SCHEDULE-BASED HYPERPATH APPROACHES TO TRANSIT ASSIGNMENT: THE IMPACT OF IMPERFECT INFORMATION

By Normen Rochau[1,2], Michael G H Bell[1], Klaus Nökel[2], and Achille Fonzone[1]

[1]PTV AG, Karlsruhe

[2]Centre for Transport Studies, Imperial College London

ABSTRACT

In a schedule-based capacitated transit network we analyze optimal passenger strategies, where a strategy is a ranked set of services at each interchange in the transit network. The passenger is assumed to take the first ranked service unless this is full in which case he chooses the second ranked service, etc. In this equilibrium model passengers choose strategies to minimize their expected cost of travel. This cost depends not only on waiting times and travel times but also on failure-to-board probabilities. These probabilities depend on the loading situation of the services. The more passengers try to use an overcrowded service, the less likely it is that they are able to board the vehicle. The model is asymmetric in the sense that passengers that are already on board have priority. This model, originally proposed by Hamdouch et al. (Hamdouch and Lawphongpanich 2008), assumed that passengers are aware of the complete network, the complete schedule and the loading situation, i.e. the failure-to-board probabilities. We analyze how imperfect information affects optimal passenger strategies and the overall loading situation in several dimensions. We also analyze how passengers who don't take all the strategies into account fare in comparison to those who do. Once a passenger has missed his preferred connection, he might not be aware of or able to determine the connection that is now best for him. Hence we analyze how optimal strategies alter when passengers consider average headways instead, thereby effectively combining schedule- and frequency-based assignment.

Keywords: Transit assignment, capacity constraints

INTRODUCTION

Path choices models for public transit are usually classified as either headway-based or schedule-based. In headway-based models (For example Spiess and Florian 1989; De Cea and Fernández 1993; Lam, Gao et al. 1999; Cominetti and Correa 2001) passengers are assumed to calculate expected waiting times based on the headways of each line. On each stop, a passenger divides the set of lines into attractive and unattractive lines. Since there is no schedule, it is not clear which line comes first. The passenger boards the first vehicle of an attractive line. This means that it is not clear, which path the passenger will take. In fact, he travels on a so-called *hyperpath* (Nguyen and Pallottino 1988).

In schedule-based models (For example Tong and Wong 1999; Nuzzolo, Russo et al. 2003; Wilson and Nuzzolo 2004) passengers are aware of the schedule and choose their path accordingly. Often it is assumed that there are differences in the perception of travel costs (in-vehicle time, waiting time, walking time, etc.) for different passengers. Therefore, path costs in many models contain a random component. This leads to a better spatial and temporal distribution of passengers compared to simple shortest path models.

More recent models (For example Poon, Wong et al. 2004; Hamdouch and Lawphongpanich 2008) also consider capacity constraints of the vehicles and congestion inside of the vehicles and on the platforms. Hamdouch et al. presented an approach to model passenger behaviour in networks with capacity constraints. When vehicle capacities are limited, it is possible that they are so crowded that some passengers cannot board. Therefore, passengers can no longer rely on a selected path; they have to determine a strategy. A strategy – according to Hamdouch et al. – consists of an ordered set of choices on each stop. Passengers are queuing on the platform. If the desired vehicle cannot be boarded, the passenger tries to board the second vehicle of his choice and so on. The model is asymmetric in the sense that passengers who are already on board have priority to passengers trying to board. A key difference between this model and models without capacity constraints is that passengers may end up on a path which they did not select in the first place. Passengers who enter the network at the same time in the same place with the same destination and preferences may use different paths and reach the destination at a different point of time. Congestion may be seen as a form of unreliability. Even if a vehicle reaches a stop according to schedule, it is not useful for a passenger who is unable to board.

A possible drawback of path choice models is that they usually presume that the passengers have a lot of knowledge. In order to calculate a shortest path it is necessary to know all connections on all potential intermediate stops. If vehicle capacities are limited, it is also necessary to have a clear concept of how likely it is that a vehicle can be boarded. This means that passengers also have to know the loading situation – in advance. While it is a viable assumption that passengers learn from experience what the best choices are in uncongested networks, we believe that this assumption does not hold in congested networks. Too many things can happen in congested networks, so that passengers will restrict their choice set to a size they can handle. We are therefore going to analyze how passengers with limited knowledge and limited access to information fare in a network which is unreliable. We take into account that information may be revealed to the passenger during his trip, for example through online information at stops.

A word on terminology: The term “optimal strategy” is usually used in the context of headway-based assignment. Since strategies as defined by Hamdouch et al. are also “optimal”, we are going to use the term optimal SB-strategy.

OPTIMAL SB-STRATEGIES

In this section we are going to explain how optimal SB-strategies may be calculated in case of full information. We are going to generalize Hamdouch's model slightly by introducing reliability of a service. In Hamdouch's model, the probability r of successfully boarding a vehicle is a calculation result based on a stop model. This calculation result is then used to calculate an optimal SB-strategy for a passenger. We treat r as input and call it *reliability*. Reliability may reflect the chances to board vehicle when there is a lot of congestion inside of the vehicle or on the platform. It may, however, also be used to reflect vehicle delay. Finally, it may reflect the fact that walking times may vary in a crowded station. If a station is crowded a passenger may miss his connection, even though there is no congestion on the vehicle or on the platform where the vehicle departs. Although we treat reliability here as a deterministic function, it may depend on the passenger's perception and thus contain a random component.

In the following we are briefly going to explain how to find an optimal SB-strategy based on reliabilities. This procedure is explained in detail in (Hamdouch and Lawphongpanich 2008).

Suppose the network together with the timetable is modelled as a time-expanded network $G(N, A)$. We assume that G is a directed, simply-connected graph without directed cycles. A node $s \in N$ has a time coordinate $t(s)$. For all arcs the time coordinate of the tail node is smaller than the time coordinate of the head node. The time coordinates induce a partial order on the set of nodes. The set of nodes may be partitioned into a partition P. The elements of P represent the same location at different times. An element O of P is called a *station*. We use an identifier O to describe an element of P or a subset of N interchangeably. In a time-expanded network an arc corresponds to a vehicle. We therefore use these terms interchangeably.

In an SB-strategy the choices on each node s correspond to an arc (s, t). They are ordered simply by their cost b', which is the sum of the cost b of the target node t plus the cost of the arc (s, t):

$$b'(t) = c(s, t) + b(t) \tag{1}$$

We define the probability that a passenger can board a certain vehicle as r. In graph terms $r(a) \in [0,1]$ is the reliability of arc a. Notice that an arc, which a passenger is not aware of, can be modelled with reliability 0. The probability that a certain arc a is used by a passenger – based on his order of choices is

$$\mathrm{p(a)} \coloneqq \mathrm{r(a)} \cdot \prod_{\mathrm{a'} \in \mathrm{d_a}} (1 - \mathrm{r(a')}) \tag{2}$$

Here d_a is the set of arcs that come before a in the passenger's choice set. We can use $p(a)$ to define the cost b of node s:

$$b(s) \coloneqq \sum_{a \in d(s)} p(a) \left(c(a) + b(h(a))\right) \tag{3}$$

Here $h(a)$ is the head node of arc a and $d(s)$ is the set of outbound arcs from node s. Thus we have a recursive formula to define node costs. This function is well-defined because the graph is acyclic.

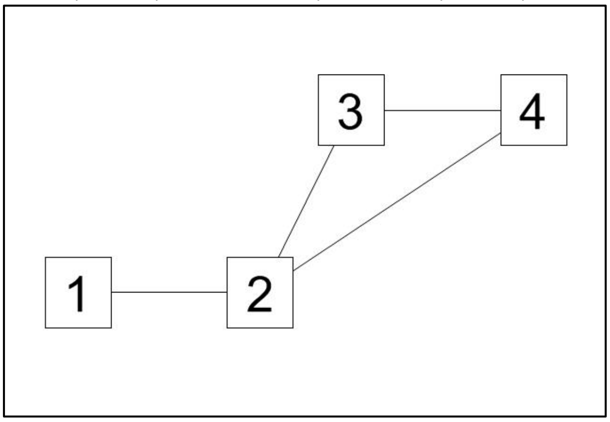
Suppose a passenger wishes to arrive at a given destination in a certain desired time interval. This is represented as a target set $S \subset N$ in the time-expanded network. The costs of reaching that target set may be calculated by a simple procedure that consists of a backward pass and a forward pass.

Algorithm: Backward pass
Input: A time-expanded network G, a target set $S \subset N$
Output: A set $R \subset N$ from which target set S may be reached reliably, cost $b(n)$ for every node $n \in R$

(1) (Initialization) $R = \emptyset, Q = S$
(2) (Selection) **If** $Q = \emptyset$ **Then** Terminate
Else Select $q \in Q$ such that $t(q) = \max_{q' \in Q} t(q')$.
(3) (Sorting) Sort $\delta^+(q) \cap R$ according to $b(q)$
(4) (Probability) Determine probabilities according to sorting with equation (2)
(5) (Cost) Determine $b(q)$ with equation (3)
(6) (Update) $R = R \cup \{q\}; Q = Q \backslash \{q\} \cup \left(\delta^-(q)\right)$ Go to Step 2.

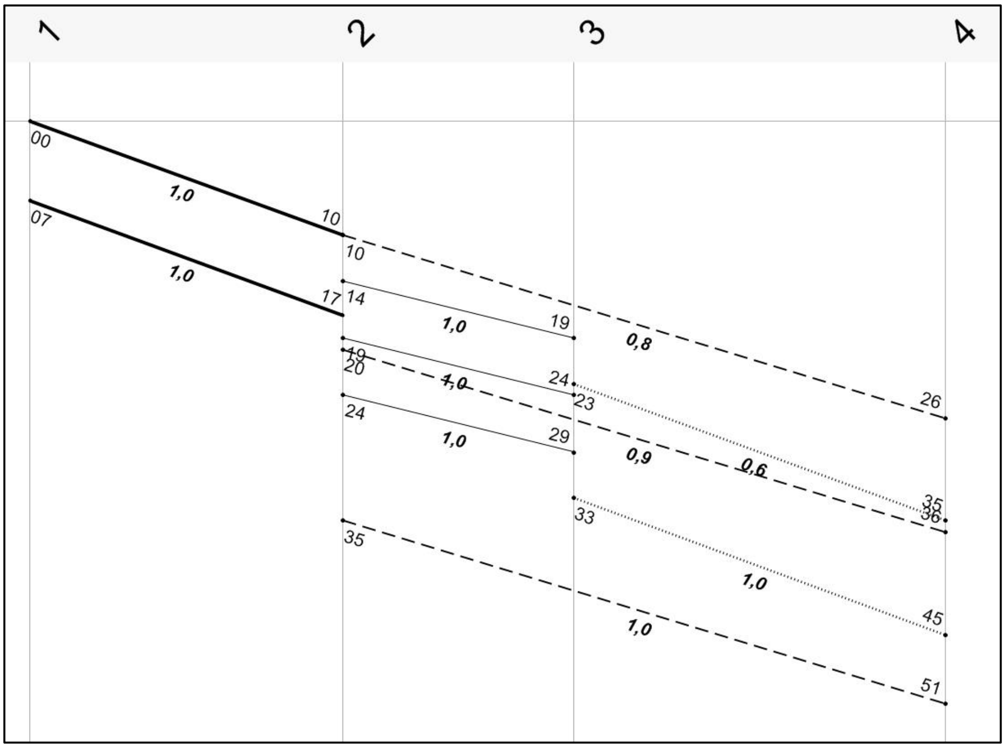
Description: In Step 1 we initialize the working set Q and the "done" set R. In Step 2 we select a node in the working set. It is a node q with a maximum time coordinate $t(q)$ of all nodes in Q. Notice that a computer will need an (arbitrary) secondary criterion if there are nodes in the working set that have the same time coordinate. In Step 3 the successor nodes of node q are sorted according to their cost b. Notice that we are only interested in the successor nodes that are contained in the "done" set R, i.e. that lie on a path to the target set S. In Step 4 the successors are sorted according to their cost. Then the probabilities of using the connections that are represented by the arcs are calculated with equation (2). In Step 5 the cost of node q is calculated with equation Y. Finally, in Step 6 the "done" set R and the working set Q are updated. Node q is removed from the working set and its predecessors are inserted. Notice that this rule does not guarantee that the probabilities of the successor nodes add up to one. It makes sense, however, to just take into account nodes from which we may reach the target set S reliably. Therefore, in Step 6 we should only include nodes that have a reliable arc into the set R. Notice that this explicitly includes nodes that lie in the same station as the target set S, but earlier. In fact, this may be the only reliable connection into S.

An example can be seen in figure 1. The network consists of four nodes. The optimal path from node 1 to node 4 when there is no congestion goes from node 1 to node 2 and then to node 4.



1: The network.

The time-expanded network can be seen in figure 2. The best connection is to leave node 1 at 7:00 and then board the vehicle arriving at 7:10 going directly to node 4. In an unreliable network the passenger may miss that connection. In that case he may board a vehicle to node 3 at 7:14 and leave node 3 at 7:23. This alternative is still faster than waiting for the next direct connection from node 2 to node 4 at 7:20. Note that the time-expanded network also contains all waiting arcs. In figure 2 these arcs would be vertical, connecting nodes that belong to the same station.


2: The Time-Expanded Network.

The backward pass algorithm calculates the following costs and ordered choice sets for each node. The rows are ordered by their time coordinate, which is the order in which the backward pass algorithm processes the nodes.

Node	Choice Set	Cost
4 / 7:51	Empty	0
4 / 7:45	Empty	0
4 / 7:36	Empty	0
4 / 7:35	Empty	0
2 / 7:35	4 / 7:51	16
3 / 7:33	4 / 7:45	12
3 / 7:29	3 / 7:33	16
4 / 7:26	Empty	0
3 / 7:24	3 / 7:33	21
2 / 7:24	3 / 7:29	21
3 / 7:23	4 / 7:35; 3 / 7:24	0.6*12+0.4*22=16
2 / 7:20	4 / 7:36; 2 / 7:25	0.9*16+0.1*25=16.9
2 / 7:19	2 / 7:20	17.9
3 / 7:19	3 / 7:23	20
2 / 7:17	2 / 7:19	19.9
2 / 7:14	2 / 7:17	22.9
2 / 7:10	4 / 7:26; 2 / 7:14	0.8*16+0.2*26.9=18.18
1 / 7:07	2 / 7:17	29.9
1 / 7:00	2 / 7:10	28.18

The choice sets have at most two elements. They only have more than one element when there is an unreliable arc. For example, in node 3/7:23 the passenger has to decide whether she wants to board the approaching vehicle or not. Boarding the vehicle clearly is the better option. Arc (3/7:23; 4/7:35) is not very reliable. This affects the cost of the upstream nodes. At node 2/7:14 the passenger has to decide whether to board the approaching vehicle or not. She decides to wait for the next vehicle, because the cost of going to node 3/7:19 would be $20 + 5 = 25$, whereas the cost of waiting only is 22.9.
The result of the backward pass can be used to calculate optimal SB-strategies from any node in R to S. In order to determine the optimal SB-strategy for a specific passenger starting at station O, a forward pass is necessary. This simply consists of finding a node $o \in O$, such that $b(o) = \min_{n \in O} b(n)$. The sorted choice sets of the strategy are defined by the cost function b, whose value was determined in the backward pass. The passenger only travels to nodes in R.

Algorithm: Forward Pass
Input: A time-expanded network $G(N, A)$, node costs b, defined on a subset $R \subset N$, a station $O \in P$ with $O \cap R \neq \emptyset$.
Output: An optimal SB-strategy to travel from O to S.

1. (Initialisation) $Q = \{o\}c$, where $o \in O$ such that $b(o) = \min_{n \in O} b(n)$
2. (Selection) **If** $Q = \emptyset$ **Then** Terminate
   Else Select $q \in Q$ such that $t(q) = \min_{q' \in Q} t(q')$
3. (Sorting) Sort $\delta^+(q) \cap R$ according to b
4. (Update) $Q = (Q \setminus \{q\}) \cup (\delta^+ \cap R)$ Go to Step 2.

In the example above, *O* is node 1. The best element of that station is the node 1/7:00. The forward pass picks up all nodes that lie on the path to the destination and discards all nodes that are not on that path. The resulting strategy looks very much like the one in table 1, except that node 1/7:10 is missing because node 1/7:00 is more attractive.
Notice that it may make sense to restrict the set in Step 4 even further, so that the nodes are added only up to the first node that can reliably be reached. This way, nodes 2/7:35 and 4/7:51 can be discarded from the strategy, because at node 2/7:24 the passenger can reliably board the arriving vehicle.

IMPERFECT INFORMATION AND IMPERFECT KNOWLEDGE

In the scenario above, a passenger who wishes to determine the optimal strategy to navigate through the unreliable network needs to know the complete physical network, the schedule, and the loading situation of each vehicle and each platform. This is a lot of knowledge. We are going to analyze how the passenger's choice of strategy is affected if he has limited knowledge or for some reason refrains from choosing the theoretically optimal options. This may happen if a passenger does not wish to leave his original route or his original sub-network.

Dimensions of Knowledge and Passenger Types

Consider the following scenario: A tourist in an unfamiliar city selects a path based on some type of information system, for example a website provided by the transit operator. The passenger does not know that the path he chose is not completely reliable. At some point he fails to board a vehicle, for example due to congestion. Now he has to find a strategy to navigate through the network based on the information he has. It is likely that the passenger will stick to his original route. However, travellers who have seen a lot of cities may start navigating according to an underground map and assume average headways. We may assume, though, that this passenger is not familiar with the bus network. A third passenger may be on her morning commute. She knows that the network is unreliable. In case she fails to board a certain vehicle, she takes into account a limited set of options to reach her workplace on a different route. This means she travels on a sub-network.
As we can see there are several factors that affect choice in unreliable networks, or when passengers have limited information:

- The knowledge of the network,
- the knowledge of the timetable and / or the headways,
- whether the passenger knows that the network is unreliable (and he knows the reliabilities of the connections),
- whether the passenger will stick to his original path or an original sub-network,
- and, connected to the previous point, whether the traveller will be able to gather additional information throughout the trip and will make use of it.

We will look at two examples: The first is a commuter. This commuter is aware that the network is unreliable and that she can't use the theoretical shortest path according to the timetable, because she will arrive late at work. The question is, when the optimal time to start her journey is. The second example is a tourist who chose a path based on pre-trip information. He does not know that the connection is unreliable. The path contains a connection with a very large headway. This means that the passenger when he fails to board this vehicle will reroute based on assumptions about line headways.

SAME ITINERARY / SAME SUB-NETWORK

In this scenario the passenger restricts her choices with respect to the lines and routes she will take. We assume that there is some optimal path which she has in mind, but she is aware that this path is unreliable. There are several sensible possibilities how choices can be restricted:

1. Use only the same lines as in the original path.
2. Use only the same route, but possibly use other lines on that route.
3. Use a limited number of possible routes.

The question is what the original path or set of paths is. In an unreliable network it is possible that the shortest path according to the schedule is not contained in this set because it is very congested.

A heuristic to find such a set of good paths is the following algorithm:

1. Calculate an optimal SB-strategy on the complete time-expanded network
2. Determine the optimal path / optimal sub-network which is contained in that strategy
3. Reduce network graph
4. Calculate an optimal SB-strategy on a reduced network, that at each station contains only arcs of the optimal line / optimal route / optimal sub-network

Step 1 and Step 4 are carried out using the original backward pass algorithm. In Step 2 a modified version of the forward pass algorithm can be used. In the updating step of that algorithm only the best options according to some criterion are inserted into the working set Q. In Step 3 the network is reduced severely so that it only contains arcs that correspond to the optimal path or sub-network. These are arcs that belong to the same line as the arcs of the shortest path, the same route or the same physical sub-network.

In our example the optimal path contained in the optimal SB-strategy is to start at node 1/7:00, go to node 2/7:10 and then node 4/7:26. This connection is not reliable, though. The passenger may decide that thinking about going to node 3 and taking the chance of missing the connection at node 3/7:23 is too complicated. He may restrict himself to path 1-2-4.

Notice that in the algorithm we cannot omit Step 4. A passenger using a strategy resulting from Steps 1-3 may not reliably arrive at the destination within the time frame defined by the target set S, because step 3 may have removed the only reliable forward arcs from some intermediate node. In this case, the passenger may not arrive at his destination at all! Therefore, a reliable strategy has to be calculated in the reduced network in Step 4.

Looking at the example we see that all connections from node 1 via node 2 to node 4 that are in the optimal SB-strategy are unreliable. This means that the strategy the passenger chose is node feasible according to our definition. The backward pass algorithm has to be executed again. The resulting strategy will contain the third, reliable trip from node 2/7:35 to node 4/7:51.

Also notice that in general there may be a better strategy than the one resulting from our procedure, which also satisfies the given constraints. There is no reason to believe that the

optimal path we constructed from the optimal SB-strategy on the full network has anything to do with an optimal strategy on a reduced network.
A second heuristic would be to calculate the actual shortest path in the network into target set S assuming that the network is reliable. Then an optimal SB-strategy search could be performed on the time-expanded network induced by the optimal path. Again, this is a heuristic and there is no reason to believe that this is an optimal choice. It seems that this algorithm may not work so well when there is a lot of congestion in the centre of a network which may be avoided with minor detours. This, however, has yet to be shown with real-world examples.

Headway-Based

Let us now assume that the passenger is not aware of a schedule and travels based on (perceived) line headways. This means that she will travel on a hyperpath. For this, she has to determine at each node, which lines are attractive and which are not. When a vehicle arrives at the station, the passenger boards it if it belongs to an attractive line. Otherwise the passenger waits for the next vehicle. In the traditional static headway-based models this behaviour leads to a distribution of passengers on various paths. In a schedule-based model it is possible to predict the route of the passenger if we assume that she can board every vehicle she chooses (and assuming that attractive vehicles never arrive at the same time). The hyperpath then collapses to a single path. If we assume that there are capacity constraints, the hyperpath is replaced by an optimal SB-strategy. The algorithm to find a strategy based on headways in a schedule-based network is the same as the backward pass algorithm above, except that there are differences in Step 3 and Step 5.

Algorithm: Backward pass based on headways
Input: A time-expanded network G, a target set $S \subset N$
Output: A set $R \subset N$ from which target set S may be reached reliably, cost $b(n)$ for every node $n \in R$

(1) (Initialization) $R = \emptyset, Q = S$
(2) (Selection) **If** $Q = \emptyset$ **Then** terminate
Else select $q \in Q$ such that $t(q) = \max_{q' \in Q} t(q')$.
(3) (Sorting) Determine which arcs in $\delta^+(q) \cap R$ belong to an attractive line
(4) (Probability) Determine probabilities according to sorting with equation (2)
(5) (Cost) Determine hyperpath cost of q: $b(q) = \frac{1+\sum_{l \in L} \lambda(l) b(l)}{\sum_{l \in L} \lambda(l)}$, where L is the set of lines at the station, $\lambda(l)$ is the frequency and $b(l)$ the cost of line l
(6) (Update) $R = R \cup \{q\}; Q = Q \backslash \{q\} \cup \left(\delta^-(q)\right)$ Go to Step 2.

Step 3: In the original algorithm, the options have to be sorted by their costs. Here, we have to decide whether a vehicle that is about to leave is attractive. In order to determine whether a vehicle belongs to an attractive line, it is not necessary to know the cost of the successor node of the waiting arc. Notice that in the time-expanded network the outbound degree of each node is usually 2, because the alternatives are to board a certain vehicle or to wait. If there is more than one vehicle of an attractive line at the stop at the same moment, these

lines are sorted according to their cost. This is determined as in equation (1) by the cost of the arc and of the successor node.
Step 5: The cost of the node has to be determined for inbound arcs that come from another station. The difference to the original backward pass is that the cost b of a node is determined differently. It is the hyperpath cost of the node, which is based on line headways. Notice that we may have to make use of "perceived" headways. Since we have a schedule the headway between individual vehicles of the same line may differ. Furthermore, given two arcs of the same line, the cost of their successor nodes in the time-expanded network of may still differ, even if the successor nodes belong to the same station. Calculation of the attractive set then has to be based on the cost of the successor node of the first outbound arc of each line.
These properties can be seen in the example. The vehicles departing from node 2 to node 4 have headways of 10 minutes and 15 minutes respectively. The costs of the successor nodes in this case is the same for all successor nodes.

Headway-Based and Local Information

In network that has a schedule it is likely that the passenger is able to get additional information when he reaches a station. The additional information may affect his judgement of which vehicles are attractive and which are not. For example there could be dynamic passenger information about when the next vehicle of each line is expected. Or there could be information about the local schedule. The additional information will lead to a new estimate of the expected travel costs.
If we compare this scenario to the full-information scenario we described first, we notice that the passenger has full local information, including the expected loading situation. The only information he has not is the schedule information for other stations.
This is reflected in the backward pass algorithm. Step 3 is the same as in the original algorithm. The passenger knows the waiting time, the cost of the outbound arcs and the cost of their successor nodes. He also may know the reliabilities. The difference to the original scenario lies in Step 5. Since the passenger only has local information, the influence of that knowledge on the cost function should not "propagate" through the network. This means that once the passenger reaches a station his estimate of the remaining travel time will change. It will be influenced by the local information. This means that we need two cost functions, one for predecessor nodes on the same station and one for predecessor nodes on different stations. In Step 5 we calculate both cost functions. The "local" cost of the node will be determined as in the original algorithm using waiting times. The "remote" cost of the node will be based on hyperpath costs as in the second algorithm.

Combining the Headway-Based and the Schedule-Based Cost Functions

If we now put together the headway-based model and the schedule-based model, we can emulate the behaviour of the passenger who starts out on a shortest path and ends up travelling based on guesses about average headways.

For this we first perform a simple shortest path search on the time-expanded network. This search does not take capacity constraints into account. This is what a contemporary internet platform would provide. However, as we have seen a path is not enough to navigate successfully in an unreliable network. If the reliability of the given path is not equal to 1, the passenger needs to have fallback options.

We may predict the passenger’s behaviour by performing a backward pass search in the time-expanded network which is based on line headways and local information. The strategy of the passenger will be constructed by joining the information from the shortest path search and the headway-based search. On each node in the time-expanded network which lies on the shortest path, the successor is defined by the shortest path given by the internet platform. The order of the other options is determined by the headway based search. When the passenger reaches a node that is not in the shortest path, we can assume that he missed a connection. Choices at this node are ordered by the headway-based search.

Notice that this procedure can even be refined. If the passenger is at a node which is not on the shortest path, but there is a choice that will lead to that path (and the passenger is aware of that), the passenger will probably use this option, even if the hyperpath-based cost tells him differently.

In our example the optimal path the passenger would start with would start at node 1/7:00, go to node 2/7:10 and then node 4/7:26. If the passenger fails to board the second vehicle, he can use online information to determine the waiting times for each vehicle at station 2. However, since there is no information available about station 3, the costs would have to be based on average (or perceived) headways.

CONCLUSION

Using strategies in schedule-based assignment gives us a wide range of possibilities to model different kinds of passenger behaviour in detail. The differences in behaviour are based on different levels of knowledge, different levels of information and the purpose of travel. When the passenger obtains new information his assessment of different options may change. Strategies in schedule-based assignment give us a way to deal with capacity constraints. They correctly lead to spatial and temporal distribution of the passenger flows.

A new motivation to model passenger behaviour in such detail is that this may be necessary to provide good (online) information for passengers. This can be compared to handheld devices that take into account accidents on motorways. However, it is not clear yet how the reliabilities should be calculated. In order to do this well it might be necessary to have very good OD matrices.

Cominetti, R. and J. Correa (2001). "Common-Lines and Passenger Assignment in Congested Transit Networks." Transportation Science **35**(3): 250.

De Cea, J. and E. Fernández (1993). "Transit Assignment for Congested Public Transport Systems: An Equilibrium Model." Transportation science : the publication of the Transportation Science Section, Operation Research Society of America **27**(2): 133-147.

Hamdouch, Y. and S. Lawphongpanich (2008). "Schedule-based transit assignment model with travel strategies and capacity constraints." Transportation Research Part B: Methodological **42**(7-8): 663-684.

Lam, W. H. K., Z. Y. Gao, et al. (1999). "A stochastic user equilibrium assignment model for congested transit networks." Transportation research B **33**(5): 351-368.

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

Nuzzolo, A., F. Russo, et al. (2003). Transit Network Modelling: The schedule-based dynamic approach. Mailand, Italien, FrancoAngeli.

Poon, M. H., S. C. Wong, et al. (2004). "A dynamic schedule-based model for congested transit networks." Transportation Research Part B: Methodological **38**(4): 343-368.

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

Tong, C. O. and S. C. Wong (1999). "A stochastic transit assignment model using a dynamic schedule-based network." Transportation Research Part B: Methodological **33**(2): 107-121.

Wilson, N. H. M. and A. Nuzzolo, Eds. (2004). Schedule-based dynamic transit modeling: Theory and applications. Operations research/computer science interfaces series. Dordrecht, London, Kluwer Academic.