node *i*
A_i^-	Set of links exiting node *i*

Problem formulation

The problem considered here closely resembles the Spiess and Florian (1989) problem, as reinterpreted in Bell (2009), only exposure to maximum delay is minimised on entry to a node rather than on exit from it. The interpretation is that links are subject to an unknown delay up to a predefined maximum. Where there is more than one potentially optimal way (link) to enter node *i*, we assume that the probability of using each is calculated to minimise the maximum exposure to delay. A potentially optimal link is referred to as an *attractive* link.

This is a *myopic* consideration of delay in much the same way that the reinterpreted Spiess and Florian method considers delay myopically. Exposure to delay is minimised on entry to a node rather than over the trip as a whole (the problem of minimising exposure to delay over the trip as a whole in a time-independent context is looked at in Schmoecker et al, 2009). It is conjectured here that the myopic minimisation of exposure to delay coupled with a focus on arrival time is a more natural way of managing uncertainty in the context route guidance, where the traveller is typically concerned about the expected time of arrival. The traveller's thinking may be expressed as follows: "There may be a delay in arriving at node *i* so I will consider the alternatives, and then assign them probabilities of use so as to minimise my maximum exposure to delay".

Probability of use here may be regarded as a measure of attractiveness. A link or node with a zero probability of use has no possibility of being optimal and is therefore unattractive. Conversely, a link or node with a probability of use equal to one is optimal and therefore has no attractive alternatives. Faced with making a choice, the user of the navigation system, one can presume, would tend to choose the more attractive options but in the knowledge that another option may ex post have been better.

When the expected time of arrival is based on probabilities of use that minimise the maximum exposure to delay myopically, the above principle can be applied recursively. If the expected time of arrival at node *j* plus the undelayed travel time from node *j* to node *i* offers the possibility of arriving earlier at node *i* than the expected time of arrival at node *i*, then the link from *j* to *i* is attractive. The underlying problem (P_0) has the form of the Spiess and Florian (1989) linear program with two differences; link travel times are a function of the pessimistically expected arrival time at the link (the objective function) and link use is calculated to minimise maximum expected delay that would be encountered at the pessimistically expected time of arrival at the link in question:

P_0: $\min_{p,w}\left(\sum_{a=(i,j)\in A} p_a c_a(u_i) + \sum_{j\in I} w_j\right)$

subject to

(1) $\sum_{a=(i,j)\in A_j^+} p_a - \sum_{a=(j,i)\in A_j^-} p_a = b_j$ for all $j \in I$

(2) $w_j \geq p_a d_a(u_i)$ for all $a = (i,j) \in A_j^+, j \in I$

(3) $p_a \geq 0$ for all $a \in A$

This formulation of the problem leaves the pessimistically expected time of arrival at each node *i* undetermined. If $p_a > 0$ then $a = (i, j) \in H$ and

(4) $u_i + c_a(u_i) \leq u_j$

since link *a* would not be attractive (and therefore not in the hyperpath) if it did not offer the possibility of reaching node *j* earlier.

Consideration of the following Lagrangean equation makes it clear that *u* is a dual variable for P_0:

(5) $L_{p,w,u,q} = \sum_{a=(i,j)\in A} p_a c_a(u_i) + \sum_{j\in I} w_j - \sum_{i\in I} u_i (b_i - \sum_{a=(i,j)\in A_j^+} p_a + \sum_{a=(j,i)\in A_j^-} p_a) - \sum_{a=(i,j)\in A} q_a (w_j - p_a d_a(u_i))$

P_0 is solved when $L_{p,w,u,q}$ is minimised with respect to *p* and *w* while being maximised with respect to *u* and *q*. Equation (5) can be rearranged as follows:

(6) $L_{p,w,u,q} = u_r - u_s + \sum_{a=(i,j)\in A} p_a (c_a(u_i) + q_a d_a(u_i) - u_j + u_i) + \sum_{j\in I} w_j \, (1 - \sum_{a=(i,j)\in A_j^+} q_a)$

This leads to the following dual problem

P_1: $\max_{u,q} (u_r - u_s)$

subject to

(7) $c_a(u_i) + q_a d_a(u_i) - u_j + u_i \geq 0$ for all $a = (i, j) \in A$
(8) $1 - \sum_{a=(i,j)\in A_j^+} q_a = 0$ for all $j \in I - \{r\}$
(9) $q_a \geq 0$ for all $a \in A$

Without loss of generality, u_r can be set to 0, so according to duality theory at the solution $u_s = \sum_{a=(i,j)\in A} p_a c_a(u_i) + \sum_{j\in I} w_j$, as the objective of the primal problem is equal to minus the objective of the dual problem. Hence at the solution u_s is the pessimistically expected cost of reaching the destination. Equation (7) is equivalent to (4) since

(10) $q_a d_a(u_i) \geq 0$ for all $a = (i, j) \in A$

Goal oriented search

It is well known that by transforming link costs through the subtraction of a node potential for the entry node and the addition of a node potential for the exit node for the cost of each link it is possible to speed up Dijkstra's algorithm when searching for the shortest path from one origin to one destination. The transformation is shown below.

(11) $c'_a(u_i) = c_a(u_i) - h_i + h_j$ for all $a = (i, j) \in A$

The resulting algorithm is known as the A* algorithm (Hart et al, 1968). Without loss of generality, we set h_s equal to 0, where node s is the destination. It has been shown that in order for the node potential to be *feasible*, it must be less than or equal to the remaining distance (or in this context travel time) to the destination. If the node potential is not feasible, the shortest path may not be found. The closer the node potential is to the remaining distance, the greater the speed-up of the search.

Bell (2009) has shown that node potentials may also be used to speed up the search in the Spiess and Florian (1989) algorithm. This discovery is made use of here.

Reverse time-dependent Hyperstar algorithm

As will be proved later, the following algorithm solves the problem:

A_0	Reverse time-dependent Hyperstar algorithm
1. Initialisation	$u_i \leftarrow \infty, i \in I - \{r\}, u_r \leftarrow 0;$ $f_i \leftarrow 0, i \in I;$ $y_i \leftarrow 0, i \in I - \{s\}, y_s \leftarrow 1;$ $L \leftarrow A;$ $H \leftarrow \emptyset$
2. Select link a	Find $a = (i,j) \in L$ with minimum $u_i + c_a(u_i) + h_j$; $L \leftarrow L - \{a\}$
3. Update node i	If $u_j \geq u_i + c_a(u_i)$ then if $u_j = \infty$ and $f_j = 0$ then $\beta \leftarrow 1$ else $\beta \leftarrow f_j u_j$, $u_j \leftarrow (\beta + \frac{1}{d_a(u_i)}(u_i + c_a(u_i))/(f_j + \frac{1}{d_a(u_i)})$, $f_j \leftarrow f_j + \frac{1}{d_a(u_i)}$ and $H \leftarrow H + \{a\};$ if $L = \emptyset$ or $u_i + c_a(u_i) > u_s$ then go to Step 3 else go to Step 1
4. Loading	For every link $a \in A$ in decreasing order of $u_i + c_a(u_i) + h_j$, if $a = (i,j) \in H$ then $p_a \leftarrow \frac{1}{f_j d_a(u_i)} y_j$ and $y_i \leftarrow y_i + p_a$ else $p_a \leftarrow 0$

Note that to avoid overflow errors this algorithm works only for positive link delays.

Assumption 1: The node potentials satisfy the triangular inequality property

(12) $h_i \leq c_a(u_i) + h_j$ for all $a = (i, j) \in A$

In this case the potentials are said to be *feasible* (Wagner and Willhalm, 2006). The implication of this assumption is that $h_i - h_s$ is less than or equal to the length of the trip from node *i* to destination *s*. Without loss of generality, we can set $h_s = 0$, so feasibility requires that h_i be less than or equal to the remaining distance (or travel time) from node *i* to the destination.

Assumption 2: $u_i > u'_i$ implies $u_i + c_a(u_i) > u'_i + c_a(u'_i)$. The implication of this assumption is that travel time cannot fall faster than real time, or in other words, it is not possible to enter a link later and exit it earlier. This condition is frequently referred as *FIFO* (first in, first out). This corresponds to the FIFO condition of Kaufman and Smith (1993).

Proposition 1: At the point at which $a = (i, j) \in A$ is selected, u_i is reduced to its final value.

Proof 1: The proof is by induction. For all nodes $i \in I - \{r\}$, u_i is set to ∞ (in practice a large number) in Step 0 and then either left unchanged or reduced in Step 2. Suppose that u_i is greater than its final value. There must be a path to *i* not so far considered offering the possibility of a lower travel time. Without loss of generality, suppose this path is via $a^{'} = (i^{'}, i) \in A$ and $a' \notin H$ (the set of links in the hyperpath). Since $u_{i'} + c_{a'}(u_{i'}) < u_i$ and as a result of Assumption 1, $h_i \leq c_a(u_i) + h_j$, so $u_{i'} + c_{a'}(u_{i'}) + h_i \leq u_{i'} + c_{a'}(u_{i'}) + c_a(u_i) + h_j < u_i + c_a(u_i) + h_j$. This implies that in Step 2, link $a^{'}$ rather than link *a* would have been selected. Since link $a^{'}$ was not selected, $u_{i'} + c_{a'}(u_{i'})$ has not been reduced to its final value. By Assumption 2, this implies that $u_{i'}$ is also greater than its final value. This argument can be repeated until we reach origin *r* and conclude that $u_r = 0$ is greater than its final value, which is clearly untrue. QED

In order to solve P_1, Algorithm A_0 must minimise u_s. Proposition 2 shows that upon termination, u_s is minimised.

Proposition 2: Upon termination, Algorithm A_0 has minimised u_s.

Proof 2: According to Step 3, Algorithm A_0 terminates when there are no more links to be selected ($L = \emptyset$) or when the selected link offers no possibility of arriving at the destination earlier ($u_i + c_a(u_i) > u_s$). In either case, u_s cannot be minimised further. From Proposition 1 we know that $u_i + c_a(u_i)$ is minimised when link $a = (i, j)$ is selected. Since $h_s = 0$, then if $u_{i'} + c_{a'}(u_{i'}) \leq u_s$ for any link $a^{'} = (i^{'}, s) \in A \backslash H$, then this link would be selected and added to H. QED

Propositions 1 and 2, however, are not sufficient to ensure that Problem P_1 has been solved by Algorithm A_0, as constraints (8) and (9) must also be satisfied. Proposition 3 shows that constraints (8) and (9) do indeed hold.

Proposition 3: Algorithm A_0 satisfies (8) and (9).

Proof 3: If there is only one link $a = (i, j) \in H$ entering node j, then by Step 3, $u_j = c_a(u_i) + d_a(u_i) + u_i$ and $q_a = 1$. Suppose link $a = (i, j)$ and $a' = (i^{'}, j)$ are in *H*, so $p_a > 0$ and $p_{a'} > 0$. By (7) we see that $c_a(u_i) + q_a d_a(u_i) + u_i = c_{a'}(u_{i'}) + q_{a'} d_{a'}(u_{i'}) + u_{i'}$. Step 3 of Algorithm A_0 ensures that $q_a \geq 0$ and $q_{a'} \geq 0$ exist such that (8) holds. If this were not so, then without loss of generality $c_a(u_i) + d_a(u_i) + u_i < c_{a'}(u_{i'}) + u_{i'}$ would be possible. Step 3, however, ensures that this cannot arise. QED

Note that Proposition 2 requires Proposition 1, which in turn requires Assumptions 1 and 2. At the solution to P_1 according to duality theory P_0 is also solved with $u_s = \sum_{a=(i,j)\in A} p_a c_a(u_i) + \sum_{j\in I} w_j$. Although Algorithm A_0 does not explicitly calculate $q_a, a \in A$, Proposition 3 ensures that values satisfying (8) and (9) exist.

Also note that not all potentially optimal paths are included in *H*, the hyperpath. This is illustrated in Fig. 1. In the brackets, the undelayed travel time and the maximum delay is given for each link. In the absence of any goal orientation ($h_i = 0$ for all $i \in I$), the link from node 1 to 2 is selected first, then the link from node 1 to 3, and finally the link from node 2 to 4, leading to the hyperpath indicated by bold arrows. There is only one elemental path from node 1 to 4 in the hyperpath, namely the path through node 3, despite the fact that the path through node 2 is potentially faster, with a travel time of 4 in the absence of delay.

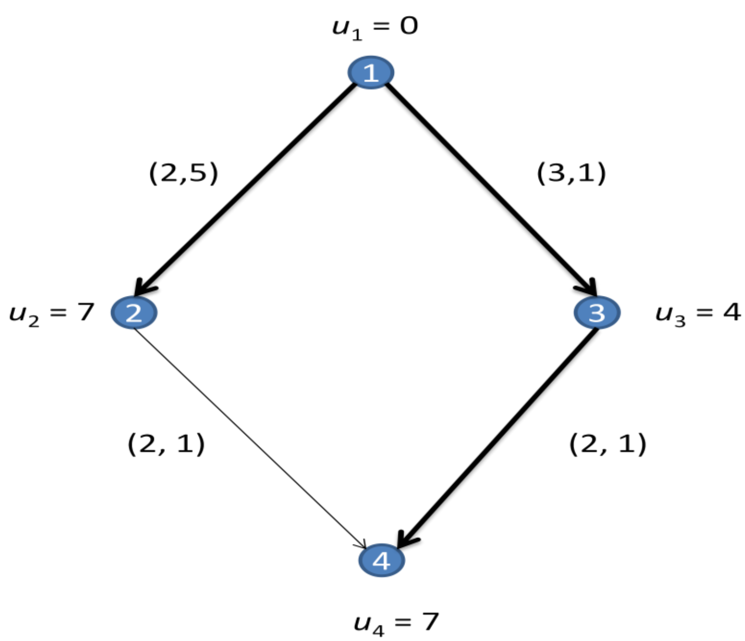

Fig. 1: Example network (for each link, the first number represents undelayed travel time and the second the maximum delay)

Minimising exposure to maximum delay

The proposed approach may be categorised as a *robust* approach, in the sense that uncertainty is represented by an interval. For example, the intervals in Fig. 1 are [2, 7] for the link from node 1 to 2, etc. The implication of constraint (2) in P_0 is that link usage on the approach to every node in the hyperpath is chosen so as to minimise the exposure to maximum delay. The following proposition ensures that if link *a* is used, and therefore node *j* is in the hyperpath, then $p_a d_a(u_i) = w_j > 0$ where $a = (i, j)$.

Proposition 4: If $p_a > 0$ then $p_a d_a(u_i) = w_j > 0$.

Proof 4: If link $a = (i, j)$ is the only link in the hyperpath leading into node j, then $p_a > 0$ and $w_j = p_a d_a(u_i)$. If link $a' = (i', j)$ is also in the hyperpath, then $p_{a'} > 0$ too and Step 4 of Algorithm A_0 ensures that $p_a d_a = p_{a'} d_{a'}$. As w_j is minimised, $w_j = p_a d_a = p_{a'} d_{a'} > 0$. QED

Ensuring FIFO

As already mentioned, there is a necessary assumption that link travel time cannot fall faster than real time (Assumption 2). This is equivalent to the FIFO condition, as it implies that it is not possible to delay entry to a link in order to exit it earlier. One way to ensure that this condition is met while allowing link travel times to be time-dependent is to suppose that each link has a speed profile which defines the speed of all vehicles on the link at any point in time (see Sung et al, 2000). Since at any point in time all vehicles are travelling at the same speed, overtaking is ruled out. Fig. 2 shows an illustrative link speed profile. The vehicle is shown arriving at the link entry in the 7th interval at t_{entry}. Until time t_i it travels at speed $v_a(7)$. Thereafter it travels at the slower speed of $v_a(8)$ until time t_{i+1}, after which it travels at the higher speed of $v_a(9)$ until it reaches the end of the link at time t_{exit}. Link travel time is then t_{exit} minus t_{entry}.

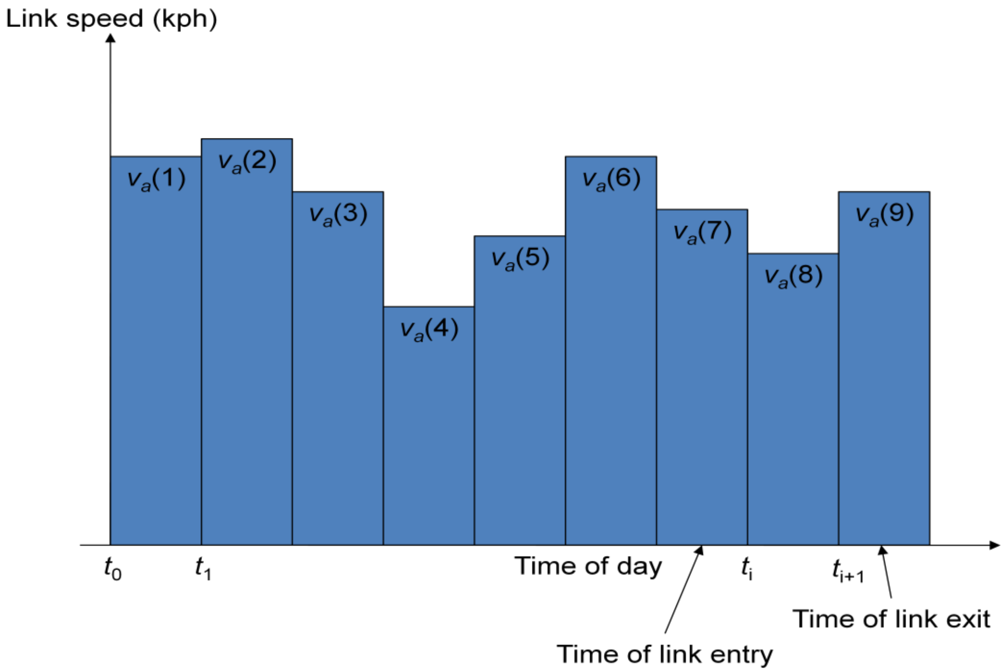

Fig. 2: Example link speed profile

Numerical example

To demonstrate the use of the reverse Hyperstar algorithm, we take an 8 node by 8 node grid network. All arcs can be travelled in either direction, leading to 224 directional links. The length of each arc $l_a, a \in A$, is set equal to 1 + R, where R is a random number in the range 0 to 1 inclusive, and is the same in either direction. The resulting link lengths are shown in Table 1. The origin r is set near the centre of the network to demonstrate the advantages of goal oriented search. The length of each arc is given in Table 1. The travel time on each arc is time-dependent.

Vehicles are assumed to travel at 50 kph for the first 6 minutes of the trip and then at 20 kph thereafter.

As the length of each link is at least 1 unit, it is straightforward to calculate a reasonable value for h_i which underestimates the travel time from node i to destination s. For example, h_{20} = 5/50 = 0.1 (2 links down and 3 across, or any other reasonable route, travelled at the higher speed of 50 kph see Fig. 3). This is a form of what is sometimes referred to in the literature as the Manhattan metric (Wagner and Willhalm, 2006).

Two cases are considered. In the first (see Fig. 3), all links are assigned a near zero maximum delay (in practice, a value of 0.0001 hours). As mentioned earlier, a positive delay is required in order to avoid overflow problems. Since all links are reliable, the hyperpath contains only one elemental path shown by the solid arrows (because link travel times were generated randomly to four decimal places, the chance of two paths having the minimum travel time is small). The diagonal number in Fig. 3 shows the time in hours to reach each node on this elemental path. The transition from the higher speed of 50 kph to the lower speed of 20 kph occurs after 0.1 hours (or 6 minutes), namely between nodes 27 and 19. The introduction of node potentials h as defined above into the algorithm reduces the number of links selected to 157 from 217.

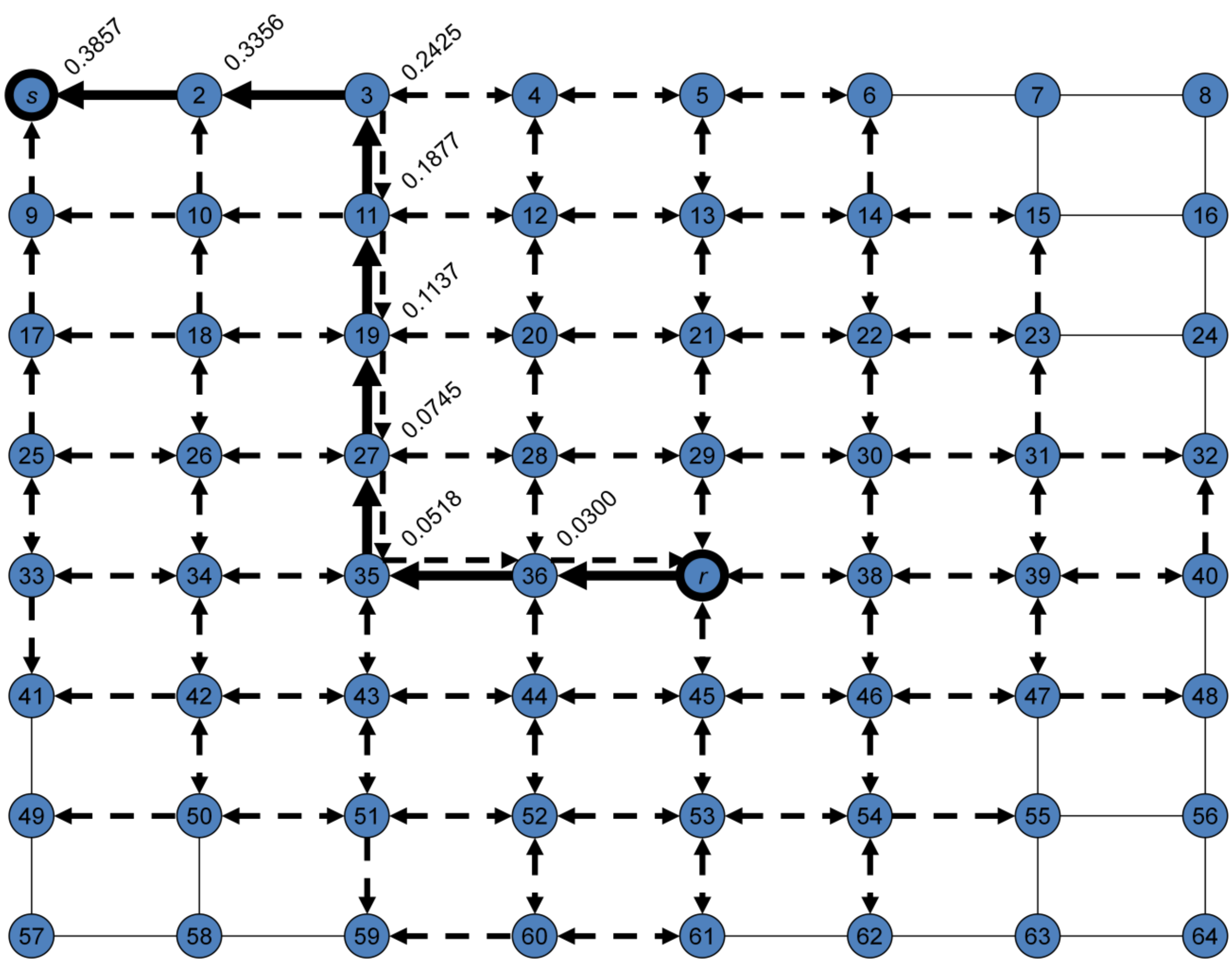

Fig. 3: Fastest path with no risk of delay (solid arrows), showing links selected (dashed arrows)

In the second case (see Fig. 4), maximum link delay is assigned a random value equal to 0.1*R*, where as before *R* is a random value in the range 0 to 1 inclusive. Some links are now significantly less reliable than others, although in this example maximum link delay is not treated as time-dependent. The hyperpath now contains numerous elemental paths. The non-zero link usage probabilities defining link attractiveness are shown against the respective links in Fig. 4. When the risk averse driver reaches node 36, two links are attractive, the link to 28 being more attractive than the link to 35, etc. The diagonal numbers against nodes 36 and *s* show the pessimistically expected travel times taken to reach the corresponding nodes. In comparison to the first case, the increase in the time to reach node 36 is due to an increase in maximum delay from 0.0001 to 0.0882 hours. As there is only one attractive route to node 36 and the driver is pessimistic, the maximum delay is added to link travel time to obtain the pessimistically expected arrival time. Hence the transition from the higher to the lower speed now occurs on the first link. The expected time to reach the destination has approximately doubled from 0.3857 to 0.7536 hours, in part due to the assumed lower speed now starting at node 36 and in part due to the allowance for delay. The introduction of *h* in case 2 leads to a reduction in the number of links selected before termination to 191 from 218.

The advantage of this reverse hyperpath algorithm over single route algorithms is that both time dependency and reasonable detours are taken into account at the route generation stage. The best route in the assumed absence of delays may expose the driver to too high a risk of delay. On the other hand, assuming that every link suffers maximum delay would be too cautious as it neglects the benefits of detouring when the driver receives congestion information en route. Note that when maximum delay is considered, it is more attractive to turn right at node 36 than to continue to node 35 since the maximum delay to arrival at node 35 is 0.0772 hours (a few minutes) compared to 0.0054 hours (a matter of seconds) for node 28.

In practice, maximum delays will rarely be realized, so the driver will typically arrive at each node before the pessimistically estimated arrival time. In this case, the hyperpath search is repeated upon arrival at each node using the actual arrival time rather than the pessimistically expected arrival time. In this context, the use of node potentials to speed up the search becomes particularly important. The amount of acceleration gained by the inclusion of node potentials in this numerical example is relatively modest, in part due to the extent of the underestimate of remaining travel time and in part due to the relatively small size of the numerical example. The estimated time per link is only 0.02 hours, leading to an estimated travel time of only 0.16 hours. Halving of the assumed speed of travel to 25 kph when calculating *h* leads to a reduction in the number of links selected in near absence of delay (case 1) from 157 to 69, but unfortunately the algorithm terminates prematurely without finding the fastest path. This is because the actual speed of travel on a significant number of links near the origin is 50 kph, leading to an over-estimate of link travel time in this region.

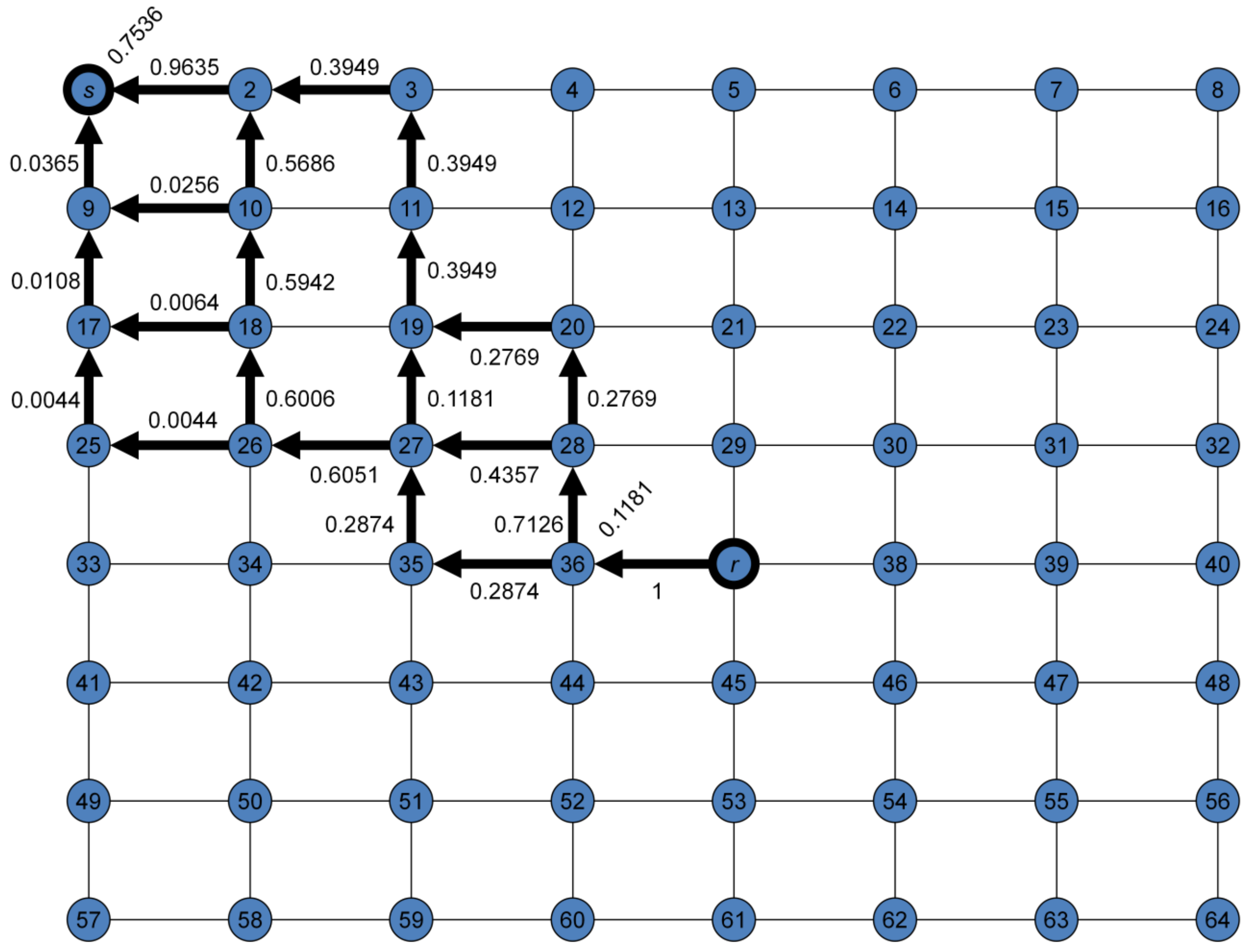

Fig. 4: Hyperpath for uncertain link travel times (number shows link attractiveness)

Conclusions

This paper builds on earlier work published by Bell (2009). In recognition of the prevalence of time-of-day variation in link travel times, an algorithm is presented for building hyperpaths in time-dependent networks from the origin to the destination. Node potentials are included in order to speed up the search, a requirement for vehicle navigation applications, particularly where the search must be repeated upon arrival at each node. The resulting algorithm, referred to in the title as the time-dependent reverse Hyperstar algorithm, may be viewed as the hyperpath version of the Chabini and Lan (2002) algorithm.

Although this algorithm has been presented in the context of risk averse vehicle navigation on road networks, it could also be applied to time tables, opening up the possibility of time-dependent hyperpath searches for transit networks. Allowance for the risk of encountering a maximum delay when connections are missed in transit networks would make the recommendations of public transport journey planners more robust. This is the subject of on-going research.

Acknowledgements

The authors are grateful for financial support from the UK Engineering and Physical Science Research Council under grant EP/E043569/1 on "Dynamic Hyperpath Search for Transit and Traffic Assignment". The other members of the Hyperpath Working Group, namely Achille Fonzone, Jinyoung Kim, Fumitaka Kurauchi, Klaus Noekel, Normen Rochau, Jan-Dirk Schmoecker, Hiroshi Shimamoto, Alan Wilson and Rabi Mishalani have contributed to the formulation of the ideas expressed in this paper.

References

Bell, M G H (2009) Hyperstar: A multi-path Astar algorithm for risk averse vehicle navigation. *Transpn Res B*, **43**, 97-107.

Chabini, I and Lan, S (2002), Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *IEEE Transactions on Intelligent Transportation Systems*, **3**, 60-74.

Chen, Y Y, Bell, M G H and Bogenberger, K (2007) Reliable pretrip multipath planning and dynamic adaptation for a centralised road navigation system. *IEEE Transactions on Intelligent Transportation Systems*, **8**(1), 14-20.

Dijkstra, E W (1959) A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**(1), 269–271.

Gallo, G, Longo, G, Pallottino, S, Nguyen, S (1993) Directed hypergraphs and applications. *Discrete Applied Mathematics*, **42**, 177-201.

Hall, R W (1986) The fastest path through a network with random time-dependent travel times. *Transportation Science*, **20**(3), 182–188.

Hart, P E, Nilsson, N J and Raphael, B (1968) A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems, Science and Cybernetics*, **4**(2), 100-107.

Kaparias, I (2008) Reliable Dynamic In-vehicle Navigation. PhD thesis, Imperial College London.

Kaufman, D and Smith, R (1993) Fastest paths in time-dependent networks for Intelligent Vehicle Highway Systems applications, *IVHS Journal*, **1**, 1–11 (1993)

Miller-Hooks, E D (2001) Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, **37**(1), 35-52.

Miller-Hooks, E D and Mahmassani, H S (1998) Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, **34**(2), 198-215.

Nielsen, L R, Andersen, K A, Pretolani, D (2005) Finding the *K* shortest hyperpaths. *Computers and Operations Research*, **32**, 1477-1497.

Nguyen, S and Pallotino, S (1988) Equilibrium traffic assignment for large scale transit networks, *European Journal of Operational Research,* **37**(2), 176–186.

Park, D and Rilett, L R (1997) Identifying multiple and reasonable paths in transportation networks: a heuristic approach. *Transportation Research Record*, **1607**, 31-37.

Park, K (2009) Learning User Preferences for Intelligent Adaptive In-vehicle Navigation. PhD thesis, Imperial College London.

Park, K, Bell, M G H, Kaparias, I and Bogenberger, K (2007) Learning user preferences of route choice behaviour for adaptive route guidance. *IET Intelligent Transport Systems,* **1**(2), 159-166.

Schmoecker, J-D, Bell, M G H, Kurauchi, F and Shimamoto, H (2009) A game theoretic approach to the determination of hyperpaths in transportation networks. In: *Transportation and Traffic Theory 2009: Golden Jubilee* (Eds: W H K Lam, S C Wong and H K Lo), Springer: Berlin, 1-18.

Spiess, H and Florian, M (1989) Optimal strategies: A new assignment model for transit networks, *Transportation Research Part B,* **23**(2), 83–102.

Sung, K, Bell, M G H, Seong, M and Park, S (2000) Shortest paths in a network with timedependent flow speeds, *European Journal of Operational Research*, **121**, 32-39.

Wagner, D and Willhalm, T (2006) Speed-Up Techniques for Shortest-Path Computations. ARRIVAL-TR-0024 (http://arrival.cti.gr/).

Table 1: Link lengths in km

From	To	Length	From	To	Length	From	To	Length	From	To	Length
1	2	1.0000	33	34	1.9984	1	9	1.3413	29	37	1.1019
2	1	1.0000	34	33	1.9984	9	1	1.3413	37	29	1.1019
2	3	1.8610	34	35	1.6762	2	10	1.8170	30	38	1.5544
3	2	1.8610	35	34	1.6762	10	2	1.8170	38	30	1.5544
3	4	1.2729	35	36	1.0859	3	11	1.0935	31	39	1.1007
4	3	1.2729	36	35	1.0859	11	3	1.0935	39	31	1.1007
4	5	1.3187	36	37	1.4953	4	12	1.0603	32	40	1.0166
5	4	1.3187	37	36	1.4953	12	4	1.0603	40	32	1.0166
5	6	1.3722	37	38	1.5100	5	13	1.6730	33	41	1.9043
6	5	1.3722	38	37	1.5100	13	5	1.6730	41	33	1.9043
6	7	1.0820	38	39	1.9532	6	14	1.5704	34	42	1.3270
7	6	1.0820	39	38	1.9532	14	6	1.5704	42	34	1.3270
7	8	1.0706	39	40	1.3387	7	15	1.3659	35	43	1.9006
8	7	1.0706	40	39	1.3387	15	7	1.3659	43	35	1.9006
9	10	1.0597	41	42	1.7054	8	16	1.3146	36	44	1.3538
10	9	1.0597	42	41	1.7054	16	8	1.3146	44	36	1.3538
10	11	1.9173	42	43	1.7740	9	17	1.6564	37	45	1.0716
11	10	1.9173	43	42	1.7740	17	9	1.6564	45	37	1.0716
11	12	1.7747	43	44	1.1977	10	18	1.9030	38	46	1.7603
12	11	1.7747	44	43	1.1977	18	10	1.9030	46	38	1.7603
12	13	1.6977	44	45	1.9136	11	19	1.4787	39	47	1.9228
13	12	1.6977	45	44	1.9136	19	11	1.4787	47	39	1.9228
13	14	1.7180	45	46	1.5893	12	20	1.7667	40	48	1.1276
14	13	1.7180	46	45	1.5893	20	12	1.7667	48	40	1.1276
14	15	1.1626	46	47	1.6894	13	21	1.5520	41	49	1.4032
15	14	1.1626	47	46	1.6894	21	13	1.5520	49	41	1.4032
15	16	1.4660	47	48	1.0936	14	22	1.4117	42	50	1.8071
16	15	1.4660	48	47	1.0936	22	14	1.4117	50	42	1.8071
17	18	1.8257	49	50	1.2800	15	23	1.5958	43	51	1.1098
18	17	1.8257	50	49	1.2800	23	15	1.5958	51	43	1.1098
18	19	1.4818	50	51	1.1612	16	24	1.6689	44	52	1.9277
19	18	1.4818	51	50	1.1612	24	16	1.6689	52	44	1.9277
19	20	1.8743	51	52	1.5437	17	25	1.4717	45	53	1.3261
20	19	1.8743	52	51	1.5437	25	17	1.4717	53	45	1.3261
20	21	1.7728	52	53	1.1744	18	26	1.1148	46	54	1.2268
21	20	1.7728	53	52	1.1744	26	18	1.1148	54	46	1.2268
21	22	1.4925	53	54	1.2681	19	27	1.5486	47	55	1.6905
22	21	1.4925	54	53	1.2681	27	19	1.5486	55	47	1.6905
22	23	1.8273	54	55	1.5768	20	28	1.0807	48	56	1.6297
23	22	1.8273	55	54	1.5768	28	20	1.0807	56	48	1.6297
23	24	1.1411	55	56	1.2852	21	29	1.8387	49	57	1.5407
24	23	1.1411	56	55	1.2852	29	21	1.8387	57	49	1.5407
25	26	1.5008	57	58	1.6098	22	30	1.9270	50	58	1.7472

26	25	1.5008	58	57	1.6098	30	22	1.9270	58	50	1.7472
26	27	1.5929	58	59	1.7464	23	31	1.7009	51	59	1.2329
27	26	1.5929	59	58	1.7464	31	23	1.7009	59	51	1.2329
27	28	1.7745	59	60	1.0763	24	32	1.5000	52	60	1.7138
28	27	1.7745	60	59	1.0763	32	24	1.5000	60	52	1.7138
28	29	1.7705	60	61	1.1065	25	33	1.1582	53	61	1.9803
29	28	1.7705	61	60	1.1065	33	25	1.1582	61	53	1.9803
29	30	1.5575	61	62	1.8059	26	34	1.9465	54	62	1.0394
30	29	1.5575	62	61	1.8059	34	26	1.9465	62	54	1.0394
30	31	1.6811	62	63	1.6120	27	35	1.1267	55	63	1.0097
31	30	1.6811	63	62	1.6120	35	27	1.1267	63	55	1.0097
31	32	1.9555	63	64	1.7408	28	36	1.3063	56	64	1.3949
32	31	1.9555	64	63	1.7408	36	28	1.3063	64	56	1.3949