Dynamic Journeying under Uncertainty

Lauri Häme*,[1]

Aalto University School of Science, PL 14100, 00076 Aalto, Finland

Harri Hakula

Aalto University School of Science, PL 14100, 00076 Aalto, Finland

---

Abstract

We introduce a journey planning problem in multimodal transportation networks under uncertainty. The goal is to find a journey, possibly involving transfers between different transport services, from a given origin to a given destination within a specified time horizon. Due to uncertainty in travel times, the arrival times of transport services at public transport stops are modeled as random variables. If a transfer between two services is rendered unsuccessful, the commuter has to reconsider the remaining path to the destination. The problem is modeled as a Markov decision process in which states are defined as paths in the transport network. The main contribution is a backward induction algorithm that generates an optimal policy for traversing the public transport network in terms of the probability of reaching the destination in time. By assuming history independence and unconditionality of successful transfers between services we obtain approximate methods for the same problem. Analysis and numerical experiments suggest that while solving the path dependent model requires the enumeration of all paths from the origin to the destination, the proposed approximations may be useful for practical purposes due to their computational simplicity. In addition to on-time arrival probability, we show how travel and overdue costs can be taken into account, making the model applicable to freight transportation problems.

Key words: Stochastic processes, Transportation, Stochastic shortest path problem, Itinerary planning problem, Markov decision processes

---

1. Introduction

The urban itinerary planning problem involves determining a path, possibly involving transfers between different transport modes, from a specified origin to a similarly specified destination in a transport network. Common criteria used for evaluating itineraries include the total duration, number of transfers and cost (Androutsopoulos and Zografos, 2009).

We study a new objective for journey planning in scheduled public transport networks, motivated by uncertainty in travel times of transport services (buses, trams, trains, ferries, ...). Our

*Corresponding author

Email addresses: `Lauri.Hame@tkk.fi` (Lauri Häme), `Harri.Hakula@tkk.fi` (Harri Hakula)

[1]Tel.: +358 40 576 3585, Fax: +358 9 470 23016

goal is to maximize the reliability[2] of an urban journey. In contrast to existing itinerary planning algorithms designed for scheduled public transport networks, where the path is a priori optimized with respect to an objective, for example, (Androutsopoulos and Zografos, 2009), we take into consideration the fact that the realized journey may differ from the original plan.

Real-time information on the status of transport services is available via mobile devices with location-based capabilities. This makes it possible for a commuter to dynamically modify the planned journey in case of a delay or cancellation. For example, if a transfer from a transport service to another is unsuccessful due to a delay, the commuter may reconsider the remaining path to the destination. Our approach is to design the journey in a way that the probability of reaching the destination in time is maximized, even if some transfers are rendered unsuccessful in the course of time. Clearly, the importance of reliability is emphasized when the number of transfers between different transport services is increased.

Taking into account the uncertainty in travel times is particularly important in difficult weather conditions when delays are common. In addition to traditional public transport with fixed schedules, uncertainty in travel times should be given special attention in flexible transport services without fixed routes (Mulley and Nelson, 2009). If the vehicle routes are modified in real time, the estimation of travel times between subsequent stops is more difficult than in the case of fixed routes.

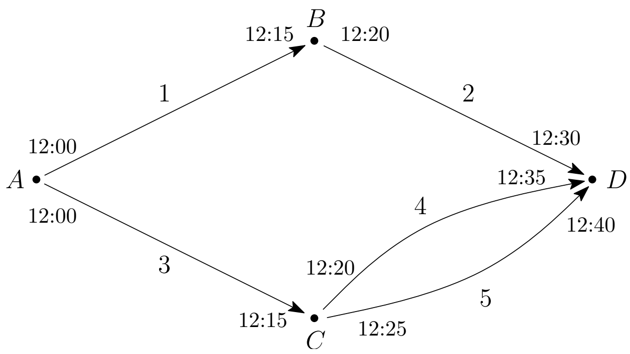

Figure 1: The difference between stochastic and deterministic journey planning for a commuter traveling from A to D. The four points represent public transport stops (A, B, C, D) and the arrows between them represent public transport services $(1, \ldots, 5)$. Initially, there are three possible journeys from A to D: $(1, 2), (3, 4)$ and $(3, 5)$. If the commuter initially chooses service 1, the success of the journey is dependent of the success of the transfer from 1 to 2 at stop B. If the commuter chooses service 3 first, the destination is reached if one of the transfers $3 \to 4$ or $3 \to 5$ is successful at stop C.

A simplified example clarifying the main difference between stochastic and deterministic journey planning is shown in Figure 1. The four nodes represent public transport stops (A, B, C, D) and the arrows between them represent scheduled public transport services $(1, \ldots, 5)$ operating

[2]In the scheduling of an activity of random duration in general and in traveling under congested conditions in particular, the value of reliability is seen to be significant (Fosgerau and Karlström, 2010). Brownstone and Small (2005); Small et al. (2005) argue that there is substantial heterogeneity in the valuation of reliability among motorists. In an empirical study on commuter behavior in California, where commuters chose between a free and a variably tolled route, the model in (Lam and Small, 2001) suggested that the average value of reliability is \$15.12 per hour for men and \$31.91 for women (in 1998 US dollars).

within a specific time horizon. Each transport service has a specific schedule determined by the scheduled departure and arrival times shown next to the arrows.

(i) Let us first consider the deterministic case where the *realized* departure and arrival times of services are assumed to be equal to the *scheduled* departure and arrival times. For a commuter traveling from A to D, there are three feasible journeys: $(1, 2)$, $(3, 4)$ and $(3, 5)$. In order to reach D as fast as possible, the commuter should follow the path $(1, 2)$.

(ii) In the stochastic case, the realized departure and arrival times of services are not necessarily equal to the scheduled times. A transfer from a service to another may fail due to a delay, even if the transfer was feasible according to the deterministic schedule. For example, if the commuter initially chooses service 1, the success of the journey from A to D is dependent of the success of the transfer from 1 to 2 at stop B. Assuming that services 1 and 3 are equally likely to be delayed, it might be reasonable to initially choose service 3: It is more probable that one of the transfers $3 \to 4$ and $3 \to 5$ is successful than that $1 \to 2$ is successful.

More generally, in the model presented in this paper, a commuter wishes to travel from an origin node v_o to a destination node v_d within a time horizon $[0, T]$ using different transport services. Each transport service is represented as a sequence of *legs*. Each leg is associated with a *start node* and *end node*, as well as a random *start time* and *end time*. Adjacent nodes in the network are connected with similarly defined walking legs.

A path from the origin to the destination is represented as a sequence of legs, in which the start node of each leg is equal to the end node of the previous leg. We assume that during the execution of a leg, the commuter receives information on which services have already visited the end node and which are yet to arrive. In other words, the customer "sees" the available successor legs of the current leg and may choose to (i) stay in the vehicle, (ii) transfer to another vehicle or (iii) get off the vehicle and start walking towards a nearby stop (or the destination). Our approach is to define an optimal policy specifying the actions that are executed in different situations in order to maximize the probability of reaching the destination before T.

In summary, the problem can be characterized as a dynamic and stochastic path finding problem. Such problems are often modeled as *Markov decision processes* (Psaraftis and Tsitsiklis, 1993; Polychronopoulos and Tsitsiklis, 1996), in which the *actions* of a decision maker at a given *state* are independent of all previous actions and states. We first present a conditional Markov model, in which the path history is included in each state by defining states as sequences of legs in the transport network. That is, the current state is determined by the path taken so far. This model is further approximated by means of history independent models, in which the current state is defined as the current leg.

This work is partially motivated by a demand-responsive transport (DRT) service currently being planned to operate in Helsinki. Helsinki Region Transport board has approved a plan under which the trial period of the service takes place from 2012 to 2014. Similarly as the current flexible service routes (Helsinki Region Traffic, 2010a), the new DRT service is designed to operate on a demand-responsive basis, that is, vehicle routes are modified according to the demand situation. The main difference to existing services is that no pre-order times for trips are required and the trips can be booked "on the fly" by means of an interactive user interface. This type of new service calls for a journey planner that is capable of communicating with flexible services as well as traditional public transport, thus combining the benefits of both transport modes.

In addition to public transport, a similar journey planning problem arises in freight transportation by for-hire carriers. For this purpose, we show how travel and overdue costs can be incorporated in the model.

The remainder of this document is organized as follows: The journey planning problem under uncertainty is formalized in Sections 2, 3 and an algorithm that generates an optimal policy for the conditional Markov decision process is presented in Section 4. In Section 5, we approximate the conditional solution by assuming history independence and compare the solutions by analysis. The solution methods are evaluated by numerical experiments in Section 6.

1.1. Related work

There is a vast literature devoted to the deterministic path-finding problem in a transit network. Zografos and Androutsopoulos (2008) classify the approaches into the following types of formulations: 1) the headway-based model, in which a constant headway for each transit line is assumed (Wong and Tong, 1998) and 2) the schedule-based model, which assumes a fixed route and timetable for each transit line.

Our approach stems from model 2, for which most existing solution approaches are based on label correcting, label setting or branch-and bound, see for example (Zografos and Androutsopoulos, 2008; Peng and Huang, 2000; Modesti and Siomachen, 1998; Huang and Peng, 2001, 2002; Horn, 2003; Tong and Richardson, 1984; Tong and Wong, 1999; Ziliaskopoulos and Wardell, 2000; Ziliaskopoulos and Mahmassani, 1993; Brub et al., 2006; Cooke and Halsey, 1966; Cai et al., 1997; Chabini, 1998; Kostreva and Wiecek, 1993; Hamacher et al., 2006; Androutsopoulos and Zografos, 2009). Heuristic solutions, that are useful when the fast solution of the problem is essential, are presented in (Bander and White, 1991) and (Tan et al., 2007).

In addition to the above-mentioned itinerary planning models, most of which are deterministic, our approach is closely related to the stochastic shortest path problem (SSPP). There are many different versions of the problem considered in the literature, each with a different meaning for the optimal path (Murthy and Sarkar, 1997). Early studies related to the problem defined the optimal path to be the one that maximizes the decision maker's expected utility. This objective is motivated by the Von Neumann-Morgenstern approach of preference judgments under uncertainty (Loui, 1983). Bard and Bennett (1991) present heuristic methods involving Monte-Carlo simulation to solve the SSPP with a general non-increasing utility function. An exact algorithm for the SSPP with a quadratic utility function is presented in (Mirchandani and Soroush, 1985).

Recent studied objectives for the stochastic shortest path problem include (i) the maximization of the probability that the length of the path does not exceed a threshold value or finding the path that maximizes the probability of arriving on time (Fan et al., 2005a; Nikolova et al., 2006b; Nie and Wu, 2009), (ii) finding the path with the greatest probability of being the shortest (Sigal et al., 1980; Kamburowski, 1985) and (iii) the minimization of expected cost, distance or travel time (Jaillet, 1992; Waller and Ziliaskopoulos, 2002; Fan et al., 2005b; Nikolova et al., 2006a; Thomas and White, 2007; Peer and Sharma, 2007).

Most studies related to the stochastic shortest path problem are considered *static*, since they call for the a priori selection of a fixed path optimizing an appropriate objective. In contrast, solutions to *dynamic and stochastic* shortest path problems[3] can be characterized as policies specifying the

[3]The dynamic and stochastic shortest path problem is also referred to as the *stochastic shortest path problem with recourse* (Waller and Ziliaskopoulos, 2002).

appropriate actions for each particular real-time scenario (Psaraftis and Tsitsiklis, 1993). The first version of the dynamic problem, in which the travel-time distributions of arcs are known and time-dependent, was presented in (Hall, 1986). A heuristic search algorithm for a similar problem is presented in (Bander and White III, 2002). The model is extended in (Fu and Rillet, 1998) to the case of continuous-time random arc costs. Miller-Hooks and Mahmassani (2000) provide an algorithm for finding the least expected cost path in a discrete-time dynamic problem.

Polychronopoulos and Tsitsiklis (1996) consider a dynamic shortest path problem in which arc costs are randomly distributed, and the arc costs are realized once the vehicle arrives at the arc. Iterative and adaptive algorithms for a similar problem are presented in (Cheung, 1998; Cheung and Muralidharan, 2000). Psaraftis and Tsitsiklis (1993) study a variation of the dynamic problem in which the distributions on arc travel times evolve over time according to a Markov process and the changes in the status of an arc are not observed until the vehicle reaches the arc. A genetic algorithm for dynamic re-routing in a similar problem setting is presented in (Davies and Lingras, 2003). Kim et al. (2005a) extend the model to include real-time information and present results regarding optimal departure times from a depot and optimal routing policies. A state-space reduction technique that significantly improves computation time for the problem is presented in (Kim et al., 2005b). Azaron and Kianfar (2003) extend the analytical results presented in (Psaraftis and Tsitsiklis, 1993) for the case where the states of the current arc and immediately adjacent arcs are known. Ferris and Ruszczynski (2000) model a problem in which arcs can fail and become unusable as an infinite-horizon Markov decision process and give an example of the behavior of an optimal policy. Again, using a Markov decision process, Thomas and White (2007) consider the problem of constructing a minimum expected total cost route from an origin to a destination that anticipates and then responds to changes in congestion.

Datar and Ranade (2000) examine a public transport model in which bus arrival times at each stop are exponentially and independently distributed and present an algorithm to generate travel plans for such conditions. Boyan and Mitzenmacher (2001) extend this approach by characterizing an optimal policy for traversing a bus network in which bus-arrival times are distributed according to a probability distribution satisfying the increasing failure rate property.

In addition to the shortest path problem, stochastic travel times have been incorporated in studies related to the traveling salesman problem (Kao, 1978) and its generalization, the vehicle routing problem. Kenyon and Morton (2003) study a vehicle routing problem with multiple vehicles and stochastic travel times, where the objective is to maximize the probability that the duration of each vehicle tour is less than a specified maximum time. Jula et al. (2006) consider a traveling salesman problem with time windows and stochastic travel times, in which the goal is to find the route with minimum expected cost. Problems with up to 80 customers are solved by means of the proposed dynamic programming solution. Russell and Urban (2007) present a similar problem with multiple vehicles in which an additional costs are incurred if the time windows are violated. By using a tabu search algorithm, problems involving 100 customers are solved.

In this work we examine a problem similar to the ones studied in (Datar and Ranade, 2000; Boyan and Mitzenmacher, 2001). The major difference is that we consider a more detailed version of the problem by assuming that each public transport service has a specific schedule, but the arrival times of the services at stops are defined as random variables. In addition, we consider the possibility of walking between adjacent stops, as in (Zografos and Androutsopoulos, 2008).

2. Model

Let $\mathcal{V}$ denote a set of *nodes* representing public transport stops in a specific area and let $\mathcal{K} \subset \mathbb{N}$ denote a set of public transport *services* operating in this area, indexed by natural numbers.

Each service $k \in \mathcal{K}$ follows a *route*, represented as a sequence of nodes $(v_1^k, \ldots, v_m^k)$ in $\mathcal{V}$. Note that it is possible for a service to visit the same node more than once. For example, if node v_i^k is included in the route twice, there exists an index $j \neq i$ for which $v_i^k = v_j^k$. Each service departs at node v_1^k at a specific time and proceeds to nodes $v_2^k, \ldots, v_m^k$ in the order determined by the route. Due to uncertainty in departure and travel times, we define the arrival times of services at nodes as real-valued random variables. Letting τ_j^k denote the *random arrival time* of service k at node v_j^k, a service k can be represented as a sequence of deterministic nodes and random arrival times $((v_1^k, \tau_1^k), \ldots, (v_m^k, \tau_m^k))$, see Figure 2. Any two arrival times τ_j^k and τ_l^h of distinct services k and h are assumed to be independent, whereas the arrival times τ_j^k and τ_l^k of a single service k at stops v_j^k and v_l^k are not necessarily independent (see Section 5.1). In this section, we examine the arrival times as arbitrarily distributed real random variables. In the numerical experiments discussed in Section 6, the arrival times are defined as gamma distributed random variables, similarly as in (Russell and Urban, 2007).

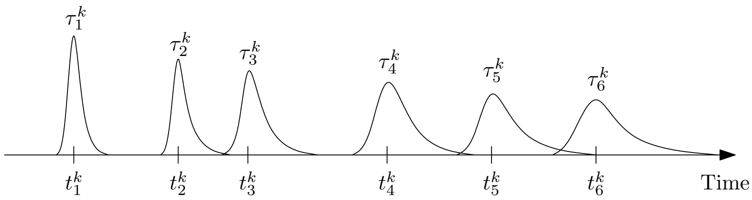

Figure 2: The schedule of a transport service. The real numbers t_j^k represent the scheduled arrival times of a transport service at six subsequent stops. The curves represent the distributions (gamma distributions in this example) of random variables τ_j^k that are used to model the actual arrival times of the service at stops v_j^k, $j \in \{1, \ldots, 6\}$.

2.1. Service legs

Each service $k \in \mathcal{K}$ can be decomposed as a set of scheduled *legs* between subsequent stops. That is, each leg has a start node, end node, start time and end time. By this decomposition, we can represent any path in the transport network as a sequence of legs indexed by natural numbers. For example, consider a transport network consisting of ten legs numbered from 1 to 10. Then, the sequences $(5, 2, 7)$ and $(5, 8, 3, 7)$ define two paths in the network involving two and three *transfers*, respectively. Each transfer between two legs has a specific probability of success. Transfers between subsequent legs of the same service are assumed to be successful with probability 1.

Formally, letting $\mathcal{L} \subset \mathbb{N}$ denote the set of indices of legs, a leg corresponding to index $i \in \mathcal{L}$ is defined by a pair of node-arrival time pairs and a service number $((v_i, \tau_i), (v_i', \tau_i'), k_i)$, where v_i is the *start node*, v_i' is the *end node*, τ_i is the *start time*, τ_i' is the *end time* and the *service number* k_i is the index of the service that executes leg i.

By using this definition, a service $((v_1^k, \tau_1^k), \ldots, (v_m^k, \tau_m^k))$ is represented by a set of legs $\{i_1^k, \ldots, i_{m-1}^k\} \subset \mathcal{L}$, where leg i_h^k is determined by

$$\left((v_{i_h^k}, \tau_{i_h^k}), (v'_{i_h^k}, \tau'_{i_h^k}), k_{i_h^k}\right) = \left((v_h^k, \tau_h^k), (v_{h+1}^k, \tau_{h+1}^k), k\right) \tag{1}$$

for $h \in \{1, \ldots, m-1\}$. Note that the end time $\tau'_{i^k_h}$ of leg i^k_h refers to the same specific random variable as the start time $\tau_{i^k_{h+1}}$ of leg i^k_{h+1}, that is, $\tau'_{i^k_h} = \tau_{i^k_{h+1}} = \tau^k_{h+1}$. To keep track which legs are subsequent legs of the same service, we define for each leg i^k_h the *set of immediate successors* $I_{i^k_h} = \{i^k_{h+1}\}$ for $h \in \{1, \ldots, m-1\}$.

2.2. *Walking legs*

After each leg traversed, a commuter may choose to continue the journey by foot. Similarly as the service legs defined above, each *walking leg* $i \in \mathcal{L}$ is determined by a start node v_i, end node v'_i, start time τ_i, end time τ'_i and service number k_i. For all walking legs, we choose $k_i < 0$ in order to have a distinction between walking legs and service legs.

Walking legs are added by associating with each service leg $((v_i, \tau_i), (v'_i, \tau'_i), k_i)$ a set $L_i \subset \mathcal{L}$ of walking legs beginning at v'_i and ending at a stop within a specific *maximum walking distance* $d_w^{\max}$ from v'_i, see Figure 3a. Such walking legs $j \in L_i$ are of the form $\left((v_j, \tau_j), (v'_j, \tau'_j), k_j\right) = \left((v'_i, \tau'_i), (v'_j, \tau'_j), -k_i\right)$, where the walking distance d_w satisfies $d_w(v'_i, v'_j) \leq d_w^{\max}$. Note that the end time τ'_i of service leg i and the start time τ_j of a walking leg $j \in L_i$ refer to the same specific random variable. For each service leg i, the walking legs are added to the set of immediate successors I_i of i, that is, $I_i \leftarrow I_i \cup L_i$.

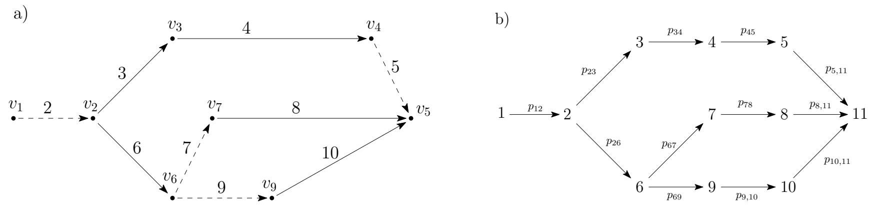

Figure 3: a) A sample transport network with eight stops and five transport services consisting of a single leg (solid arrows $3, 4, 6, 8, 10$). Adjacent stops are connected with walking legs (dashed arrows $2, 5, 7, 9$). b) A directed graph representing the relations of legs in Figure 3a. The origin and destination are represented by legs 1 and 11. The prior transfer probability from leg i to leg j is denoted by p_{ij}. Note that $p_{12} = p_{45} = p_{67} = p_{69} = 1$, since $2, 5, 7$ and 9 are walking legs.

The numerical experiments presented in Section 6 are restricted to the case in which there are no sequential walking legs, that is, no walking leg is followed by another walking leg. However, sequential walking legs can be added to the network iteratively: First, walking legs (indexed with service number $-k_i$) are added to the end of each each service leg i (step 1). Then, walking legs (indexed with service number k_j) are added to the end of each walking leg j (step 2). Step 2 is repeated until the desired amount of walking legs is obtained. The number of walking legs included in the model is thus controlled by two parameters: The maximum walking distance $d_w^{\max}$ and the maximum number of sequential walking legs.

2.3. *Transfer probability*

A transfer from leg i to leg j is possible only if leg i ends at the node from which leg j begins, that is, $v'_i = v_j$ (see Figure 3b). Letting $\mathcal{L}$ denote the set of legs and $S_i = \{j \in \mathcal{L} \mid v'_i = v_j\}$ denote the *successor set* of leg i, the *prior transfer probability* from leg i to leg j is defined by

$$p_{ij} = \begin{cases} P(\tau_i' \leq \tau_j), & \text{if } j \in S_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Note that if j is a walking leg beginning at the end of leg i or if i and j are successive legs of the same service, by definition we have $p_{ij} = 1$, since τ_i' and τ_j refer to the same random variable. The legs and prior transfer probabilities form a directed graph (Figure 3b).

More generally, since the start and end time of a leg are not necessarily independent, the transfer probability from i to j depends on the *path* taken to get to i. A path can be represented as an acyclic sequence of legs $(i_1, \ldots, i_m)$ satisfying $i_{h+1} \in S_{i_h}$ for all $h \in \{1, \ldots, m-1\}$, see Figure 4. The path is *successful*, if $\tau_{i_h}' \leq \tau_{i_{h+1}}$ for all $h \in \{1, \ldots, m-1\}$. Note that if i_{h+1} is an immediate successor of i_h, that is, $i_{h+1} \in I_{i_h}$, we have $\tau_{i_h}' = \tau_{i_{h+1}}$ since the start time of $i_{h+1} \in I_{i_h}$ refers to the same specific random variable as the end time of i_h.

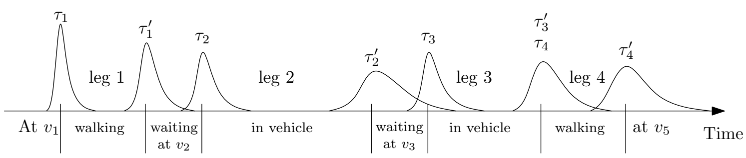

Figure 4: Deterministic and stochastic representations of a path. The ticks on the time axis represent the schedule of a deterministic itinerary from v_1 to v_5. The curves represent the distributions (gamma distributions in this example) of random variables τ_i, τ_i' that are used to model the start and end times of four legs that define the corresponding stochastic path. (1) A commuter starts walking at τ_1 from the origin v_1, and arrives at stop v_2 at τ_1'. (2) A transport service departs at v_2 at τ_2 and travels to stop v_3, arriving at τ_2'. (3) A transport service departs at v_3 at τ_3 and arrives at stop v_4 at τ_3'. (4) Immediately at $\tau_4 = \tau_3'$, the commuter continues by foot to the destination v_5, arriving at τ_4'. Note that the path is successful with probability $P(\tau_1' \leq \tau_2 \cap \tau_2' \leq \tau_3) = P(\tau_1' \leq \tau_2) \cdot P(\tau_2' \leq \tau_3 \mid \tau_1' \leq \tau_2)$.

3. Problem formulation

The dynamic journeying problem is defined as follows. Let $[0, T] \subset \mathbb{R}$ be the time horizon of the problem, let v_1 denote the origin node and v_d denote the destination node. The *origin leg* is defined by $((v_1, \tau_1), (v_1, \tau_1), 0)$, where $P(\tau_1 = 0) = 1$, and the *destination leg* is defined by $((v_d, \tau_d), (v_d, \tau_d), 0)$, where $P(\tau_d = T) = 1$. Although it is possible to reach the destination node before T, the start time of the destination leg equals T with probability 1. By this definition, we can represent the entire journey as legs, including the origin and the destination. If a journey ends at a successful transfer to the destination leg, we know that the commuter has reached the destination node before T or at T.

With no loss of generality, all legs for which the start or end time is outside the time horizon $[0, T]$ with probability 1 and all legs for which the start node is the destination node (except the destination leg) are excluded from the problem. The cropped set of legs is defined by

$$\mathcal{L}_{[0,T]} = \{i \in \mathcal{L} \mid P(0 \leq \tau_i \leq T) > 0 \text{ and } P(0 \leq \tau_i' \leq T) > 0\} \setminus \{i \in \mathcal{L} \setminus \{d\} \mid v_i = v_d\}.$$

For clarity, we define $n := |\mathcal{L}_{[0,T]}|$ and the legs in $\mathcal{L}_{[0,T]}$ are numbered from 1 to n, where the origin leg is indexed by 1 and the destination leg is indexed by n.

The problem is modeled as a finite-state Markov decision process $(S, A, P_\cdot(\cdot,\cdot), R_\cdot(\cdot,\cdot))$, where the parameters are defined as follows.

3.1. States

The set of *states* S consists of paths $(1, i_1, \ldots, i_m)$ of legs, where $i_h \in \{1, \ldots, n\}$ for $h \in \{0, \ldots, m\}$, beginning from the origin leg 1. The state (1) is referred to as the *origin state*. Each state $s = (i_0, i_1, \ldots, i_m) \in S$, where $i_0 = 1$, is associated with a set of successor states $S_s = \{(i_0, i_1, \ldots, i_m, j) \in S \mid j \in S_{i_m}\}$, where S_{i_m} is the successor set of leg i_m. Similarly, the set of immediate successors of state s is defined by $I_s = \{(i_0, i_1, \ldots, i_m, j) \in S \mid j \in I_{i_m}\}$, where I_{i_m} is the immediate successor set of leg i_m. Formally, the set of states is defined by

$$S = \{(i_0, \ldots, i_m) \mid i_0 = 1 \text{ and } i_{h+1} \in S_{i_h} \text{ for } h \in \{0, \ldots, m-1\}\}. \tag{3}$$

The above definition of a state includes travel history in the form of a path, see Figure 5. The history of performed actions is not taken into account in our model.

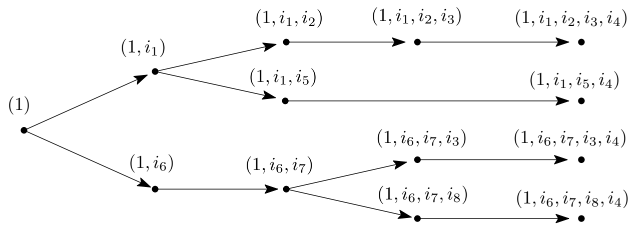

Figure 5: States of the markov decision process. A state is defined as a path consisting of legs, beginning from the origin leg 1. The states s in the figure are connected to successor states $s' \in S_s$ by arrows. Since the travel history is included in the definition of a state, the state space has a tree structure. In this example, the destination leg is denoted by i_4 and the set of destination states equals $\mathcal{D} = \{(1, i_1, i_2, i_3, i_4), (1, i_1, i_5, i_4), (1, i_6, i_7, i_3, i_4), (1, i_6, i_7, i_8, i_4)\}$.

The set of *destination states*, that is, the set of states $(1, \ldots, i_m) \in S$ for which the last leg i_m is the destination leg, is defined by $\mathcal{D} = \{(1, \ldots, i_m) \in S \mid i_m = n\}$. Since legs beginning from the destination node are not included in the problem, we have $S_s = I_s = \emptyset$ for all $s \in \mathcal{D}$.

The start and end times of state s are denoted by the random variables τ_s and τ'_s, respectively. The distributions of τ_s and τ'_s, where $s = (i_0, i_1, \ldots, i_m)$ and $i_0 = 1$, are defined by

$$f_{\tau_s}(x) = f_{\tau_{i_m}}(x \mid \cap_{h=0}^{m-1} \tau'_{i_h} \le \tau_{i_{h+1}}), \tag{4}$$

$$f_{\tau'_s}(x) = f_{\tau'_{i_m}}(x \mid \cap_{h=0}^{m-1} \tau'_{i_h} \le \tau_{i_{h+1}}), \tag{5}$$

for $x \in \mathbb{R}$. Note that if state s' is an immediate successor of state s, that is, $s' \in I_s$, the end time of state s and the start time of state s' refer to the same specific random variable $\tau'_s = \tau_{s'}$. In addition, if i_m is the first leg in the path executed by service k_{i_m}, that is, $k_{i_h} \neq k_{i_m}$ for all $h \in \{0, \ldots, m-1\}$, we have $f_{\tau_s}(x) = f_{\tau_{i_m}}(x)$ due to the fact that the arrival times of different services are assumed to be independent.

The service number of state $s = (i_0, i_1, \ldots, i_m)$ is defined by $k_s = k_{i_m}$.

3.2. Actions

The set of *actions* A consists of sets A_s of actions available at states $s \in S$. An action $a \in A_s$ is defined as a *preference order* of the successor states $s' \in S_s$, that is, a bijection $a : S_s \to \{1, \ldots, |S_s|\}$, where $a(s')$ denotes the ranking of the successor state $s' \in S_s$ in the preference order. The successor states of s ranked by the preference order a are denoted by $s^{a,a(s')}$. Given the the sorted successor states $s^{a,1}, \ldots, s^{a,|S_s|} \in S_s$ of s, the commuter transfers to state $s^{a,k}$ if (i) the transfer to $s^{a,g}$ is unsuccessful for $1 \le g < k$ and (ii) the transfer to $s^{a,k}$ is successful.

3.3. Transition probabilities

$P_a(s, s') = P(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability that action a in state s at step t will lead to state $s' \in S_s$ at step $t+1$. Given the preference order $s^{a,1}, \dots, s^{a,|S_s|} \in S_s$ defined by action $a \in A_s$, we have

$$P_a(s, s^{a,k}) = P\left(\tau'_s \leq \tau_{s^{a,k}} \cap \bigcap_{g=1}^{k-1} \tau'_s > \tau_{s^{a,g}}\right). \tag{6}$$

If the service numbers of all pairs of distinct successor states $s', s'' \in S_s$ of state s satisfy $k_{s'} \neq k_{s''}$, we have $P_a(s, s^{a,k}) = \int_{-\infty}^{\infty} f_{\tau'_s}(x) P\left(x \leq \tau_{s^{a,k}}\right) \prod_{g=1}^{k-1} P\left(x > \tau_{s^{a,g}}\right)\, dx$, since the start times of legs with different service numbers are independent. Note that since the end time τ'_s of state s and the start time of an immediate successor state $s' \in I_s$ refer to the same random variable, that is, $\tau'_s = \tau_{s'}$ for all $s' \in I_s$, the transfer probability (6) satisfies

$$P_a(s, s^{a,k}) = \begin{cases} 0, & \text{if } \{s^{a,1}, \dots, s^{a,k-1}\} \cap I_s \neq \emptyset\,, \\ P\left(\bigcap_{g=1}^{k-1} \tau'_s > \tau_{s^{a,g}}\right), & \text{if } \{s^{a,1}, \dots, s^{a,k-1}\} \cap I_s = \emptyset \text{ and } s^{a,k} \in I_s\,, \\ P\left(\tau'_s \leq \tau_{s^{a,k}} \cap \bigcap_{g=1}^{k-1} \tau'_s > \tau_{s^{a,g}}\right), & \text{otherwise.} \end{cases} \tag{7}$$

For example, let $s^{a,1}, s^{a,2}, s^{a,3}$ denote three successor states of state s, sorted in the preference order defined by action a. If $s^{a,2}$ is an immediate successor of s, that is, $s^{a,2} \in I_s$, and $s^{a,1} \notin I_s$, we have $P_a(s, s^{a,1}) = P\left(\tau'_s \leq \tau_{s^{a,1}}\right)$, $P_a(s, s^{a,2}) = P\left(\tau'_s > \tau_{s^{a,1}}\right)$ and $P_a(s, s^{a,3}) = 0$.

3.4. Rewards

$R_a(s, s')$ is the expected immediate reward received after transition from state $s \in S$ to state $s' \in S$ with transition probability $P_a(s, s')$. Since the objective is to arrive at a destination state $s \in \mathcal{D}$, we define $R_a(s, s')$ by

$$R_a(s, s') = \begin{cases} 1, & \text{if } s' \in \mathcal{D}\,, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Note that $R_a(s, s')$ is independent of the action a. Thus, for the remainder of this document, we will use the notation $R(s, s') = R_a(s, s')$.

While the main focus of this paper is on on-time arrival probability, travel and overdue costs can be included with minimial effort as follows. Instead of maximizing the probability of reaching the destination leg, the goal is to maximize the expected profit gained by arriving at the destination node.

Let u denote the revenue (or utility) received upon arrival at the destination node and $c(s)$ denote a real valued positive function representing the cost of state s. For example, $c((i_0, \dots, i_m))$ could be defined as the sum of the *leg costs* c_i, that is, $c((i_0, \dots, i_m)) = \sum_{h=0}^{m-1} c_{i_h}$. Let C_o denote the *overdue cost*, that is, the cost of arriving at the destination node after T. By introducing an additional *late destination leg* $((v_d, \tau_l), (v_d, \tau_l), 0)$, where $P(\tau_l = \infty) = 1$, and the set of *late destination states* $\mathcal{D}'$, for which the last leg is the late destination leg, the reward function is written in the general form

$$R'(s, s') = \begin{cases} u - c(s'), & \text{if } s' \in \mathcal{D}\,, \\ u - c(s') - C_o, & \text{if } s' \notin \mathcal{D} \text{ and } s' \in \mathcal{D}'\,, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

4. Problem solution

The solutions to Markov decision processes are characterized as policies, that is, functions π that specify the action $a(s)$ that the commuter chooses when in state s. The goal is to find a policy π that maximizes the expected reward. Generally, the calculation of an optimal policy requires two arrays indexed by state: value V, which contains real values, and policy π which contains actions. In our problem, $V(s)$ corresponds to the probability of reaching a destination state $s' \in \mathcal{D}$ from state s by following a policy that maximizes the probability of reaching a destination state. Note that the value $V((1))$ of the origin state (1) is strictly positive if and only if there exists a state $s \in S$ for which $S_s \cap \mathcal{D} \neq \emptyset$ and $P(\tau_s \leq T) > 0$.

Similarly as in (Bellman, 1957), the value $V(s)$ is defined by

$$V(s) := \max_{a \in A_s} \left\{ \sum_{s' \in S_s} P_a(s, s') \left(R(s, s') + V(s') \right) \right\} \tag{10}$$

for all $s \in S$. Note that since $V(s)$ represents a probability, we have $0 \leq V(s) \leq 1$ for all $s \in S$. In addition, $V(s) = 0$ for all $s \in \mathcal{D}$, since $S_s = \emptyset$ for all $s \in \mathcal{D}$. Since $R(s, s') = 0$ for all $s' \notin \mathcal{D}$ and $R(s, s') = 1$ for all $s' \in \mathcal{D}$, we have $0 \leq R(s, s') + V(s') \leq 1$ for all $s \in S$.

An optimal policy is characterized as follows: When at state s, the available transfer options $X \subset S_s$ to successor states are revealed. The commuter transfers to a state $s' \in X$ for which $R(s, s') + V(s')$ is maximized. Formally, an optimal action a at state s is determined by the following theorem.

Theorem 1. *Let $s \in S$ be a state. An action $a \in A_s$ is optimal satisfying Equation* (10)*, if*

$$R(s, s^{a,1}) + V(s^{a,1}) \geq R(s, s^{a,2}) + V(s^{a,2}) \geq \ldots \geq R(s, s^{a,|S_s|}) + V(s^{a,|S_s|}), \tag{11}$$

where the successor states ranked by action a are denoted by $s^{a,a(s')}$ for all $s' \in S_s$.

Proof. Let $S_s = \{s^1, \ldots, s^{|S_s|}\}$ denote the successor set of state s. Given an arbitrary realization $\{\tau'_s = t'_s, \tau_{s^1} = t_{s^1}, \ldots, \tau_{s^{|S_s|}} = t_{s^{|S_s|}}\}$ of the end time of state s and the start times of states $s^k \in S_s$, the conditional value of s satisfies

$$V(s \mid t'_s, t_{s^1}, \ldots, t_{s^{|S_s|}}) = \max_{\{k \in \{1,\ldots,|S_s|\} \mid t'_s \leq t_{s^k}\}} \left\{ R(s, s^k) + V(s^k) \right\}.$$

Note that $t_{s^k} = t'_s$ for all $s^k \in I_s$. Let a be an action satisfying Equation (11). Since the conditional transition probability is given by

$$P_a(s, s^{a,k} \mid t'_s, t_{s^{a,1}}, \ldots, t_{s^{a,|S_s|}}) = \begin{cases} 1, & \text{if } t'_s \leq t_{s^{a,k}} \text{ and } t'_s > t_{s^{a,g}} \text{ for } 1 \leq g < k, \\ 0, & \text{otherwise,} \end{cases}$$

we have

$$V(s \mid t'_s, t_{s^{a,1}}, \ldots, t_{s^{a,|S_s|}}) = \sum_{k=1}^{|S_s|} P_a(s, s^{a,k} \mid t'_s, t_{s^{a,1}}, \ldots, t_{s^{a,|S_s|}}) \left(R(s, s^{a,k}) + V(s^{a,k}) \right).$$

Integrating over $\mathbb{R}^{|S_s|+1}$ yields

$$V(s) = \sum_{k=1}^{|S_s|} P_a(s, s^{a,k}) \left(R(s, s^{a,k}) + V(s^{a,k})\right). \tag{12}$$

□

Theorem 1 gives an optimal action for a state s, given that the values $V(s')$ of its successor states are known. In the following, we present an algorithm for calculating the values for all states.

4.1. Backward induction algorithm

The values $V(s)$ of states can be determined by means of backward induction, as shown in Algorithm 1. By executing Prob((1)), the program recursively calculates values and optimal actions for all states s that are reachable from the origin state (1).

Initially, we only know the values of destination states, that is, $V(s) = 0$ for all $s \in \mathcal{D}$. Thus, the first states for which the value can be calculated are the ones that precede a destination state. The algorithm then proceeds backwards until the value $V((1))$ of the origin state is calculated.

Algorithm 1: A recursive function Prob(s) for calculating the value $V(s)$ for state s.

forall $s' \in S_s$ *(successor set of s)* **do**
    $V(s') \leftarrow \text{Prob}(s')$;
end
Determine an optimal action a by sorting the states $s' \in S_s$ in descending order of $R(s, s') + V(s')$;
$V(s) \leftarrow \sum_{k=1}^{|S_s|} P_a(s, s^{a,k}) \left(R(s, s^{a,k}) + V(s^{a,k})\right)$ ($V(s) = 0$ for destination states);
Return $V(s)$;

5. Analysis

In the following examination, we present theoretical results related to optimal policies described in the previous section. For clarity, walking legs are not considered in this section. However, since walking legs and service legs are defined similarly apart from the service number, the results are extended to handle walking legs with minimal effort.

We first prove an intuitive result: The probability of reaching the destination from a given state s by following an optimal policy is always greater than or equal to the probability of success of any predetermined path from s to the destination.

Lemma 2. *Let $(i_0, i_1, \ldots, i_{d-1}, i_d)$, where $i_0 = 1$, be a destination state. Let p_{s_h} denote the probability of success of the path $(i_h, \ldots, i_d)$ from leg i_h to the destination leg i_d, given that the commuter is at state $s_h = (i_0, i_1, \ldots, i_{h-1}, i_h)$. Then, $V(s_h)$ defined by Equation* (12) *satisfies $V(s_h) \geq p_{s_h}$ for all $h \in \{0, \ldots, d-1\}$.*

Proof. For clarity, we will prove the case in which the service numbers of distinct successor states $s', s'' \in S_s$ satisfy $k_{s'} \neq k_{s''}$. Clearly, $p(s_h)$ satisfies $p(s_h) = P(\tau'_{s_h} \leq \tau_{s_{h+1}})p(s_{h+1})$ for all $h \in$

$\{0, \ldots, d-1\}$. Note that for an optimal action a, we can choose $s_{d-1}^{a,1} = s_d$ since $R(s_{d-1}, s_d) + V(s_d) = 1$. Thus,

$$V(s_{d-1}) = \sum_{k=1}^{|S_{s_{d-1}}|} P_a(s_{d-1}, s_{d-1}^{a,k}) \left(R(s_{d-1}, s_{d-1}^{a,k}) + V(s_{d-1}^{a,k}) \right) \geq P_a(s_{d-1}, s_{d-1}^{a,1}) \left(R(s_{d-1}, s_{d-1}^{a,1}) + V(s_{d-1}^{a,1}) \right)$$

$$= P_a(s_{d-1}, s_d) \overbrace{(R(s_{d-1}, s_d) + V(s_d))}^{=1} = P(\tau'_{s_{d-1}} \leq \tau_{s_d}) = p(s_{d-1}).$$

The proof is completed by showing that $V(s_{h+1}) \geq p(s_{h+1})$ implies $V(s_h) \geq p(s_h)$ for all $h \in \{0, \ldots, d-2\}$.

Let $s_h \in \{s_0, s_1, \ldots, s_{d-2}\}$ be an arbitrarily chosen state and assume that $V(s_{h+1}) \geq p(s_{h+1})$ holds. Let $a(s_{h+1})$ denote the ranking of s_{h+1} in the ordered successor states of s_h, that is, $s_h^{a,a(s_{h+1})} = s_{h+1}$. Then, $V(s_h^{a,k}) \geq V(s_{h+1})$ for all $k \in \{1, \ldots, a(s_{h+1})\}$ and

$$V(s_h) \geq \sum_{k=1}^{a(s_{h+1})} V(s_h^{a,k}) \int_{-\infty}^{\infty} f_{\tau'_{s_h}}(x) P(x \leq \tau_{s_h^{a,k}}) \prod_{g=1}^{k-1} P(x > \tau_{s_h^{a,g}})\, dx$$

$$\geq V(s_{h+1}) \sum_{k=1}^{a(s_{h+1})} \int_{-\infty}^{\infty} f_{\tau'_{s_h}}(x) P(x \leq \tau_{s_h^{a,k}}) \prod_{g=1}^{k-1} P(x > \tau_{s_h^{a,g}})\, dx$$

$$= V(s_{h+1}) \int_{-\infty}^{\infty} f_{\tau'_{s_h}}(x) \left(1 - P(x > \tau_{s_h^{a,a(s_{h+1})}}) \overbrace{\prod_{g=1}^{a(s_{h+1})-1} P(x > \tau_{s_h^{a,g}})}^{\leq 1} \right) dx$$

$$\geq V(s_{h+1})(1 - P(\tau'_{s_h} > \tau_{s_{h+1}})) = P(\tau'_{s_h} \leq \tau_{s_{h+1}}) V(s_{h+1}) \geq P(\tau'_{s_h} \leq \tau_{s_{h+1}}) p(s_{h+1}) = p(s_h).$$

□

Note that Lemma 2 is true for any path that maximizes the probability of success for reaching the destination. By using the above result, we show that the ratio of probabilities $V(s_h)/p(s_h)$ is an increasing function of the distance from the destination.

Theorem 3. *Using the notation of Lemma 2, we have $V(s_h)/p(s_h) \geq V(s_{h+1})/p(s_{h+1}) \geq 1$ for all $h \in \{0, \ldots, d-1\}$.*

Proof. Given that the commuter is at state $s_h = (i_0, \ldots, i_h)$, the probability of success of the path $(i_h, \ldots, i_d)$ satisfies $p(s_h) = P(\tau'_{s_h} \leq \tau_{s_{h+1}}) p(s_{h+1})$. Similarly as in the proof of Lemma 2, we see that $V(s_h) \geq P(\tau'_{s_h} \leq \tau_{s_{h+1}}) V(s_{h+1})$ and thus

$$\frac{V(s_h)}{p(s_h)} \geq \frac{P(\tau'_{s_h} \leq \tau_{s_{h+1}}) V(s_{h+1})}{P(\tau'_{s_h} \leq \tau_{s_{h+1}}) p(s_{h+1})} = \frac{V(s_{h+1})}{p(s_{h+1})} \geq 1.$$

□

Theorem 3 establishes a fundamental difference between dynamic and static journey planning: The importance of being able to reconsider the remaining path is emphasized when the number of transfers is increased.

5.1. *Leg duration models*

Thus far we have considered the general case where the start and end times of legs are defined as arbitrary real-valued random variables. For a more specific analysis, we present two alternative ways of modeling the durations of legs.

5.1.1. *Independent durations*

Perhaps the most intuitive way of defining the start and end times of a leg of a public transport service is to assume that the *duration* of each leg i is a real-valued random variable D_i with a strictly positive distribution and that the durations of legs are independent. Given a service consisting of legs $i_1, \ldots, i_r$ and the start time τ_{i_1}, the end time of leg i_h and the start time of leg i_{h+1} satisfy $\tau'_{i_h} = \tau_{i_{h+1}} = \tau_{i_1} + \sum_{h=1}^{r} D_{i_h}$. Note that in this model the uncertainty of start and end times are greater at the end of the route (see Figure 2). In addition, the conditional transfer probabilities are smaller than the prior transfer probabilities: For example, let us consider a transfer $j \to l$ with prior transfer probability $p_{jl} = P(\tau'_j \leq \tau_l) = P(\tau_j + D_j \leq \tau_l)$. Given that the commuter has already successfully transferred from i to j, where $P(\tau'_i \leq \tau_j) < 1$, the transfer $j \to l$ is generally less likely to be successful, that is, $P(\tau_j + D_j \leq \tau_l \mid \tau'_i \leq \tau_j) \leq P(\tau_j + D_j \leq \tau_l)$. In this model, the start and end time distributions of states are characterized by the following theorem.

Theorem 4. *Let $s = (i_0, \ldots, i_m) \in S, i_0 = 1$ be a state for which all legs are executed by different services, that is, $g \neq h \Rightarrow k_{i_g} \neq k_{i_h}$ for all $g, h \in \{0, \ldots, m\}$. Then, the density function $f_{\tau_s}(x)$ of the start time τ_s of state s is given by $f_{\tau_s}(x) = f_{\tau_{i_m}}(x)$ and the cumulative distribution function $F_{\tau_s}(x)$ of the end time τ'_s of state s is given by*

$$F_{\tau'_s}(x) = \int_{-\infty}^{\infty} f_{\tau_{i_0}+D_{i_0}}(x'_0) \int_{x'_0}^{\infty} f_{\tau_{i_1}}(x_1) \int_{x_1}^{\infty} f_{x_1+D_{i_1}}(x'_1) \cdots \int_{x'_{m-1}}^{\infty} f_{\tau_{i_m}}(x_m) \int_{x_m}^{x} f_{x_m+D_{i_m}}(x'_m) dx'_m dx_m \ldots dx'_0. \tag{13}$$

Proof. Since the start and end times of legs with different service numbers are independent, by definition (4) we get $f_{\tau_s}(x) = f_{\tau_{i_m}}(x \mid \cap_{h=0}^{m-1} \tau'_{i_h} \leq \tau_{i_{h+1}}) = f_{\tau_{i_m}}(x)$.

The cumlative distribution function of τ'_s is defined by $F_{\tau'_s}(x) = P(\tau'_s \leq x)$. Let us use the notation $s_h = (i_0, \ldots, s_h)$ for all $h \in \{0, \ldots, m\}$. Clearly, the density function of τ'_{s_0} is given by $f_{\tau'_{s_0}}(x) = f_{\tau_{i_0}+D_{i_0}}(x)$. Furthermore,

$$F_{\tau_{s_{h+1}}}(x) = \int_{-\infty}^{\infty} f_{\tau'_{s_h}}(x'_h) \int_{x'_h}^{x} f_{\tau_{i_{h+1}}}(x_{h+1}) dx_{h+1} dx'_h \quad \text{and}$$
$$F_{\tau'_{s_{h+1}}}(x) = \int_{-\infty}^{\infty} f_{\tau_{s_{h+1}}}(x_{h+1}) \int_{x_{h+1}}^{x} f_{x_{h+1}+D_{i_{h+1}}}(x'_{h+1}) dx'_{h+1} dx_{h+1}.$$

Since $f_{\tau_{s_{h+1}}}(x_{h+1}) = \frac{d}{dx} F_{\tau_{s_{h+1}}}(x)$, combining the equations yields a recursive equation

$$F_{\tau'_{s_{h+1}}}(x) = \int_{-\infty}^{\infty} f_{\tau'_{s_h}}(x'_h) \int_{x'_h}^{\infty} f_{\tau_{i_{h+1}}}(x_{h+1}) \int_{x_{h+1}}^{x} f_{x_{h+1}+D_{i_{h+1}}}(x'_{h+1}) dx'_{h+1} dx_{h+1} dx'_h.$$

□

Theorem 4 gives us a formula for calculating the end time distribution of a state defined as a sequence of legs, given that the legs are executed by different services. For a state $s = (i_0, \ldots, i_m)$ for which some subsequent legs $i_g, \ldots, i_h$ are immediate successors, equation (13) can be used by joining these legs to produce a single leg with duration $\sum_{k=g}^{h} D_{i_k}$.

In practice, the start and end time distributions of states can be determined by means of random sampling, as discussed in Section 5.2.

5.1.2. Independent start and end times

In some cases it might be reasonable to think that the driver of a transport service is capable of adapting to the situation: If the service is behind schedule, the driver makes an effort to reach the remaining stops on the route in time by increasing the pace. That is, the durations of subsequent legs of a service are not necessarily independent. This scenario can be approximated by assuming that the start and end times of all legs are independent random variables. In this case, the transition probabilities between legs are conditionally independent of all previous states and actions.

Since the travel history is ignored, the states are definded as legs, and the state space is defined by

$$S = \mathcal{L}_{[0,T]} = \{1, \ldots, n\}, \tag{14}$$

where state 1 denotes the orgin leg and state n denotes the destination leg, see Figure 6. Furthermore, the start and end times of states coincide with the unconditional start and end times of legs, in contrast to Equations (4) and (5), where the start and end times are defined as conditional random variables.

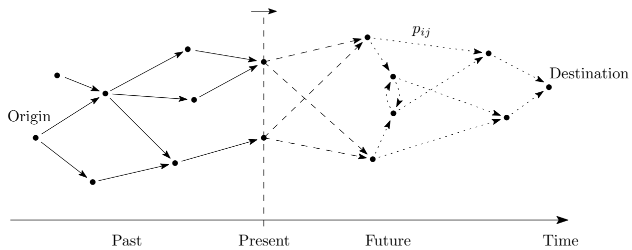

Figure 6: A timeline of the history independent markov decision process. The points denote states (=legs) and the arrows between the points denote transfers between states. The solid arrows denote successful *past transfers*, which form an acyclic graph. The dashed arrows denote *present transfers*, which are realized at the present time. The dotted arrows denote *future transfers* $i \to j$, each of which is successful with prior probability p_{ij}. Note that there may exist states i, j for which both $p_{ij} > 0$ and $p_{ji} > 0$. Over time, future transfers become present transfers and present transfers become past transfers.

We suggest that the history independent model (14) can be used to approximate to the conditional case (3). The following theorem characterizes the relation between the two models.

Theorem 5. *Let i, j and k denote three legs and let D_j denote the random duration of leg j. Given that the transfer $i \to j$ is successful and assuming that the durations of legs are independent, the conditional probability $P(\tau'_j \leq \tau_k \mid \tau'_i \leq \tau_j)$ of a successful transfer $j \to k$ satisfies*

$$P(\tau'_j \leq \tau_k) \geq P(\tau'_j \leq \tau_k \mid \tau'_i \leq \tau_j) \geq \frac{P(\tau'_i \leq \inf \operatorname{supp} f_{\tau_j})}{P(\tau'_i \leq \tau_j)} P(\tau'_j \leq \tau_k),$$

where $\inf \operatorname{supp} f_{\tau_j}$ *denotes the greatest real number* t *for which* $f_{\tau_j}(t) = 0$ *and* $t \leq x$ *for all* x *satisfying* $f_{\tau_j}(x) > 0$.

Proof. Defining $t = \inf \operatorname{supp} f_{\tau_j}$, the probability that both transfers $i \to j$ and $j \to k$ are successful satisfies

$$
\begin{aligned}
P(\tau_j' \leq \tau_k \cap \tau_i' \leq \tau_j) &= \int_{-\infty}^{\infty} f_{\tau_i'}(x_i') \int_{x_i'}^{\infty} f_{\tau_j}(x_j) P(x_j + D_j \leq \tau_k)\, dx_j\, dx_i' \\
&\geq \int_{-\infty}^{t} f_{\tau_i'}(x_i') \int_{t}^{\infty} f_{\tau_j}(x_j) P(x_j + D_j \leq \tau_k)\, dx_j\, dx_i' = P(\tau_i' \leq t) P(\tau_j' \leq \tau_k).
\end{aligned}
$$

□

Theorem 5 states that if the distributions of τ_i' and τ_j are relatively far apart, the conditional probability $P(\tau_j' \leq \tau_k \mid \tau_i' \leq \tau_j)$ of a successful transfer from j to k is close to the unconditional probability $P(\tau_j' \leq \tau_k)$. In this case, the two leg duration models discussed above converge and the independent durations model can be approximated by assuming independence of start and end times.

5.2. Algorithms for the independent durations model

For the independent leg durations model (Section 5.1.1), the start and end time distributions can be calculated by means of random sampling as follows. For each leg $i \in \{1, \ldots, n\}$, we calculate $R \in \mathbb{N}$ realizations for the start and end times, that is, sets $T_i = \{t_i(1), \ldots, t_i(R)\}$ and $T_i' = \{t_i'(1), \ldots, t_i'(R)\}$, which approximate the distributions of τ_i and τ_i', respectively. The random sampling algorithm method is described in Algorithm 2.

Algorithm 2: A recursive function Sample(i, t, q) for calculating the start and end time distributions of legs. The function Sample(i, t_i, q) is called for all $q \in \{1, \ldots, R\}$ and for all initial legs i of services (see Equation (1)), where t_i is a random number from the distribution τ_i.

```
t_i(q) ← t;
Choose a random number d according to distribution D_i;
t'_i(q) ← t + d;
forall i' ∈ I_i (immediate successor set of i) do
    Sample(i', t'_i(q), q);
end
```

Given the random samples of the start and end times of legs, the approximate start and end time distributions $T_s = \{t_s(1), \ldots, t_s(Q)\}$ and $T_s' = \{t_s'(1), \ldots, t_s'(Q)\}$ of states $s \in S$ are determined by repeating the procedure described in Algorithm 3 for all $q \in \{1, \ldots, R\}$.

5.3. Algorithms for the history independent model

Equation (14) suggests an approximation for the conditional markov decision process (3) by assuming history independence. Since no information on already visited states is considered, the history independent probabilities can be computed more efficiently than the conditional values given by Algorithm 1.

In particular, since the state is defined by the current leg, that is, $S = \{1, \ldots, n\}$, the number of states is in most cases significantly smaller than in the history dependent model. Letting $V_H(s)$

Algorithm 3: A recursive function Sample2(s, q) for calculating the start and end time distributions of states. The function Sample2$((1), q)$ is called for all $q \in \{1, \ldots, R\}$, where (1) is the origin state.

$i \leftarrow$ last leg of s;
$T_s \leftarrow T_s \cup \{t_i(q)\}$;
$T'_s \leftarrow T'_s \cup \{t'_i(q)\}$;
forall $s' \in S_s$ *(successor set of s)* **do**
  $i' \leftarrow$ last leg of s';
  if $t'_i(q) \leq t_{i'}(q)$ **then**
    Sample2(s', q);
  end
end

Algorithm 4: A recursive function Prob(s) for calculating an optimal policy for the history independent model. Initially, set $V_H(s) \leftarrow -1$ for all $s \in \{1, \ldots, n-1\}$ and $V(n) \leftarrow 0$.

if $V_H(s) \geq 0$ *($V_H(s)$ has already been determined)* **then**
  Return $V_H(s)$;
end
forall $s' \in S_s$ *(successor set of s)* **do**
  $V_H(s') \leftarrow$ Prob(s');
end
Determine an optimal action a by sorting the states $s' \in S_s$ in descending order of $R(s, s') + V_H(s')$;
$V_H(s) \leftarrow \sum_{k=1}^{|S_s|} P_a(s, s^{a,k}) \left(R(s, s^{a,k}) + V_H(s^{a,k})\right)$;
Return $V_H(s)$;

denote the history independent value of state s, we may initially define $V_H(s) = -1$ for all $s \in \{1, \ldots, n-1\}$ and $V_H(n) = 0$ in order to keep track of which states have already been visited by the algorithm. The history independent version of Algorithm 1 is presented in Algorithm 4.

While the recursive solution presented in Algorithm 4 produces the value for all states that can be reached from a given initial state s, the values can be efficiently computed for all states simultaneously by using a power method as follows.

Let us first write Equation (12) in matrix form. Let $s^{a,1}, \ldots, s^{a,|S_s|} \in S_s$ denote the ordered successor states of s and let J denote a $(n \times n)$-matrix defined by

$$J_{ss'} = \begin{cases} 1, & \text{if } s = s' = n, \\ P_a(s, s'), & \text{if } s \neq s' \text{ and } s' \in S_s, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Element $J_{ss'}$ represents the probability that state s' succeeds state s given that the commuter follows a policy determined by action a.

Letting $V_H = (V_H(1), \ldots, V_H(n-1), 1)^T$ denote the *value vector*, Equation (12) can be written in the form $V_H = JV_H$. Note that the last element $V_H(n)$ is replaced by 1 due to the fact that $(V_H(n) + R(s, n)) = 1$ for all $s \in S$.

In other words, V_H is the dominant eigenvector of J, that is, the eigenvector corresponding to eigenvalue 1. However, the calculation of V_H is not straightforward since the elements $J_{ss'}$ depend on the descending order of the elements $V_H(s^{a,k})$.

We propose a variation of the power method (Meyer, 2000) for determining J as well as the dominant eigenvector V_H, see Algorithm 5. The main idea is that the vector V_H is consecutively updated by multiplying matrix J by V_H. After each update, the successor states s' of each state s are sorted in descending order of value $V_H(s')$.

Algorithm 5: A variation of the power method for calculating J and the dominant eigenvector V_H.

```
Set V_H = (1, ..., 1);
repeat
    forall s ∈ {1, ..., n} do
        Determine action a by sorting the states s' ∈ S_s in descending order of R(s, s') + V_H(s');
        forall s^{a,k} ∈ S_s do
            J_{s s^{a,k}} ← P_a(s, s^{a,k});
        end
    end
    V_H ← J V_H;
until convergence ;
```

5.4. Unconditional transfers

The history independent model defined by Equation (14) can be further approximated by assuming that successful transfers from a state s to its successor states are independent events, which is not generally true. However, if the spreads of the time distributions are relatively small, we obtain a reasonable approximation by assuming independence of successful transfers. From a practical viewpoint, since we are particularly interested in the choice for the commuter in each situation, the values are not as interesting as the *descending order* of the values, that is, the action at each state. We suggest that even if approximate models are used, the actions of the commuter are in most cases similar as with more accurate models since the descending order of the probabilities is preserved.

Since transfers between legs are assumed to be independent events, the transition probability function (6) satisfies

$$P_a(s, s^{a,k}) = P\left(\tau'_s \le \tau_{s^{a,k}} \cap \bigcap_{g=1}^{k-1} \tau'_s > \tau_{s^{a,g}}\right) = \sum_{k=1}^{h} p_{ss^{a,k}} \prod_{g=1}^{k-1} (1 - p_{ss^{a,g}}), \tag{16}$$

where $p_{ss^{a,k}}$ denotes the prior transfer probability from s to $s^{a,k}$, as defined in Equation (2). Since the state space is equal to the state space of the history independent model, Algorithms 4 and 5 can be used to calculate the unconditional values .

Theorem 6. *The history independent value $V_H(s)$ defined by model* (14) *and the unconditional value $V_U(s)$ defined by* (16) *satisfy $V_H(s) \le V_U(s)$ for all $s \in \{1, \ldots, n\}$.*

Proof. Letting a denote an optimal action, by Equation (12) we get

$$\begin{aligned}
V_H(s) &= \sum_{k=1}^{h} (R(s, s^{a,k}) + V_H(s^{a,k})) \int_{-\infty}^{\infty} f_{\tau'_s}(x) P(x \le \tau_{s^{a,k}}) \prod_{g=1}^{k-1} P(x > \tau_{s^{a,g}})\, dx \\
&\le \sum_{k=1}^{h} (R(s, s^{a,k}) + V_H(s^{a,k})) \int_{-\infty}^{\infty} f_{\tau'_s}(x) P(x \le \tau_{s^{a,k}})\, dx \cdot \prod_{g=1}^{k-1} \int_{-\infty}^{\infty} f_{\tau'_s}(x) P(x > \tau_{s^{a,g}})\, dx \\
&= \sum_{k=1}^{h} (R(s, s^{a,k}) + V_H(s^{a,k})) P(\tau'_s \le \tau_{s^{a,k}}) \prod_{g=1}^{k-1} P(\tau'_s > \tau_{s^{a,g}}) \\
&= \sum_{k=1}^{h} (R(s, s^{a,k}) + V_H(s^{a,k})) p_{ss^{a,k}} \prod_{g=1}^{k-1} (1 - p_{ss^{a,g}}).
\end{aligned}$$

Noting that $V_H(n) = V_U(n) = 0$ for the destination state n, the proof is completed by backward induction. □

Note in particular that if the transfer probabilities satisfy $P(\tau'_s \leq \tau_{s^{a,k}}) \in \{0, 1\}$ for all successor states $s^{a,k} \in S_s$, we have $V_H(s) = V_U(s) = V_U(s^{a,k^*})$, where k^* is the smallest index for which $P(\tau'_s \leq \tau_{s^{a,k}}) = 1$. In this case, the history independent and unconditional coincide. In other words, the approximation given by (16) is seen to be most accurate when the variances of the arrival times are small, which means that the transfer probabilities are likely to be close to zero or one.

Furthermore, if each state has a single successor state, the history independent and unconditional models yield the same result. We suggest that the accuracy of the unconditional model is greatest when the average number of successor states is small.

5.5. Summary of models

In summary, the conditional (3), history independent (14) and unconditional (16) models described above can be characterized by means of two types of conditionality: (i) *History dependence* refers to the fact that the executed part of a journey affects the probability of success of the remaining part of the journey. (ii) *Transfer conditionality* refers to the fact that the success of different transfers at a given stop are not independent events.

The conditional model takes into account both types of conditionality whereas in the unconditional model, neither of the conditionalities are considered. The relations between the conditionalities and the three models are presented in the following table.

Type of conditionality \ Model	Conditional model (3)	History independent model (14)	Unconditional model (16)
History dependence	Yes	No	No
Transfer conditionality	Yes	Yes	No

6. Numerical experiments

In the following, we present computational results obtained by the conditional model (3), history independent model (14) and the unconditional model (16). In the conditional model, the durations of legs are assumed to be independent as described in Section 5.1.1 and in the history independent and unconditional models, the start and end times of legs are assumed to be independent, as described in Section 5.1.2.

First, the computational performance of the different models is compared in randomized and real-life instances (Section 6.1). Then, we study the probability of success of journeys in regular networks as a function of transfer margin and the number of transfer options at each stop, as well as the number of transfers (Section 6.2).

In all experiments, the travel times between stops are defined as shifted gamma distributed random variables similarly as in (Russell and Urban, 2007; Chiang et al., 1980). More precisely, given the average travel time t_{uv} between stops u and v, we define the travel time τ_{uv} as a random variable $\tau_{uv} \sim \text{Gamma}(\alpha t_{uv}, \beta, \delta t_{uv})$, where the $\text{Gamma}(\alpha, \beta, \delta)$ distribution is defined by the probability density function

$$f(x) = \frac{(x-\delta)^{\alpha-1} e^{-(x-\delta)/\beta}}{\beta^\alpha \Gamma(\alpha)} \quad \text{for } x > \delta \geq 0. \tag{17}$$

The mean of a Gamma(α,β,δ) distribution is $\alpha\beta + \delta$, that is, by choosing α,β and δ such that $\alpha\beta + \delta = 1$, we have $E(\tau_{uv}) = t_{uv}$.

We assume that at the beginning of the time horizon at $t = 0$, the locations of all vehicles providing service are known. Thus, the deterministic schedules $((v_1, t_1), \ldots, (v_p, t_p))$ of services are converted into random schedules $((v_1, \tau_1), \ldots, (v_p, \tau_p))$ by defining $\tau_1 \sim$ Gamma($\alpha t_1, \beta, \delta t_1$) and $\tau_i = \tau_{i-1} + \tau_{v_{i-1}v_i}$ for $i \in \{2, \ldots, p\}$.

The values of the α,β and δ parameters are $(1, 0.25, 0.75)$, which corresponds to the coefficient of variation $\sigma/\sqrt{\mu} = 0.25$ (Russell and Urban, 2007).

The algorithms were implemented in Mathematica and the tests were performed on a 2.2 GHz Dual Core Intel PC.

6.1. *Random and real-life networks*

In this section we present computational results of tests conducted on randomized instances and real-life data. The real-life instances are based on the tram schedules of Helsinki and the parameters of the randomized instances are based on the size of the public transport network of Helsinki including trams, buses and subways, which is approximately ten times the size of the tram network (Helsinki City Transport, 2011).

The real-life tram network consists of ten tram lines, operated by 50 trams, and 154 stops. In the randomized instances, there are 100 tram lines (and a subway line), operated by 500 trams (and 10 subway vehicles), and approximately 1500 stops. The instances are available at `http://math.tkk.fi/~lehame/dju`.

In order to be able to compare the proposed solution methods, we focus on problems which can be solved by all three models (conditional, history independent and unconditional), even though the approximate methods could be used to solve larger instances in resonable computation times.

6.1.1. *Real-life instances*

The tram schedules of Helsinki Region Transport available at (Helsinki Region Traffic, 2010b). In each instance, the origin and destination nodes v_o and v_d are chosen randomly from the set of 154 tram stops. The departure time is set equal to 9:00 and the length of the time horizon is defined by $T = 1.2d(v_o, v_d)$, where $d(v_o, v_d)$ is the expected duration of the shortest path from v_o to v_d in the tram network. The expected duration of the shortest path is computed by assuming that each edge between adjacent nodes in the tram network (see Figure 7) is associated with a positive duration, extracted from the timetables.

Each service k operating within the time horizon is determined by a *scheduled route*, that is, a sequence $((v_0^k, t_0^k), \ldots, (v_p^k, t_p^k))$ of nodes, where t_i^k is the mean arrival time at node v_i^k for $i = 0, \ldots, p$. The tram network consisting of ten tram lines, operated by 50 trams (=services), is shown in Figure 7. The maximum walking distance equals $d_w^{\max} = 0.25$, that is, all pairs of nodes, the distance between which is less than 0.25 km, are connected by walking legs. The speed of each walking leg equals $v_w = 5$ km/h and the duration of walking legs is modeled similarly as the duration of service legs, see Equation (17).

By means of the above procedure, ten instances marked with 'r' were generated by converting the deterministic travel times to gamma distributed stochastic travel times, as described in the beginning of this section. In each case, the legs of the tram services, for which the start and end time is within the time horizon with strictly positive probability, were included. In addition, all

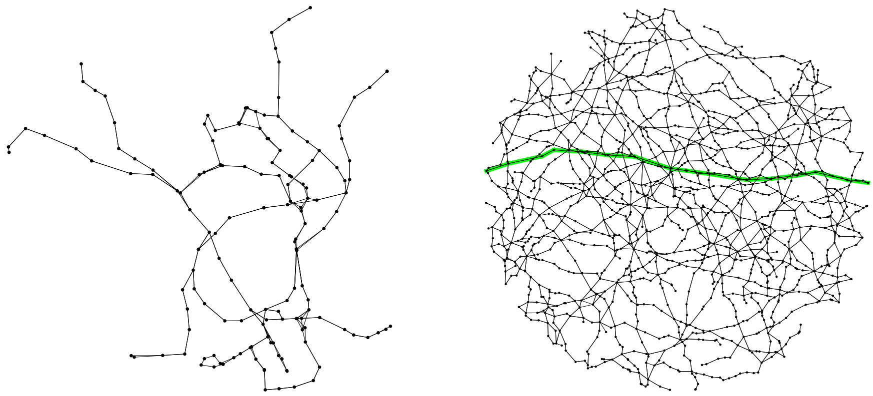

Figure 7: The real-life and randomized networks. The left figure shows the tram network of Helsinki consisting of ten tram lines and 154 stops. The right figure shows a randomized network consisting of 100 tram lines, one subway line (highlighted) and 1511 stops.

legs i for which $P(\tau'_i + \tau(v'_i, v_d) \leq T) = 0$, where $\tau(v'_i, v_d) \sim \text{Gamma}(\alpha d(v'_i, v_d), \beta, \delta d(v'_i, v_d))$ is the random duration of the shortest path from v'_i to v_d, were excluded.

6.1.2. Randomized instances

The randomized instances are divided into two sets: in the first set marked with 'd', the network consists of 100 lines operated by 500 services. In the second set marked with 'dm', there is an additional subway line operated by ten services. The set 'd' represents an extended tram network. The subway line is included in the set 'dm' due to the fact that in large urban areas, long journeys benefit from the multimodal nature of the network, resulting in relatively small travel times for long journeys.

First, n nodes are distributed randomly and uniformly in a disk in $\mathbb{R}^2$ with radius r (unit = 1 km). Then, the edges (roads) between the nodes are generated by Delaunay triangulation. Then, in order to remove the long edges on the border of the triangulation, the network is cropped by including the nodes within radius $0.9r$ from the center of the disk and the edges between them. The radius r of the disk is chosen such that the expected Delaunay edge length $E(L)$ is equal to the average distance (0.46 km) between adjacent stops in the tram network (Section 6.1.1). Since $E(L)$ is given by $E(L) \approx 1.132\lambda^{-\frac{1}{2}}$, where $\lambda = n/\pi r^2$ is the spatial density of nodes (Ritzerveld, 2007), we have $r = \sqrt{\frac{n}{\pi}}E(L)/1.132$.

The number of tram services in each instance is 500 and the services are divided equally among 100 lines. The lines are defined as a cycles in the Delaunay graph as follows.

1. The start node v_0^k of line k is chosen randomly from the set of all nodes.
2. The end node v_p^k of line k is chosen randomly from the set of nodes that define the convex hull of the network.
3. The shortest path $v_0^k, v_1^k, \ldots, v_{p-1}^k, v_p^k$ from the start node v_0^k to the end node v_p^k is determined. Line k is defined by the cycle $v_0^k, v_1^k, \ldots, v_{p-1}^k, v_p^k, v_{p-1}^k \ldots, v_1^k$, that is, a round-trip path between v_0^k and v_p^k.

Each line k is operated by five services h, whose starting points $v_0^{k,h}$ are chosen at equal intervals on the cycle that defines line k. Each service h operating on line k is determined by a scheduled route, that is, a sequence $(v_0^{k,h}, t_0^{k,h}), \ldots, (v_q^{k,h}, t_q^{k,h})$ of nodes and scheduled arrival times, where subsequent nodes $v_i^{k,h}, v_{i+1}^{k,h}$ are subsequent nodes of the cycle. The scheduled arrival time at node $v_0^{k,h}$ equals $t_0^{k,h} = 0$ and the scheduled arrival time at node $v_i^{k,h}$ is defined by $t_i^{k,h} = t_{i-1}^{k,h} + \|v_{i-1}^{k,h} - v_i^{k,h}\|/v$ for $i = 1, \ldots, q$, where $v = 17$ km/h.

In the subway instances, the subway line and services are generated similarly as the tram lines and services except that the start and end nodes of the subway line are chosen from the opposite sides of the convex hull of the network (see Figure 7). In addition, 2/3 of the nodes in the shortest path are removed due to the fact that the average distance between adjacent subway stops is approximately three times the distance between adjacent tram stops (Helsinki Region Traffic, 2010b). Furthermore, the speed of the subway services equals $v_m = 40$ km/h.

Similarly as in the real-life instances, all pairs of nodes, the distance between which is less than 0.25 km, are connected by walking legs.

The origin and destination nodes v_o and v_d are chosen randomly from the union of the nodes included in the tram (and subway) lines. The length of the time horizon is defined by $T = 1.2 \cdot d(v_o, v_d)$, where $d(v_o, v_d)$ is the expected duration of the shortest path from v_o to v_d in the network. The expected duration of the shortest path is determined by assuming that each edge (v_i, v_j) in the randomized network (see Figure 7) is associated with a positive duration ($\|v_i - v_j\|/v$ for tram edges, $\|v_i - v_j\|/v_m$ for subway edges).

Since not all nodes in the initial Delaunay triangulation are included in any of the tram or subway lines in the above procedure, an initial set of tests were conducted in order to determine the number n of nodes such that the number of nodes included in the tram and subway lines was approximately 1500. As a result of these tests, the initial number of nodes was set to $n = 2800$.

20 instances were generated by means of the above procedure by including in each case the legs of the services for which the end time is within the time horizon with strictly positive probability. In addition, all legs i for which $P(\tau_i' + \tau(v_i', v_d) \leq T) = 0$, where $\tau(v_i', v_d) \sim \text{Gamma}(\alpha d(v_i', v_d), \beta, \delta d(v_i', v_d))$ is the random duration of the shortest path from v_i' to v_d, were excluded.

6.1.3. Results

Table 1: Computational results of real-life instances, 99.9% screening. The decision time for each model is given by the sum of screening time and algorithm time.

Instance	Number of legs	Time horizon T (min)	Screening time (s)	Conditional model		History independent model		Unconditional model	
				Algorithm time (s)	Probability $V((1))$	Algorithm time (s)	Probability $V_H(1)$	Algorithm time (s)	Probability $V_U(1)$
r1	745	20.4	455.29	12.42	0.989	0.96	0.989	0.54	0.989
r2	559	18.	474.83	62.67	0.693	5.68	0.693	3.99	0.693
r3	90	9.6	36.33	4.15	0.482	0.19	0.5	0.14	0.504
r4	355	15.6	1419.24	898.54	0.839	2.64	0.855	2.08	0.859
r5	558	24.	234.2	5.94	0.181	0.49	0.202	0.28	0.203
r6	704	22.8	1646.43	98.31	0.677	3.8	0.678	3.04	0.678
r7	615	19.2	1642.85	30.05	0.555	3.27	0.59	2.66	0.602
r8	562	21.6	1028.88	20.94	0.295	1.42	0.445	1.08	0.455
r9	508	15.6	787.51	883.16	0.588	1.84	0.669	1.43	0.677
r10	85	10.8	37.57	4.64	0.311	0.59	0.312	0.49	0.312

The computational results of the real-life instances are shown in Tables 1, 2, 3, and the results of the randomized instances are shown in Tables 4, 5. In all tables, the first column shows the name of the instance, the second column shows the number of legs included the instance and the third column shows the length of the time window in minutes.

In order to be able to compute the probability in the conditional model, which requires the enumeration of all paths, the set of paths from the origin state to destination states was initially

Table 2: Computational results of real-life instances, 90% screening. The decision time for each model is given by the sum of screening time and algorithm time.

Instance	Number of legs	Time horizon T (min)	Screening time (s)	Conditional model		History independent model		Unconditional model	
				Algorithm time (s)	Probability $V((1))$	Algorithm time (s)	Probability $V_H(1)$	Algorithm time (s)	Probability $V_U(1)$
r1	745	20.4	1.41	8.53	0.989	0.72	0.989	0.43	0.989
r2	559	18.	1.03	9.52	0.692	0.64	0.693	0.53	0.693
r3	90	9.6	0.13	1.3	0.473	0.1	0.487	0.08	0.487
r4	355	15.6	2.59	51.79	0.835	1.07	0.853	0.69	0.856
r5	558	24.	0.77	2.88	0.178	0.36	0.185	0.13	0.185
r6	704	22.8	2.9	14.16	0.676	1.1	0.676	0.8	0.676
r7	615	19.2	2.91	4.98	0.555	0.57	0.578	0.42	0.591
r8	562	21.6	1.68	4.58	0.301	0.4	0.41	0.22	0.419
r9	508	15.6	2.05	65.03	0.588	1.5	0.669	1.16	0.677
r10	85	10.8	0.17	2.98	0.31	0.34	0.312	0.3	0.312

Table 3: Computational results of real-life instances, heuristic screening. The decision time for each model is given by the sum of screening time and algorithm time.

Instance	Number of legs	Time horizon T (min)	Screening time (s)	Conditional model		History independent model		Unconditional model	
				Algorithm time (s)	Probability $V((1))$	Algorithm time (s)	Probability $V_H(1)$	Algorithm time (s)	Probability $V_U(1)$
r1	745	20.4	1.14	6.53	0.988	0.62	0.989	0.35	0.989
r2	559	18.	0.48	8.59	0.686	0.62	0.693	0.54	0.693
r3	90	9.6	0.11	2.09	0.473	0.15	0.498	0.12	0.498
r4	355	15.6	1.46	33.06	0.834	0.84	0.853	0.56	0.854
r5	558	24.	0.38	1.18	0.162	0.2	0.173	0.04	0.173
r6	704	22.8	1.06	10.28	0.676	0.83	0.676	0.58	0.676
r7	615	19.2	0.53	4.14	0.548	0.39	0.577	0.27	0.589
r8	562	21.6	0.9	4.7	0.295	0.42	0.41	0.22	0.419
r9	508	15.6	1.07	18.88	0.586	0.66	0.641	0.47	0.643
r10	85	10.8	0.09	2.19	0.31	0.34	0.312	0.3	0.312

narrowed by excluding paths for which the probability of success is less than a minimum probability p with confidence level $1 - p$. The duration of this screening phase (in seconds) is shown in the third column of the tables.

The remaining columns show the algorithm times for the different models and the corresponding probability values for the origin state. The *decision time* for each model is given by the sum of screening time and algorithm time. The decision time corresponds to the time needed to produce an optimal policy in each instance, that is, the time needed to suggest a decision for the commuter.

Table 1 shows the results for the ten real-life instances described in Section 6.1.1. In the screening phase, paths for which the probability of success is less than 0.1% with a 99.9% confidence level were excluded. This screening phase was executed by performing $R = 6905$ random tests, in each of which all paths beginning from the origin state were enumerated (see Algorithms 2 and 3). For all states not excluded during the screening phase, the distributions of the start and end times in the conditional model were approximated by random sampling with $R = 10000$ samples.

Table 2 shows the corresponding results obtained by excluding paths for which the probability of success is less than 10% with a 90% confidence level. In this case, the screening phase involved $R = 22$ random tests and the conditional distributions were approximated by $R = 1000$ samples.

Table 3 shows the results obtained by using the 90% accuracy described above and a heuristic screening function: All legs $((v_i, \tau_i), (v_i', \tau_i'))$, for which the shortest distance $d(v_i', v_d)$ from the end node v_i' to the destination node v_d is greater than the shortest distance $d(v_i, v_d)$ from the start node v_i to the destination, were excluded from the search.

The corresponding results of the randomized instances are shown in Tables 4 and 5. Due to the size of the randomized networks, the 99.9% screening phase could not be executed in reasonable computing times, and thus only results with 90% confidence level are reported.

By looking at the screening times in Tables 1 and 2, we see that the time required for obtaining a 99.9% confidence level requires typically $200 \ldots 600$ times as much computational effort as obtaining a 90% confidence level. This is justified by the fact that the number of random tests re-

Table 4: Computational results of randomized instances, 90% screening. The instances marked with 'd' correspond to the extended tram network and the instances marked with 'dm' include a subway line. The decision time for each model is given by the sum of screening time and algorithm time.

Instance	Number of legs	Time horizon T (min)	Screening time (s)	Conditional model		History independent model		Unconditional model	
				Algorithm time (s)	Probability $V((1))$	Algorithm time (s)	Probability $V_H(1)$	Algorithm time (s)	Probability $V_U(1)$
d1	4152	31.7	580.9	5.24	0.064	0.43	0.125	0.13	0.125
d2	4926	30.8	159.7	34.58	0.553	2.06	0.553	1.73	0.553
d3	362	15.3	1.9	355.32	0.892	1.64	0.899	1.36	0.92
d4	1309	21.4	76.5	61.5	0.362	2.27	0.407	1.59	0.407
d5	5927	39.1	5308.4	7.22	0.539	0.61	0.539	0.1	0.539
d6	2455	28.7	729.6	54.19	0.135	1.13	0.179	0.58	0.185
d7	5283	35.6	1922.2	59.11	0.565	3.9	0.596	2.96	0.609
d8	1874	21.	445.5	3.37	0.364	1.05	0.769	0.76	0.769
d9	4437	38.4	381.	9.94	0.337	1.22	0.433	0.44	0.433
d10	4896	35.	10974.5	55.12	0.605	2.59	0.639	1.82	0.639
dm1	171	9.4	0.1	2.3	0.615	0.26	0.632	0.12	0.667
dm2	3119	29.5	611.9	11.22	0.468	1.76	0.61	1.34	0.61
dm3	1576	23.3	21.6	7.98	0.886	0.79	0.894	0.45	0.894
dm4	3719	25.7	562.8	34.07	0.425	3.17	0.505	2.37	0.572
dm5	1264	19.1	6.7	1.95	0.826	0.31	0.83	0.07	0.83
dm6	2195	21.4	3.9	1.8	0.102	0.28	0.125	0.07	0.125
dm7	1683	17.5	60.4	3.47	0.4	1.26	0.4	0.92	0.4
dm8	1526	18.7	4.	7.11	0.829	0.92	0.838	0.62	0.838
dm9	656	17.1	0.9	1.46	0.356	0.22	0.358	0.06	0.358
dm10	622	14.1	1.8	4.69	0.699	1.1	0.703	0.81	0.703

Table 5: Computational results of randomized instances, heuristic screening. The instances marked with 'd' correspond to the extended tram network and the instances marked with 'dm' include a subway line. The decision time for each model is given by the sum of screening time and algorithm time.

Instance	Number of legs	Time horizon T (min)	Screening time (s)	Conditional model		History independent model		Unconditional model	
				Algorithm time (s)	Probability $V((1))$	Algorithm time (s)	Probability $V_H(1)$	Algorithm time (s)	Probability $V_U(1)$
d1	4152	31.7	0.87	2.6	0.064	0.45	0.125	0.13	0.125
d2	4926	30.8	0.6	4.62	0.553	0.99	0.553	0.59	0.553
d3	362	15.3	0.14	3.25	0.866	0.68	0.875	0.5	0.913
d4	1309	21.4	0.33	8.54	0.083	0.54	0.208	0.26	0.22
d5	5927	39.1	0.69	7.61	0.525	0.62	0.539	0.1	0.539
d6	2455	28.7	2.34	14.1	0.116	0.8	0.162	0.36	0.162
d7	5283	35.6	2.96	34.6	0.557	2.76	0.595	2.05	0.609
d8	1874	21.	3.15	3.4	0.363	1.06	0.769	0.75	0.769
d9	4437	38.4	3.7	12.09	0.337	1.22	0.433	0.42	0.433
d10	4896	35.	4.51	20.28	0.597	2.45	0.639	1.62	0.639
dm1	171	9.4	0.07	2.36	0.615	0.27	0.632	0.12	0.667
dm2	3119	29.5	2.4	11.98	0.443	3.74	0.61	2.86	0.61
dm3	1576	23.3	0.86	1.92	0.867	0.3	0.87	0.08	0.87
dm4	3719	25.7	3.63	12.59	0.425	2.53	0.505	1.93	0.572
dm5	1264	19.1	0.68	1.99	0.82	0.31	0.83	0.07	0.83
dm6	2195	21.4	0.37	1.86	0.099	0.28	0.125	0.07	0.125
dm7	1683	17.5	0.44	3.53	0.4	1.24	0.4	0.94	0.4
dm8	1526	18.7	0.21	1.46	0.826	0.22	0.833	0.06	0.833
dm9	656	17.1	0.11	1.48	0.33	0.22	0.358	0.06	0.358
dm10	622	14.1	0.36	3.	0.688	0.43	0.703	0.13	0.703

quired are 6905 and 22, respectively. The screening time is significantly greater in the randomized instances, as can be seen by looking at Table 4. Even obtaining the 90% confidence level requires three hours screening time in the worst case (instance d10). However, the screening time can be decreased by using a heuristic function (Tables 3,5). In some cases, for example dm1, the heuristic screening is relatively accurate compared to 90% screening. However, there are also cases in which the heuristic screening produces a significant error (for example d4).

By looking at the algorithm times in the tables, we see that the unconditional and history independent algorithms are significantly, typically of order ten times, faster than the conditional model. Furthermore, in a part of the instances, the difference between the results obtained by the three models is insignificant. For example, in instance r6, the probability of reaching the destination state is equal with three-digit precision for all studied models (Tables 1, 2). However, the results also indicate that the modeling error in the history independent and unconditional models is significant in some cases. For example, in instance d8, the conditional probability is 0.364 and the probability obtained by the history independent and unconditional models is 0.769. The difference between the history independent model and the unconditional model is relatively small in all cases.

As a conclusion, we state that even though the algorithm time of the conditional model is significantly greater compared to the history independent and unconditional models, the conditional model is justified by the fact that it may be used to estimate the error in approximate methods. By heuristic screening, the decision time (screening time + algorithm time) in the studied instances is half a minute at most in the conditional model, and a few seconds at most in the other models.

6.2. Regular networks

The purpose of the following experiments is to compare the different models with respect to three parameters, namely, transfer margin, number of transfer options at each node and the number of transfers. While these experiments do not represent any particular real-life scenario, the results give us an idea of the main differences of the proposed models. Results on more realistic cases are presented in Section 6.1.

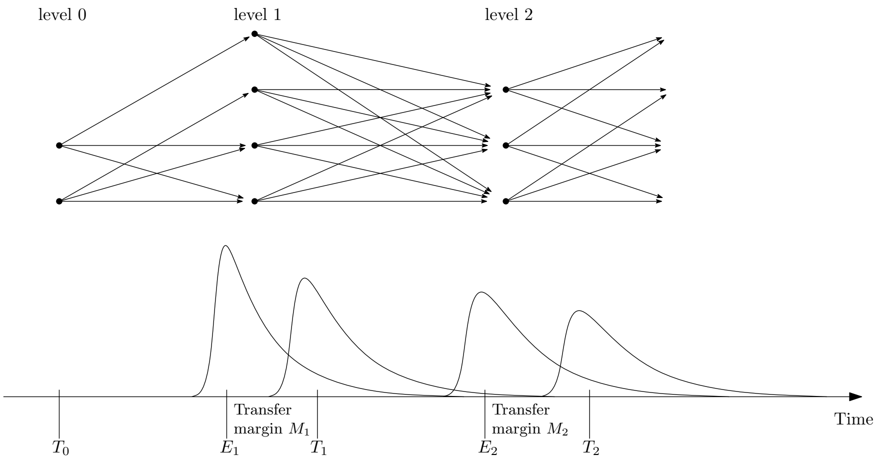

Figure 8: An example of a journey involving two transfers ($N = 2$). In this example there are two nodes at level 0, four nodes at level 1 and three nodes at level 2. For each node h at level $i \in \{0, 1, 2\}$ there are $K = 3$ services starting at h and ending at a node at level $i + 1$. The expected start time for each service starting at level i is denoted by T_i and the expected end time for each service ending at level $i + 1$ is given by $E_{i+1} = T_i + 1$. The expected start time for each service starting at level $i + 1$ is defined by $T_{i+1} = E_{i+1} + M_{i+1}$, where M_{i+1} is the studied transfer margin at level $i + 1$.

We consider the following class of scheduled networks: There are m nodes arranged in $N + 1$ *levels* numbered by $0, 1, \ldots, N$, (see Figure 8). The commuter departs at level 0 and the goal is to successfully transfer between two services at each level $\{1, \ldots, N\}$. For each node h at level $i \in \{0, \ldots, N - 1\}$, there are K services, each consisting of a single leg, which starts at node h at level i and ends at a node at level $i + 1$. The mean start time of each service starting at level i is denoted by T_i and the mean end time of each service ending at level i is denoted by E_i. The mean start time at level 0 satisfies $T_0 = 0$. The mean end time E_i satisfies $E_i = T_{i-1} + 1$ for all $i \in \{1, \ldots, N\}$. The mean start time T_i is defined by $E_i + M_i$ for all $i \in \{1, \ldots, N\}$, where M_i is the *transfer margin* at level i.

The start times of services starting at level i are independently and identically distributed according to the distribution Gamma($\alpha T_i, \beta, \sigma T_i$). In the conditional model, the arrival time τ'_k of each service k is defined by $\tau'_k = \tau_k + D_k$, where τ_k is the start time of service k and D_k is a random variable defined by $D_k \sim \text{Gamma}(\alpha, \beta, \delta)$. Note that the realization of τ_k affects the distribution of τ'_k.

In the history independent and unconditional models, the start and end times of a leg are assumed to be independent and thus the end time τ'_k of transport service k ending at level i is defined by $\tau'_k \sim \text{Gamma}(\alpha E_i, \beta, \delta E_i)$. Note that this coincides with the prior distribution of $\tau_i^{k'}$ in the conditional model.

Table 6: A summary of performed experiments.

Experiment number	Number of transfers N	Transfer margin M $(\cdot\sigma)$	Number of transfer options K	Studied parameter	Difference between models	Figure number
1	2	$-2\ldots,3$	$1,3$	Transfer margin M	Greatest with $-1 < M < 2$	9
2	2	$0.5, 1.5$	$1,\ldots,10$	Number of transfer options K	Increases with K	10
3	$1,\ldots,10$	1.5	$1,2,3$	Number of transfers N	Increases with N	11

A summary of performed experiments and the corresponding results is shown in Table 6.

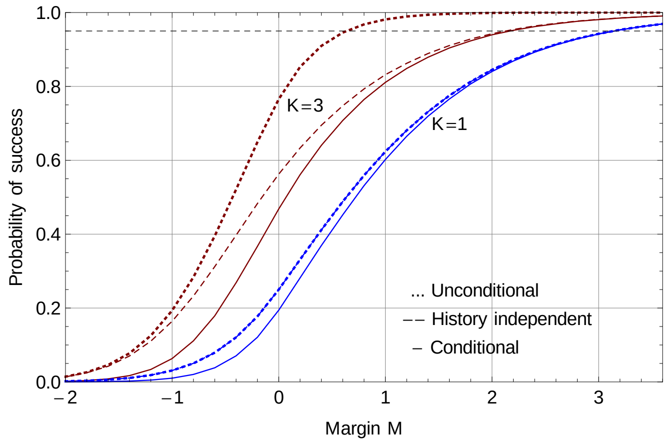

Figure 9: Experiment 1: Comparison of three probability models as a function of the transfer margin. The solid, dotted and dashed lines represent the probability of reaching the destination in the conditional model, history independent model and the unconditional model, respectively, with $N = 2$ and $K = 1, 3$. The history independent and unconditional models coincide when $K = 1$. With $K = 3$, the difference between the unconditional model and the conditional model is significant. The approximate methods become accurate with tight and loose transfer margins and the difference is greatest with $-1 < M < 2$.

6.2.1. Experiments

In the following three experiments we study the differences between the results obtained by the conditional, history independent and unconditional models.

Experiment 1

Let us first study the differences between the three models defined by Equations (3), (14) and (16). At first we compare the models as a function of transfer margin. We study the transfer

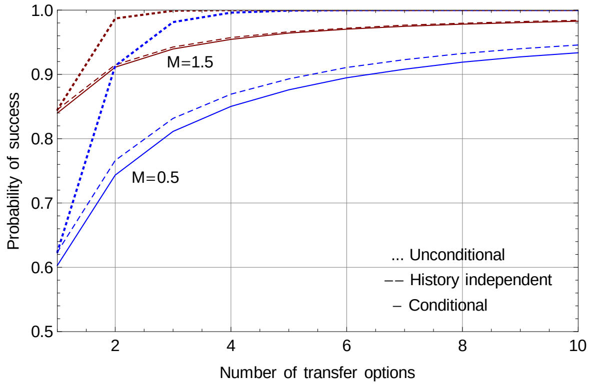

Figure 10: Experiment 2: Comparison of three probability models as a function of the number of transfer options K. The solid, dashed and dotted lines represent the probability of reaching the destination in the conditional model, history independent model and the unconditional model, respectively, with $N = 2$ and $M = 0.5, 1.5$. The unconditional model becomes less accurate when the number of transfer options is increased. With $M = 1.5$, the difference between the history independent model and the conditional model is relatively small and with $M = 0.5$, the difference is greater but does not increase significantly with K.

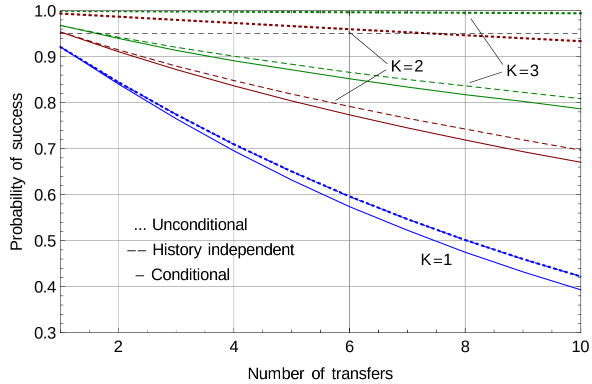

Figure 11: Experiment 3: Comparison of three probability models as a function of the number of transfers N. The solid, dashed and dotted lines represent the probability of reaching the destination in the conditional model, history independent model and the unconditional model, respectively, with $M = 1.5$ and $K = 1, 2, 3$. The difference between the unconditional model and the conditional model increases with N. The history independent model remains relatively accurate even if the number of transfers is increased.

margin as proportional to standard deviation: at each level i, the transfer margin M_i is defined by $M_i = \sigma_i M = \sqrt{E_i}\beta M$, where E_i is the expeced arrival time at level i and M is the ratio of transfer margin M_i to standard deviation σ_i.

The solid, dashed and dotted lines in Figure 9 represent the probability of a successful journey in the conditional model, history independent model and the unconditional model, respectively, with $N = 2$ and $K = 1, 3$.

As mentioned in the end of Section 5.4, we see that the history independent and unconditional models coincide when $K = 1$. With $K = 3$, the difference between the unconditional model and the conditional model is significant, whereas the history independent model is relatively accurate in this case. Note that the approximate methods become accurate with tight and loose transfer margins and that the difference is greatest with $-1 < M < 2$.

Experiment 2

Let us then compare the three models as a function of the number of transfer options K. The solid, dashed and dotted lines in Figure 10 represent the probability of a successful journey in the conditional model, history independent model and the unconditional model, respectively, with $N = 2$ and $M = 0.5, 1.5$.

By looking at the figure it is clear that the unconditional model becomes less accurate when the number of transfer options is increased. With $M = 1.5$, the difference between the history independent model and the conditional model is relatively small and with $M = 0.5$, the difference is greater but does not increase significantly with K.

Experiment 3

Finally, we examine the difference between the three models as a function of the number of transfers N. The solid, dashed and dotted lines in Figure 11 represent the probability of a successful journey in the conditional model, history independent model and the unconditional model, respectively, with $M = 1.5$ and $K = 1, 2, 3$.

Clearly, the difference between the unconditional model and the conditional model increases with N. The history independent model remains relatively accurate even if the number of transfers is increased.

7. Conclusions

We consider a dynamic journey planning problem in scheduled transportation networks. Due to uncertainty in travel times, the arrival times of transport services at stops are defined as random variables. The problem is modeled as a Markov decision process in which the travel history is included in the definition of a state. We present an algorithm for producing an optimal policy for traveling from a given origin to a given destination, assuming that the remaining path to the destination can be modified at any time during the journey. We also show that the importance of being able to reconsider the remaining path to the destination is emphasized when the number of transfers is increased.

The history dependent solution is further approximated by assuming history independence and unconditionality of successful transfers between transport services. The different models are evaluated by means of analysis and numerical experiments. The results suggest that the proposed approximations may be useful for practical purposes due to their computational simplicity.

References

Androutsopoulos, K. N., Zografos, K. G., 2009. Solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network. European Journal of Operational Research 192, 18–28.

Azaron, A., Kianfar, F., 2003. Dynamic shortest path in stochastic dynamic networks: Ship routing problem. European Journal of Operational Research 144, 138–156.

Bander, J., White, C. C., 1991. A new route optimization algorithm for rapid decision support. Proc. Vehicle Navigat. Inf. Syst. Conf. Part 2 P-253, 709–728.

Bander, J. L., White III, C. C., 2002. A heuristic search approach for a nonstationary shortest path problem with terminal costs. Transportation Science 36, 218–230.

Bard, J. F., Bennett, J. E., 1991. Arc reduction and path preference in stochastic acyclic networks. Management Science 37 (2), 198–215.

Bellman, R., 1957. A markovian decision process. Journal of Mathematics and Mechanics (6).

Boyan, J., Mitzenmacher, M., 2001. Improved results for route planning in stochastic transportation networks. In: Proceedings of the Twelfth Annual Symposium on Discrete Algorithms. pp. 895–902.

Brownstone, D., Small, K. A., 2005. Valuing time and reliability: assessing the evidence from road pricing demonstrations. Transportation Research Part A 39, 279–293.

Brub, J., Potvin, J., Vaucher, J., 2006. Time-dependent shortest paths through fixed sequence of nodes: Application to a travel planning problem. Comput. Oper. Res. 33 (6), 1838–1856.

Cai, X., Kloks, T., Wong, C. K., 1997. Time-varying shortest path problems with constraints. Networks 29 (3), 141–149.

Chabini, I., 1998. Discrete dynamic shortest path problems in transportation applications. Transp. Res. Rec. 1645, 170–175.

Cheung, R. K., 1998. Iterative methods for dynamic stochastic shortest path problems. Naval Research Logistics 45, 769–789.

Cheung, R. K., Muralidharan, B., 2000. Dynamic routing for priority shipments in ltl service networks. Transportation Science 34, 86–98.

Chiang, Y. S., , Roberts, P. O., 1980. A note on transit time and reliability for regular-route trucking. Transport Research Part B 14s (1-2), 59–65.

Cooke, K. L., Halsey, E., 1966. The shortest route through a network with time-dependent intermodal transit times. J. Math. Anal. Appl. 14, 493–498.

Datar, M., Ranade, A., 2000. Commuting with delay prone buses. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 22–29.

Davies, C., Lingras, P., 2003. Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks. European Journal of Operational Research 144, 27–38.

Fan, Y., Kalaba, R., Moore, J., 2005a. Arriving on time. Journal of Optimization Theory and Applications 127 (3), 497–513.

Fan, Y., Kalaba, R., Moore, J., 2005b. Shortest paths in stochastic networks with correlated link costs. Computers and Mathematics with Applications 49, 1549–1564.

Ferris, M. C., Ruszczynski, A., 2000. Robust path choice in networks with failures. Networks 35, 181–194.

Fosgerau, M., Karlström, A., 2010. The value of reliability. Transportation Research Part B 44, 38–49.

Fu, L., Rillet, L., 1998. Expected shortest paths in dynamic and stochastic traffic networks. Transportation Research 32 (7), 499–516.

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

Hamacher, H., Ruzika, S., Tjandra, S., 2006. Algorithms for time-dependent bicriteria shortest path problems. Discrete Optim. 3 (3), 238–254.

Helsinki City Transport, Nov. 2011. Helsinki city transport annual reports.
URL `http://www.hel.fi/hki/hkl/en/About+HKL/Annual+Reports`

Helsinki Region Traffic, Oct. 2010a. Jouko neighbourhood routes and service routes.
URL `http://www.hsl.fi/EN/passengersguide/ServiceroutesandJoukoroutes/`

Helsinki Region Traffic, Oct 2010b. Kalkati.net XML representation of helsinki regional transport authority (HSL) timetable data.
URL `http://developer.reittiopas.fi/`

Horn, M., 2003. An extended model and procedural framework for planning multi-modal passenger journeys. Transp. Res. Part B 37 (7), 641–660.

Huang, R., Peng, Z. R., 2001. An integration of network data model and routing algorithms for online transit trip planning. In: Proc. 80th Annu. Meeting Transp. Res. Board. Washington, DC.

Huang, R., Peng, Z. R., 2002. Schedule-based path finding algorithms for online transit trip planning. In: Proc. 81st Annu. Meeting Transp. Res. Board. Washington, DC.

Jaillet, P., 1992. Shortest path problems with node failures. Networks 22, 589–605.

Jula, H., Dessouky, M., Ioannou, P. A., 2006. Truck route planning in nonstationary stochastic networks with time windows at customer locations. IEEE Transactions on Intelligent Transportation Systems 7 (1), 51–62.

Kamburowski, J., 1985. A note on the stochastic shortest route problem. Operations Research 33 (6), 696–698.

Kao, E. P. C., 1978. A preference order dynamic program for a stochastic traveling salesman problem. Operations Research 26 (6), 1033–1045.

Kenyon, A. S., Morton, D. P., 2003. Stochastic vehicle routing with random travel times. Transportation Science 37 (1), 69–82.

Kim, S., Lewis, M., White III, C., 2005a. Optimal vehicle routing with real-time traffic information. IEEE Transactions on Intelligent Transportation Systems 6 (2), 178–188.

Kim, S., Lewis, M., White III, C. C., 2005b. State space reduction for non-stationary stochastic shortest path problems with real-time traffic congestion information. IEEE Transactions on Intelligent Transportation Systems 6 (3), 273–284.

Kostreva, M., Wiecek, M., 1993. Time dependency in multiple objective dynamic programming. J. Math. Anal. Appl. 173 (1), 289–307.

Lam, T. C., Small, K. A., 2001. The value of time and reliability: measurement from a value pricing experiment. Transportation Research Part E: Logistics and Transportation Review 37 (2-3), 231–251.

Loui, R. P., 1983. Optimal paths in graphs with stochastic or multidimensional weights. Communications of the ACM

26, 670–676.
Meyer, C. D., 2000. Matrix Analysis and Applied Linear Algebra. SIAM, Philadelphia.
Miller-Hooks, E. D., Mahmassani, H. S., 2000. Least expected time paths in stochastic, time-varying transportation networks. Transportation Science 34, 198–215.
Mirchandani, P. B., Soroush, H., 1985. Optimal paths in probabilistic networks: A case with temporary preferences. Computers & Operations Research 12, 365–383.
Modesti, P., Siomachen, A., 1998. A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. Eur. J. Oper. Res. 111 (3), 495–508.
Mulley, C., Nelson, J. D., 2009. Research in Transportation Economics 25 (1), 39 – 45.
Murthy, I., Sarkar, S., 1997. Exact algorithms for the stochastic shortest path problem with a decreasing deadline utility function. European Journal of Operational Research 103, 209–229.
Nie, Y. M., Wu, X., 2009. Shortest path problem considering on-time arrival probability. Transportation Research Part B 43, 597–613.
Nikolova, E., Brand, M., Karger, D., 2006a. Optimal route planning under uncertainty. In: Proceedings of International Conference on Planning and Scheduling, Lake District, England.
Nikolova, E., Kelner, J., Brand, M., Mitzenmacher, M., 2006b. Stochastic shortest paths via quasi-convex maximization. In: Proceedings of European Symposium of Algorithms, Zurich, Switzerland. pp. 552–563.
Peer, S., Sharma, D. K., 2007. Finding the shortest path in stochastic networks. Computers and Mathematics with Applications 53, 729–740.
Peng, Z. R., Huang, R., 2000. Design and development of interactive trip planning for web-based transit information systems. Transp. Res. Part C 8 (1), 409–425.
Polychronopoulos, G. H., Tsitsiklis, J. N., 1996. Stochastic shortest path problems with recourse. Networks 27, 133–143.
Psaraftis, H. E., Tsitsiklis, J. N., 1993. Dynamic shortest paths in acyclic networks with markovian arc costs. Operations Research 41 (1), 91–101.
Ritzerveld, N. G. H., 2007. Doctoral Thesis, Leiden University.
Russell, R. A., Urban, T. L., 2007. Vehicle routing with soft time windows and erlang travel times. Journal of the Operational Research Society 59, 1220–1228.
Sigal, C. E., Pritsker, A. A. B., Solberg, J. J., 1980. The stochastic shortest route problem. Operations Research 28 (5), 1122–1129.
Small, K., Winston, C., Yan, J., 2005. Uncovering the distribution of motorists' preferences for travel time and reliability. Econometrica 73 (4), 1367–1382.
Tan, M. C., Tong, C. O., Wong, S. C., Xu, J., 2007. An algorithm for finding reasonable paths in transit networks. J. Adv. Transp. 41 (3), 285–305.
Thomas, B., White, C., 2007. The dynamic shortest path problem with anticipation. European Journal of Operational Research 176 (2), 836–854.
Tong, C. O., Richardson, A. J., 1984. A computer model for finding the timedependent minimum path in transit systems with fixed schedules. J. Adv. Transp. 18 (2), 145–161.
Tong, C. O., Wong, S. C., 1999. A stochastic transit assignment model using a dynamic schedule-based network. Transp. Res. Part B 33 (2), 107–121.
Waller, S., Ziliaskopoulos, A., 2002. On the online shortest path problem with limited arc cost dependencies. Networks 40 (4), 216–227.
Wong, S. C., Tong, C. O., 1998. Estimation of time-dependent origin-destination matrices for transit networks. Transp. Res. Part B 32 (1), 35–48.
Ziliaskopoulos, A., Mahmassani, H., 1993. Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. Transp. Res. Rec. 1408, 94–100.
Ziliaskopoulos, A., Wardell, W., 2000. An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. Eur. J. Oper. Res. 125 (3), 486–502.
Zografos, K. G., Androutsopoulos, K. N., 2008. Algorithms for itinerary planning in multimodal transportation networks. IEEE Transactions on Intelligent Transportation Systems 9 (1).