Arrival time dependent routing policies in public transport

Kristóf Bérczi [a,b,*], Alpár Jüttner [a,b], Marco Laumanns [c], Jácint Szabó [c]

[a] *MTA-ELTE Egerváry Research Group, Budapest, Hungary*
[b] *Department of Operations Research, Eötvös Loránd University, Budapest, Hungary*
[c] *IBM Research, Rüschlikon, Switzerland*

ARTICLE INFO

ABSTRACT

We present a routing system that considers uncertainties, which are prevalent in any real transport system. Given desired departure or arrival times and a utility function representing the traveller's preferences, our method computes not just a single path through the network, but a more sophisticated and adaptive journey plan called routing policy. For each stop and time instance, a policy specifies the list of services that the passenger is recommended to take.

We show that the problem of finding an optimal policy is NP-hard. We also give a polynomial-time algorithm for a relaxation of the problem when the number of recommended services is limited at each stop and time. A computational case study for the public transport network of Budapest shows that the obtained routing policies can lead to substantial travel time savings compared to deterministic plans, and that considering multiple service policies leads to an improvement compared to previous solutions using single-service policies.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Public transport plays an essential role in reducing the traffic load, but disturbances due to congestion as well as planned or unplanned events such as maintenance work or accidents can have strong effects on travel times of public transport vehicles and its quality of service. As a result, public transport is often perceived as unreliable and therefore not always used to its full potential.

Journey planning is a key process in public transport, where travellers get informed how to make the best use of a given system for their individual travel needs. Nowadays, public transport providers routinely offer journey planning applications either on their websites or via mobile applications. For a given journey request, these applications usually offer one or several routes, which are linear sequences of activities or legs that form the itinerary. A common trait of the underlying journey planning algorithms is that they assume a deterministic environment. However, as changing traffic conditions can have strong effects on travel times, vehicles in public transport often deviate from their schedule.

On the other hand, there are often multiple alternative services which a passenger can choose from at a given location. Public transport providers have started to take advantage of such options and offer dynamic journey planning capabilities, for example push services to alert travellers of broken connections and to deliver updated journey plans by re-planning the journey anew from the current location and based on the current traffic conditions. Despite providing some adaptability, linear journey plans are not able to capture the full amount of flexibility inherent in multi-service public transport systems, because re-planning always occurs after some deviation from the assumed original conditions has occurred.

---

* Corresponding author at: MTA-ELTE Egerváry Research Group, Budapest, Hungary.
E-mail addresses: berkri@cs.elte.hu (K. Bérczi), alpar@cs.elte.hu (A. Jüttner), mlm@zurich.ibm.com (M. Laumanns), mlm@zurich.ibm.com (J. Szabó).

To fully exploit the given flexibility and to pre-plan adaptive decisions accordingly, we propose to use the concept of a routing policy instead of a linear journey plan. A routing policy is a state dependent routing advice at each location. A routing advice may consist of more than one service and specifies exactly which service to take in each situation. The traveller may define an arrival time dependent utility value at the destination, representing her preferences regarding arrival time deviations and delays. The goal is to find a policy that maximizes the expected value of the utility that can be achieved by following the policy.

Previous work. Finding shortest paths in a network with respect to given arc travel times is a central problem in combinatorial optimization. The fundamental results of Dijkstra [9], Bellman [3], Ford [14] and Floyd [13] are well-known examples of route planning algorithms. Due to their wide applicability, improved versions of these algorithms were introduced, e.g. the A* algorithm by Hart et al. [19], which is the basis of many street map based route planners. A common property of these algorithms is that they work with static arc travel times and provide a single route as a solution.

There have been various attempts to address uncertainty in routing problems. The travel time of an arc can be modelled as a random variable with a given probability distribution. In this model the effectiveness of a route can be measured in many ways: both the expected travel time and the reliability are worth to be considered. When time-independent distributions are assumed, one can take the mean travel times and find the path with minimum expected travel time by Dijkstra's algorithm. Time-dependent distributions have also been considered. In [18], Hall showed that standard shortest path algorithms do not necessarily find minimum travel time paths in this case. He introduced a 'time-adaptive decision rule' in which the next node is defined for each step as a function of the arrival time to the node. However, mostly heuristics are known for these problems, see for example [2,27].

A more general framework is obtained if the quality of a route is measured in terms of a utility function. Loui [21] showed that optimal solutions can be found for numerous special cases such as linear, exponential or quadratic utilities. In one of the earliest works, Frank [15] defined a reliable optimal path as a path maximizing the probability of arriving before a given time. This problem was later given the name Stochastic On-Time Arrival (SOTA) problem [10]. Murthy and Sarkar [22] gave an exact solution to the SOTA problem in case the travel times follow normal distributions.

Besides looking for optimal paths, a lot of research has been done for finding optimal routing plans, mainly based on models and techniques from stochastic dynamic programming. A recourse model for stochastic shortest paths was considered in [24]. Fan et al. [10] applied stochastic dynamic programming to find on-line travel plans, where the next arc is computed depending on the travel times on the path already visited. Fan and Nie [12] reformulated the SOTA problem of [15] in terms of stochastic dynamic programming. Their standard successive approximation algorithm converges in acyclic networks, but its running time is unbounded in the general case. In [11] they proposed a pseudo-polynomial discrete approximation algorithm for the same problem, which converges in a finite numbers of steps. Hoy and Nikolova [20] presented a dynamic programming algorithm to find approximately optimal arrival time dependent paths in directed acyclic networks. Results on the SOTA problem were theoretically and numerically further improved by Samaranayake et al. [25]. The computational difficulties come from the fact that continuous-time convolution products have to be computed several times, which is hard to solve analytically. Although discretizing the problem might help, the computational effort of the convolution steps remains large; therefore methods speeding up this process are of interest, see e.g. [26].

There is a fundamental difference between individual transport and public transport routing problems. In the individual transport case, due to the absence of transfers, the delay uncertainties on road segments simply add up. Hence, if a distribution is assumed on the arcs that fulfils certain properties, the travel time distribution of a given path can easily be computed. These results cannot be directly applied to the public transport case, where transfers are fundamental elements of the system. Botea et al. [6] and Dibbelt et al. [8] investigated public transport journey planning in the presence of uncertainty. They proposed a model in which uncertainty appears as estimated times of arrival of transport vehicles and the duration of actions such as walking. In [6] a heuristic solution is found by using Weighted AO* as a baseline search method, extended with different speed-up enhancements. The solution is a probabilistic plan where branches correspond to single choices at a given stop and have probabilities. The quality of the solution is measured in terms of robustness to uncertainty. In [8] a dynamic programming algorithm is given to find an arrival time dependent single choice routing plan minimizing expected arrival time.

If there are multiple services serving the same stop, there might be several good service choices depending on the actual arrival times. Thus, even if we were able to find a single choice with largest expected utility, the solution may be suboptimal. The difficulty of the problem therefore lies in selecting a set of 'alternatives' for each node instead of a single path. This framework was previously studied by Nonner [23] who proposed a method that computes a list of services for each stop depending on the waiting time distributions, and the user is recommended to take the service from the list that arrives first. As a respective model, Nonner suggested to determine the waiting time distributions as a function of the passing frequencies of the services at a given node. The method is also able to use online data provided by monitoring devices, in order to re-plan and update the lists of alternatives for each stop whenever new information concerning delays, position of a vehicle, unexpected events, or level of congestion arrives. The notion of 'routing alternatives' of [23] was further developed by the first author et al. [4]. They adopted the notion of 'policies' and provided current time dependent policies consisting of not only a single service but a set of services, so that the resulting travelling advice gets more robust to uncertainty.

Contribution. The aim of the present paper is to examine arrival time dependent policies with multiple service choices for monotone non-increasing utility functions. We formalize the problem and introduce the notion of arrival time dependent policies in Section 2. In Section 3 we show that there always exists an optimal policy, although finding one is NP-hard. However, if the number of services that a policy may contain is bounded by a constant k for each stop and time then an

optimal policy can be found in polynomial-time; the algorithm is discussed in Section 4. Section 5 gives an evaluation of arrival time dependent policies by comparing them to deterministic routing plans through extensive simulations.

2. Stochastic model

In the forthcoming discussion, we use a simplification of real public transport systems. For ease of representation, we assume that the transport system consists of individual services between two nodes, that is, each service has only two stops: its origin and its destination node. We also assume that the probability of two services arriving to the same stop at the same time is negligible. Of course this simplified set-up gives a very rough model of real public transport systems, but enables the usage of a much simpler, clearer notation and descriptions of the algorithms. It is important to note that the presented methods and formulas can be adapted to the case when services have several stops and the services may arrive to the same stop simultaneously. Accordingly, the computational case study of Section 5 uses a more complex and realistic model which considers services with several stops, and services are allowed to arrive to a given stop at the same time.

The stochastic model for the public transport network that we consider is the following. The *set of stops* is represented by a set of nodes V. We denote the *set of services* by S. The set of services leaving from node v is denoted by S_v and the maximum number of services leaving from any node is denoted by $\Delta = \max_{v \in V}\{|S_v|\}$. The *origin* and *destination nodes* of service $s \in S$ are denoted by o_s and d_s, respectively. The *transport network* is then represented by a directed graph $D = (V, A)$ where $vw \in A$ if and only if there is a service $s \in S$ with $o_s = v$ and $d_s = w$. The indicator variable of some logical statement is denoted by χ, for example, $\chi(x \in X)$ is 1 if $x \in X$ and 0 otherwise.

We consider two kinds of probabilistic data: the probability distribution of arrival and departure times, and the probability distribution of travel times. To characterize these, we introduce the following random variables: given a service s, the random variable of the departure time of s from o_s is denoted by D_s, the random variable of the arrival time of s to d_s is denoted by A_s, and the random variable of the travel time of s from o_s to d_s is denoted by T_s. It is assumed that there exists a global constant $\tau > 0$ such that $T_s \geq \tau$ for each $s \in S$. Clearly, we have

$$A_s = D_s + T_s.$$

For a given random variable X, we will denote the corresponding *cumulative distribution function (CDF)* by $F_X(t)$ and the *probability density function (PDF)*, if exists, by $f_X(t)$. The *expected value* of X is denoted by $E[X]$, while the *conditional expected value* of X given an event H is denoted by $E[X|H]$. If X is a discrete random variable then $P[X \in Z]$ denotes the probability of X having a value from Z.

Our goal is to give advice how passengers should best use the given stochastic transport network for their travel request from a given origin to a given destination. We formalize such a travel advice as a routing policy. Informally, a routing policy assigns a set of services to each stop and time. The set of recommended services depends on the passenger's arrival time to the stop, and will not change during the stay at the given stop. More precisely, a *policy* is a function $\mathcal{P} : V \times \mathbb{R} \to 2^S$ such that $\mathcal{P}_v(t) := \mathcal{P}(v, t) \subseteq S_v$. Let $V' \subseteq V$ and assume that for each $v \in V'$ a time $t_v \in \mathbb{R}$ is given. If $\mathcal{P}_v(t)$ is only defined for $t \geq t_v$ where $t_v \in \mathbb{R}$ for all $v \in V$, then it is called a *partial policy*.

The interpretation of a policy is the following. If a passenger arrives to stop v at time t_0 and a service $s \in \mathcal{P}_v(t_0)$ departs at time $t > t_0$ from stop v, then the passenger is recommended to take the service at time t. Ties among services leaving simultaneously is broken uniformly at random. The set of possible policies is denoted by $\mathfrak{P}$.

We will characterize the quality of a policy by the utility it provides to the passenger. A *utility function* is a function $u : \mathbb{R} \to \mathbb{R}$ which represents the user's preference according to the time of arrival to the destination node. We assume the existence of a certain 'deadline' (t_{end}) after which the user's utility is $-\infty$.

Assume that a destination node v_d, a utility function $u : \mathbb{R} \to \mathbb{R}$ and a policy $\mathcal{P}$ are given. For any stop $v \in V$ and time $t \in \mathbb{R}$, the *utility induced by $\mathcal{P}$ at stop v and time t* is denoted by $\mathcal{U}_v^{\mathcal{P}}(t)$. That is, $\mathcal{U}_v^{\mathcal{P}}(t)$ denotes our expected utility supposing we arrive to stop v at time t and we continue our journey according to the policy $\mathcal{P}$. The *conditional utility induced by $\mathcal{P}$ and service s at stop v and time t* is denoted by $\mathcal{U}_v^{\mathcal{P}}(t \mid s)$ and is defined as our expected utility if we take service s at stop v and time t. When considering continuous time, the induced utility can be computed as

$$\mathcal{U}_v^{\mathcal{P}}(t) = \int_t^\infty \sum_{s \in \mathcal{P}_v(t)} \left(\mathcal{U}_v^{\mathcal{P}}(t'|s) f_{D_s}(t') \prod_{\substack{s' \in \mathcal{P}_v(t) \\ s' \neq s}} \left(1 - \int_t^{t'} f_{D_{s'}}(t'') dt'' \right) \right) dt',$$

where

$$\mathcal{U}_v^{\mathcal{P}}(t|s) = \int_\tau^\infty \mathcal{U}_{d_s}^{\mathcal{P}}(t + t') f_{T_s}(t') \, dt'.$$

To circumvent the computational difficulties involved when handling continuous functions, we restrict the problem to a *finite time horizon* and discretize the planning horizon into small intervals of a few seconds. We can therefore assume that every event happens at time $t \in [T] = \{0, 1, \ldots, T\}$, where T corresponds to the deadline. Consequently, all distributions are discrete on the set $\{0, 1, \ldots, T\}$. In the discrete version, a policy is a function $\mathcal{P} : V \times [T] \to 2^S$. The travel time of a service is strictly positive and, as the time is discretized, at least as large as one time interval.

As we assumed that services do not arrive simultaneously, in the discrete case we have

$$\mathcal{U}_v^{\mathcal{P}}(t) = \sum_{t'=t}^{T} \sum_{s \in \mathcal{P}_v(t)} \left(\mathcal{U}_v^{\mathcal{P}}(t'|s) P[D_s = t'] \prod_{\substack{s' \in \mathcal{P}_v(t) \\ s' \neq s}} \left(1 - \sum_{t''=t}^{t'-1} P[D_{s'} = t''] \right) \right),$$

and

$$\mathcal{U}_v^{\mathcal{P}}(t|r) = \sum_{t'=1}^{T-t} \mathcal{U}_{d_r}^{\mathcal{P}}(t + t') P[T_r = t'].$$

Before discussing how to find policies with high expected utility, let us call attention to a peculiar feature of arrival time dependent travel plans. By definition, such a policy solely depends on the arrival time to a given stop and so does not condition on the time until the deadline. However, there might be situations when it is worth reconsidering and changing the set of suggested services according to the amount of time spent at the given stop. Such routing policies, called *current time dependent policies*, are also of interest and are considered in [5]. A significant difference between the two approaches is that *optimality* can be defined for arrival time dependent policies (see Section 3), while an analogue concept does not work for the current time dependent case as an optimal policy may not exist at all. The results of the present paper, including the existence of an optimal policy, are meant when the space of policies is restricted to the arrival time dependent ones.

3. Optimal policies

A policy is called *optimal* if $\mathcal{U}_v^{\mathcal{P}}(t) \geq \mathcal{U}_v^{\mathcal{P}'}(t)$ for each $\mathcal{P}' \in \mathfrak{P}$, $v \in V$ and $t \in [T]$. The notion of an optimal policy is demonstrated by the following example.

Example 1. Assume that the network consists of two nodes v and w and three services s_1, s_2 and s_3 going form v to w. The services depart from v as follows: s_1 departs in intervals I_1 and I_2 with probability $\frac{1}{2}$ and induces utility 3, s_2 departs in interval I_3 with probability 1 and induces utility 2, and s_3 departs in intervals I_2 and I_3 with probability $\frac{1}{2}$ and induces utility 1. Not arriving to stop w at all has utility 0.

In this very simple case it is possible to guess the optimal policy. Note that it makes sense to include a service in the set of recommended ones for a given interval even if the service in question departs with probability 0 as a policy do not change after the passenger's arrival to the stop.

In I_3, the optimal policy consists of the single service s_2 as it arrives with probability 1 and is clearly better than s_3. For interval I_2, the optimal policy contains s_1 and s_2. Indeed, taking only s_1 would not be enough as it may have already left in I_1. Finally, for I_1 the optimal policy consists of the single service s_1 as it arrives in I_1 or I_2 with probability 1 and has the highest induced utility.

By definition, the set of recommended services depends only on the arrival time to a given stop. That is, the induced utility does not depend on the policy at the same stop in later time instances. This suggests that an optimal policy may always exist. Beside showing that this impression is indeed true, the proof of the next theorem also provides a recurrence relation between the optimal policies at different times.

Theorem 2. *There always exists an optimal policy.*

Proof. Recall that the length of a time interval is a strictly positive lower bound for the travel times. Let $t \in [T]$ and assume that $\mathcal{P}$ is a partial policy which is already defined for all $t' > t$ optimally, that is, $\mathcal{U}_v^{\mathcal{P}}(t') \geq \mathcal{U}_v^{\mathcal{P}'}(t')$ for each $\mathcal{P}' \in \mathfrak{P}$, $v \in V$ and $t' > t$. Take an arbitrary stop $v \in V$ and a subset $X \subseteq S_v$ of services at v. We define a new partial policy $\mathcal{P}^X$ as follows:

$$\mathcal{P}_w^X(t') = \begin{cases} \mathcal{P}_w(t') & \text{if } t' > t, \\ X & \text{if } t' = t \text{ and } w = v. \end{cases}$$

Let $\mathcal{P}_v(t) = \mathcal{P}_v^{X_{opt}}(t)$ for $X_{opt} = \text{argmax}_{X \subseteq S_v} \{\mathcal{U}_v^{\mathcal{P}^X}(t)\}$. By applying the same procedure to all $v \in V$, we get an extension of the original policy which is optimal for $t' \geq t$.

The optimal policy is always empty at the destination node. As the utility function is $-\infty$ after a certain deadline T, the induced utility is $-\infty$ in time T for each node v different from the destination node and for each possible policy at v. Hence an optimal policy can be determined recursively by going back in time as described above. ☐

We note that in order to determine an optimal policy $\mathcal{P}_v(t)$ for a given node v and time t, we had to compute $\mathcal{U}_v^{\mathcal{P}^X}(t)$ for every subset X of S_v, hence the proof of Theorem 2 does not provide a polynomial-time algorithm. In fact we will show that finding an optimal policy is NP-hard. A similar result for the recourse model for stochastic shortest paths and least expected arrival time utility was given by [24]. We prove NP-hardness for the continuous time case by extending the approach of Nonner [23].

Theorem 3. *Given a monotone non-increasing utility function at the destination node, the problem of determining an optimal policy is NP-hard.*

Proof. We prove via reduction from EXACTCOVERBY3SETS (X3C): given a 3-uniform hypergraph $H = (V, \mathcal{E})$, decide if there exists a subset $\mathcal{E}' \subset \mathcal{E}$ of hyperedges such that each node $v \in V$ is contained in exactly one of them. Such a subset $\mathcal{E}'$ is called a perfect matching of the hypergraph. It is known that finding a perfect matching in 3-uniform hypergraphs is NP-hard [16, SP2].

Given an instance $H = (V, \mathcal{E})$ of X3C with $|V| = n$ and $|\mathcal{E}| = m$, we define a network and a utility function such that an optimal policy in the network corresponds to a perfect matching of H. Let $V = \{v_1, \ldots, v_n\}$ and $\mathcal{E} = \{e_1, \ldots, e_m\}$. We note that n is necessarily a multiple of 3.

Consider a network consisting of 2 stops v, w and $m + 1$ services going from v to w, denoted by $s_0, s_1, \ldots, s_m$. Let $0 < p, q < 1$ be real numbers satisfying $0 < p + q < 1$. The exact values of p and q will be defined later on. In time interval $[0, 1)$, all s_i arrive uniformly at random with probability $0 < p < 1$. In time interval $[1, 2)$, all s_i with $i \geq 1$ arrive uniformly at random with probability $0 < q < 1$. Finally, in time interval $[2, 3]$, services $s_1, \ldots, s_m$ arrive according to the construction of [23] where arriving probabilities are multiplied by $0 < 1 - p - q < 1$. For sake of completeness, we give a description of Nonner's construction.

Let $r = 1/n$ and define $t_i := 1/(1 - r)^{\lfloor \frac{i}{2} \rfloor \frac{n}{3}}$ for $i = 1, \ldots, 2n - 1$. We set $\bar{t}_1 := 0$ and $\bar{t}_i := \sum_{j=1}^{i-1} t_j$ for $i = 2, \ldots, 2n$. For $v_i \in V$ and $e_j \in \mathcal{E}$, define

$$p_j^{2i-1} := \begin{cases} r & \text{if } v_i \in e_j, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$p_j^{2i} := \begin{cases} 0 & \text{if } v_i \in e_j, \\ r & \text{otherwise.} \end{cases}$$

For each hyperedge $e_j \in \mathcal{E}$, we define a random variable ξ_j with $P[\xi_j = \bar{t}_i] := p_j^i$ for $i = 1, \ldots, 2n$. These random variables are well-defined as $\sum_{i=1}^{2n} p_j^i = 1$. Now we are in the position to define the arrival times in interval $[2, 3]$: service s_j arrives at time $2 + \bar{t}_i$ with probability $(1 - p - q)p_j^i$ for $j = 1, \ldots, m$ and $i = 1, \ldots, 2n$.

We note that each service arrives in time interval $[0, 3]$ with probability 1 except s_0 which either arrives in $[0, 1)$ or do not arrive at all. Now we turn to the definition of the travel times and the utility function. The travel times are deterministic, and that taking s_0 in $[0, 1)$ has utility 1, taking s_i in $[0, 1)$ has utility -1, taking s_i in $[1, 2)$ has utility $-M \leq -1$ for $i = 1, \ldots, m$, and taking any of the s_i's at time t in time interval $[2, 3]$ has utility $-M - t$.

Let $\mathcal{P}^{opt}$ be an optimal policy. Then $\mathcal{P}_v^{opt}(0)$ certainly contains s_0 as it is the only service with positive utility. However, $\mathcal{P}_v^{opt}(0)$ cannot consist of the single service s_0 as otherwise the induced utility would be $-\infty$ due to the fact that s_0 may not arrive at all. The following claim was proved in [23].

Claim 4. *Let $\mathcal{P}$ be a policy maximizing $\mathcal{U}_v^{\mathcal{P}}(2)$ under the restriction $|\mathcal{P}_v(2)| \leq n/3$. Then $\mathcal{P}_v(2)$ corresponds to a solution of X3C.*

As services $s_1, \ldots, s_n$ play exactly the same role in time interval $[0, 2)$, the claim implies the following: if we could choose the parameters M, p and q in such a way that $\mathcal{P}_v^{opt}(0)$ consists of exactly $n/3 + 1$ services, then $\mathcal{P}_v^{opt}(0)$ in fact contains s_0 and a subset of $\{s_1, \ldots, s_m\}$ corresponding to a perfect matching of H, showing that our problem is also hard. For any integer $1 \leq k \leq m$, let

$$T_k = \min_{\substack{\sigma \subseteq \{s_1, \ldots, s_m\} \\ |\sigma| = k}} \mathbb{E}[\min_{i \in \sigma} D_{s_i} | D_{s_i} \geq 2 \, \forall i = 1, \ldots, m].$$

Assume for a moment that $\mathcal{P}_v^{opt}(0)$ consists of $k + 1$ services. Recall that one of these services is s_0. Then the induced utility $\mathcal{U}_v^{\mathcal{P}^{opt}}(0)$ can be computed as follows:

$$\begin{aligned} \mathcal{U}_v^{\mathcal{P}^{opt}}(0) = &\frac{1-k}{1+k}(1 - (1 - p)^{k+1}) \\ &- M(1 - p)((1 - p)^k - (1 - p - q)^k) \\ &- (M + T_k)(1 - p)(1 - p - q)^k. \end{aligned} \tag{1}$$

That is, in order to ensure that $\mathcal{P}_v^{opt}(0)$ consists of exactly $n/3+1$ services, it suffices to find M, p and q such that the maximum of (1) as a function of k is attained at $k = \frac{n}{3}$. As each s_i ($i = 1, \ldots, m$) arrives in time interval $[0, 3]$ with probability 1, $T_k \leq 3$ for every $1 \leq k \leq m$. Fix $p = q = \frac{1-\varepsilon}{2}$. For a sufficiently small ε, we have

$$\mathcal{U}_v^{\mathcal{P}^{opt}}(0) \approx \frac{1-k}{1+k}(1 - \frac{1}{2^{k+1}}) - M\frac{1}{2^{k+1}}. \tag{2}$$

Let $f(k) := \frac{2^{k+2} + k^2 \ln 2 - \ln 2 - 2}{k^2 \ln 2 + 2k \ln 2 + \ln 2}$. The derivative of (2) is zero for $M = f(k)$. That is, for $M = f(n/3)$ the maximum of (2) – and so the maximum of (1)– is attained in $k = n/3$ as required, thus concluding the proof. $\square$

```
Procedure OPTIMAL-k-POLICY
    P_v(t) := ∅ for each v ∈ V, t ∈ [T]
    U^P_{v_d}(t) := u(t) for each t ∈ [T]
    t := T
    while t ≥ 0 do
        for v ∈ V, v ≠ v_d do
            for X ⊆ S_v, |X| ≤ k do
                for w ∈ V, t < t' ≤ T do
                  |  P^X_w(t') := P_w(t')
                end
                P^X_v(t) := X
                Compute U^{P^X}_v(t)
            end
            P_v(t) := arg max_X{U^{P^X}_v(t)}
        end
        t := t − 1;
    end
end
```

Fig. 1. Determining an optimal k-policy.

4. Determining an optimal k-policy

By Theorem 3, finding an optimal policy is NP-hard. Note that the proof of Theorem 2 does not provide an efficient algorithm for finding an optimal policy as for a fixed node v and time t it requires to compute $\mathcal{U}^{\mathcal{P}^X}_v(t)$ for every subset X of S_v.

However, if the number of services that a policy may contain for each node–time pair is bounded by a constant k, then the problem becomes tractable. We call a policy a *k-policy* if $|\mathcal{P}_v(t)| \leq k$ for each $v \in V, t \in [T]$. The set of k-policies is denoted by $\mathfrak{P}_k$. A k-policy $\mathcal{P}$ is called *optimal* if it is optimal among $\mathfrak{P}_k$, that is, $\mathcal{U}^{\mathcal{P}}_v(t) \geq \mathcal{U}^{\mathcal{P}'}_v(t)$ for each $\mathcal{P}' \in \mathfrak{P}_k$, $v \in V$ and $t \in \mathbb{R}$. Let $\mathcal{P}^k_v(t)$ denote the optimal k-policy at stop $v \in V$ and time $t \in [T]$. The proof of Theorem 2 sheds light on the following recurrence relation between the optimal policy values:

$$\mathcal{P}^k_v(t) = \underset{\substack{X \subseteq S_v \\ |X| \leq k}}{\mathrm{argmax}}\{\mathcal{U}^{\mathcal{P}^X}_v(t)\} \tag{3}$$

where

$$\mathcal{P}^X_w(t') = \begin{cases} \mathcal{P}^k_v(t') & \text{if } t' > t, \\ X & \text{if } t' = t \text{ and } w = v. \end{cases}$$

Hence we get the following recurrence relation between the optimal utility function values:

$$\mathcal{U}^{\mathcal{P}^k}_v(t) = \max_{\substack{X \subseteq S_v \\ |X| \leq k}} \sum_{t'=t}^{T} \sum_{s \in X} \left(\mathcal{U}^{\mathcal{P}^k}_v(t'|s)P[D_s = t'] \prod_{\substack{s' \in X \\ s' \neq s}} \left(1 - \sum_{t''=t}^{t'-1} P[D_{s'} = t''] \right) \right).$$

The above recurrence relations make the usage of a dynamic programming approach possible. Starting from the end of the time horizon, one can go back step-by-step in time to determine an optimal k-policy, thus giving a polynomial-time dynamic programming algorithm, called OPTIMAL-k-POLICY, for determining an optimal k-policy.

Initially, set $\mathcal{P}_v(t) = \emptyset$ for each $v \in V$ and $t \in [T]$, $\mathcal{U}^{\mathcal{P}}_v(t) = -\infty$ for $v \neq v_d$ and $\mathcal{U}^{\mathcal{P}}_{v_d}(t) = u(t)$, where $u(t)$ is the utility function given by the passenger. At a general stage of the algorithm $\mathcal{P}_v(t')$ is already defined for each $v \in V$ and $t' > t$, and (3) is used to determining the policy for every stop $v \in V$ in time t, see Fig. 1.

Recall that the maximum number of services using the same stop was denoted by Δ. A very rough estimation shows that the running time of the algorithm is $O(|V| \cdot T^3 \cdot \Delta^k \cdot k^2)$. If k is constant, this means to have a polynomial-time algorithm for computing an optimal policy.

5. Computational case study

In this section we compare the proposed routing method with the deterministic algorithm. The computations are based on the General Transit Feed Specification (GTFS) data of the Budapest public transport Network [7]. GTFS is a common format for public transport schedules and provides for trip planning functionality, see [17]. The algorithms are implemented in C++ with heavy use of the open source LEMON graph optimization library (see [1]). The code is platform independent, the actual tests were performed on a Linux based server with a 3.50 GHz Intel Core i7-5930K CPU and 32 GB RAM.

5.1. Extending the model

For the description and analysis of the algorithm above, we have assumed a simplified model where the services were given as simple trips between two nodes without intermediate stops. For the computational case study, we use a more complex and realistic model which, according to the GTFS data format, considers services with several trips (following each other in a prescribed order). A service has several trips consisting of several hops, hence the definition of policy must be modified accordingly. In this case, a policy is a function $\mathcal{P} : V \times [T] \to 2^{S \times V}$ such that if $(s, w) \in \mathcal{P}_v(t)$ then

- $s \in S_v$,
- $w \neq v$ is a stop of service s which is after v,
- $(s, w') \notin \mathcal{P}_v(t)$ for each stop $w' \neq w$ of service s.

In the extended model, $(s, w) \in \mathcal{P}_v(t_0)$ means the following: if a passenger arrives to stop v at time t_0 then she is suggested to take service s whenever it arrives and to get off from it at stop w.

5.2. Data preprocessing and generation of the stochastic model

In order to prepare the necessary input data, the raw GTFS data is preprocessed in several steps. In a first step, irrelevant services are discarded so that only services working on the date of travel are considered. In a second step the list of services is filtered according to the given passenger preferences, only keeping those services that theoretically may play role in the journey. Simultaneously, a deterministic shortest path is computed between the start and the destination node as a reference solution.

Next, a probabilistic model is set up for determining arrival, departure and travel time distributions. The model may be based on the assumption of a simplified distribution (e.g., Gamma distribution for travel times), but can also take into account GTFS real-time data and statistics obtained from real-life measurements.

5.3. Policy generation

As a starting step, an optimal deterministic solution is computed which has the highest utility in the time period defined by the user and has the latest departure time from the starting node. This is later used as a reference solution by the method to find a proper policy. Then, a dynamic programming procedure is used to determine the induced utility of being at stop v at time t for each node and time. The length of a time interval was set to 20 s.

In the main part of the procedure, a queue of 'active' nodes is maintained, which is initialized with the destination node v_d. At the beginning, only v_d is marked as active. In the following iterations, assume that policies and induced utilities are computed for all $v \in V$ and $t' > t$. The top node v is removed from the queue, and for each service $s \in S_v$, time $t \in [T]$ and stop w preceding v on the route of s we update $\mathcal{U}_w^{\mathcal{P}}(t)$, $\mathcal{U}_w^{\mathcal{P}}(t \mid s)$ according to OPTIMAL-k-POLICY, and also change $\mathcal{P}_w(t)$ if necessary. If an improvement is found for some triple (s, t, w), then stop w is pushed to the end of the queue and is marked as 'active'. This Bellman–Ford-like algorithm computes the induced utilities for each stop and each time. The algorithm terminates when no improvement is found or an upper bound on the number of iterations is reached.

As the result is not a single route, an important question is how to represent the obtained policy. First of all, it is unnecessary to show the policy at each node, as this is only interesting for the starting node and nodes that are possibly visited during the journey. Moreover, to decrease the amount of data written out, $\mathcal{P}_v(t)$ is only displayed if it is not identical to $\mathcal{P}_v(t-1)$. The output of the model – which is a policy for each node – is further post-processed as the presence of multiple choices may result in complicated route plans. For example in case of several equally good stops it may be unclear where to take alight from a given service. The deterministic solution computed earlier is used to break ties based on latest departure times.

5.4. Test instance and results

The above procedure was applied to the GTFS data of the Budapest public transport network. This timetable consists of 5290 stops and 135.959 services. It is not straightforward how to measure the efficiency of a policy in a stochastic environment. As the introduction of policies versus simple routes was motivated by the lack of reliability of deterministic travel plans, we measure the performance of a policy by comparing it to the usual deterministic approach. We chose an arbitrary stop v as a destination node from the map of Budapest uniformly at random, and determined the policy for each stop different from v for a given date and time. We considered the SOTA (Stochastic On Time Arrival) problem in all cases, that is, the utility function at v was defined to be 1 up to a certain deadline t_{end} until which the passenger wishes to arrive to the destination node v, and 0 afterwards. The policies thus obtained were tested by running 500 simulations for each node.

In each instance of the simulation, the departure and travel times were assumed to follow normal distributions with expected value equal to the deterministic departure and travel times of the corresponding service, respectively. For each service, the deviations of its travel time distributions were gradually increased while following the route of the service in order to model that uncertainties between stops add up. Deviations were also changed according to time of day (light or
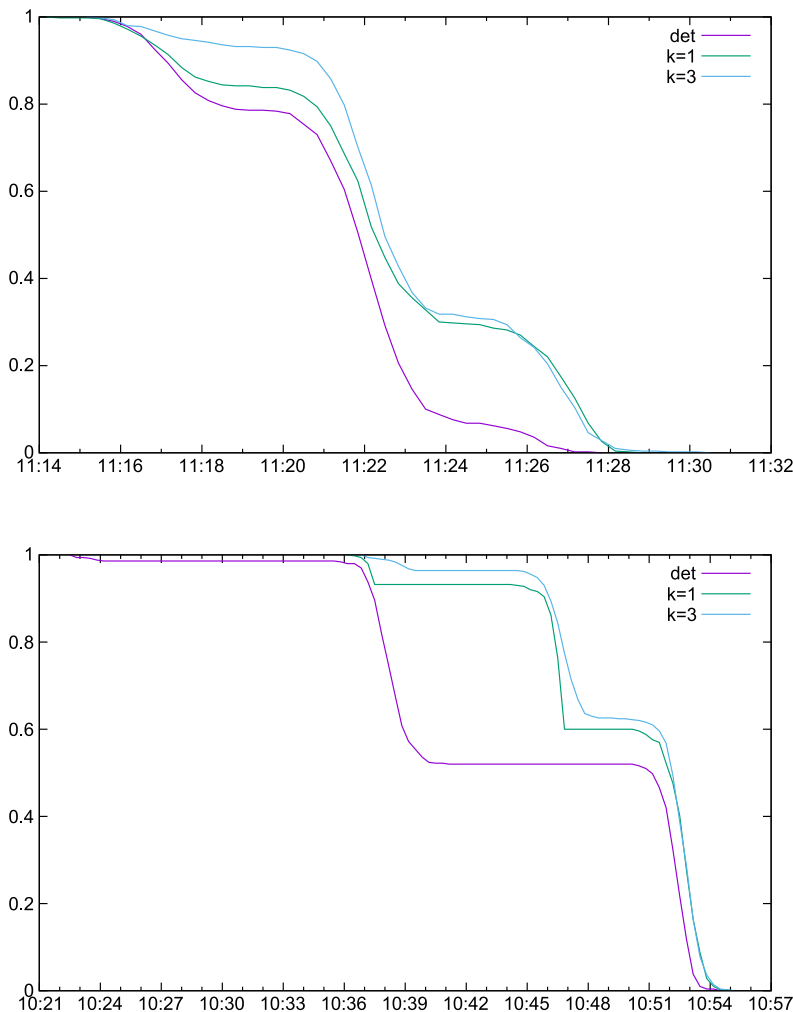
Fig. 2. Probability of arriving on time vs. departure time.

heavy traffic) and type of the service (bus, tram, subway). This method is fast and is based on reasonable assumptions on the nature of traffic, but the proposed method could be also used with historical data.

For each node w different from v and each time $t \in [T]$, we checked whether a passenger leaving from node w at time t and following either the deterministic solution or the policy succeeds to get to v in time. In both cases, for each node w we computed the latest time instance when a user has to leave from w in order to arrive to v in time with probability at least 90%. These values can be compared to each other for the two solutions, hence giving a benchmark for evaluating their effectiveness. In addition, it enables us to introduce the notion of *safety margin* which tells us how much earlier we should start compared to the latest deterministic departure so that we achieve a 90% probability of arriving in time.

Fig. 2 provides illustrative examples of the achievable probability of arriving on time for different departure times from a fixed stop to a destination node. The results show that the usage of policies significantly increases the probability of arriving on time, which may be particularly useful at late departure times.

Fig. 3 compares the deterministic solution and the policy for a given destination node v. Each dot on the scatter plot represents a starting node w different from v. The x coordinate of a dot is the time when a passenger has to leave from w in order to arrive to v in time with probability at least 90% when following the route of the policy, while its y coordinate is defined similarly based on the deterministic solution. That is, dots above the line $y = x$ corresponds to stops from where one may leave later when following the induced policy instead of the deterministic route.

Fig. 4 shows the average safety margins as a function of the deterministic travel time. The average is taken from computing the policies from all possible start positions to 4 representative destinations chosen at different parts of the city.

As we are considering k-policies, the maximum number k of services that the policy may contain for a given time is a parameter that has a significant effect on the efficiency of the solution. Although larger values of k result in better policies, it also leads to more complicated travel plans, which might discourage passengers from using the system in real applications.
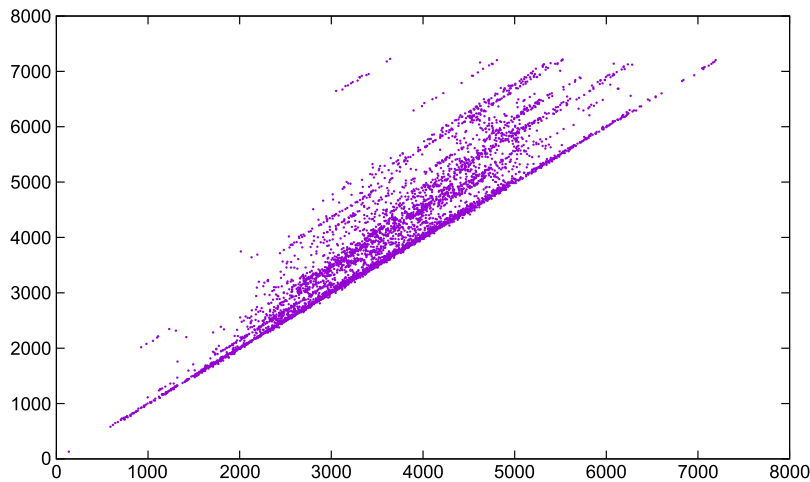
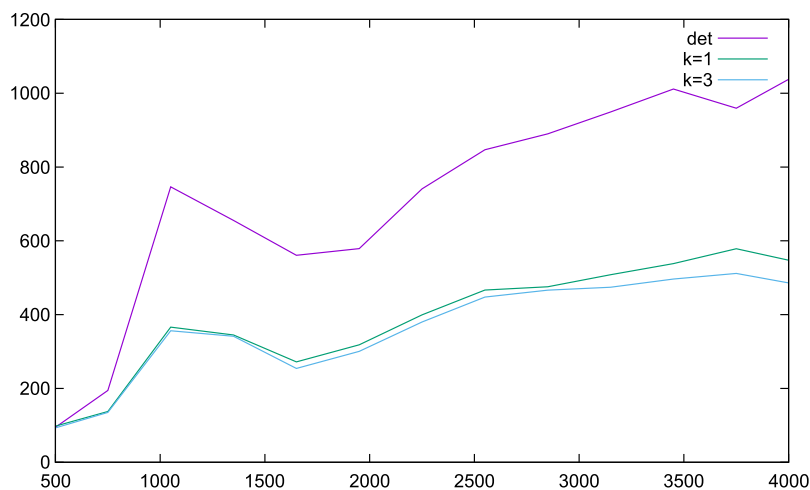**Fig. 3.** Policy vs. deterministic route.

Fig. 4. Average safety margin vs. deterministic travel time.

However, our simulations suggest that moderate values of k already lead to substantial improvements while increasing k beyond 3 leads to negligible marginal improvement. Thus, for the considered public transport system of Budapest, choosing a value of $k = 3$ seems to be a reasonable compromise between solution quality and policy complexity, and we expect similar settings to perform well in other public transport networks as well.

6. Conclusion

We presented a public transport journey planning approach to cope with the typically non-deterministic public transport environment. The overall goal was to find travel policies that maximize expected utility.

At each location, a policy can encode multiple service choices, from which the traveller should take depending on which one will be arriving first. Instead of deterministic travel and arrival/departure times our model uses probability distributions, which result in a more robust, and on average more effective journey plan. Our approach can handle any passenger-defined monotone non-increasing utility functions, which allow to encode sophisticated passenger preferences beyond just 'arriving in time' or 'as fast as possible'.

Acknowledgements

References

[1] D. Balázs, A. Jüttner, P. Kovács, LEMON–an open source C++ graph template library, Electron. Notes Theor. Comput. Sci. 264 (5) (2011) 23–45. http://lemon.cs.elte.hu.

[2] J.L. Bander, C.C. White, A heuristic search approach for a nonstationary stochastic shortest path problem with terminal cost, Transp. Sci. 36 (2) (2002) 218–230.

[3] R. Bellman, On a routing problem, Technical report, DTIC Document.

[4] K. Bérczi, A. Jüttner, M. Korom, M. Laumanns, T. Nonner, J. Szabó, Stochastic route planning in public transport, U.S. Patent No US9482542 B2, Publication date Nov 1, 2016.

[5] K. Bérczi, A. Jüttner, M. Laumanns, J. Szabó, Stochastic route planning in public transport, Transp. Res. Proc. 27 (2017) 1080–1087.

[6] A. Botea, E. Nikolova, M. Berlingerio, Multi-modal journey planning in the presence of uncertainty, in: Proceedings of the International Conference on Automated Planning and Scheduling ICAPS-13, 2013, pp. 20–28.

[7] Budapest Közlekedési Központ, Gtfs developer info, http://www.bkk.hu/en/developers/.

[8] J. Dibbelt, B. Strasser, D. Wagner, Delay-robust journeys in timetable networks with minimum expected arrival time, in: Stefan Funke, Matúš Mihalák (Eds.), 14th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, in: OpenAccess Series in Informatics (OASIcs), vol. 42, 2014, pp. 1–14.

[9] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.

[10] Y. Fan, R.E. Kalaba, J.E. Moore II, Arriving on time, J. Optim. Theory Appl. 127 (3) (2005) 497–513.

[11] Y. Fan, Y. Nie, Arriving-on-time problem: discrete algorithm that ensures convergence, Transp. Res. Record: J. Transp. Res. Board (1964) 193–200.

[12] Y. Fan, Y. Nie, Optimal routing for maximizing the travel time reliability, Netw. Spat. Econ. 6 (3–4) (2006) 333–344.

[13] R.W. Floyd, Algorithm 97: shortest path, Commun. ACM 5 (6) (1962) 345.

[14] L.R. Ford, Network flow theory, Technical report, DTIC Document.

[15] H. Frank, Shortest paths in probabilistic graphs, Oper. Res. 17 (4) (1969) 583–599.

[16] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide To the Theory of NP-Completeness, 1979.

[17] GTFS, What is gtfs? https://developers.google.com/transit/gtfs/. (Accessed 6 March 2015).

[18] R.W. Hall, The fastest path through a network with random time-dependent travel times, Transp. Sci. 20 (3) (1986) 182–188.

[19] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107.

[20] D. Hoy, E. Nikolova, Approximately optimal risk-averse routing policies via adaptive discretization, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 3533–3539.

[21] R.P. Loui, Optimal paths in graphs with stochastic or multidimensional weights, Commun. ACM 26 (9) (1983) 670–676.

[22] I. Murthy, S. Sarkar, Exact algorithms for the stochastic shortest path problem with a decreasing deadline utility function, European J. Oper. Res. 103 (1) (1997) 209–229.

[23] T. Nonner, Polynomial-time approximation schemes for shortest path with alternatives, in: Proceedings of the European Symposium on Algorithms ESA-12, 2012, pp. 755–765.

[24] G.H. Polychronopoulos, J.N. Tsitsiklis, Stochastic shortest path problems with recourse, Networks 27 (2) (1996) 133–143.

[25] S. Samaranayake, S. Blandin, A. Bayen, A tractable class of algorithms for reliable routing in stochastic networks, Transp. Res. C 20 (1) (2012) 199–217.

[26] S. Samaranayake, S. Blandin, A. Bayen, D. Delling, L. Liberti, Speedup techniques for the stochastic on-time arrival problem, in: 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, in: OpenAccess Series in Informatics (OASIcs), vol. 25, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 83–96.

[27] L. Yang, X. Zhou, Constraint reformulation and a Lagrangian relaxation-based solution algorithm for least expected time path problem, Transp. Res. B 59 (2014) 22–44.