Project Number 288094

eCOMPASS

eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

STREP
Funded by EC, INFSO-G4(ICT for Transport) under FP7

eCOMPASS – TR – 057

Robust Routing in Urban Public Transportation: Evaluating Strategies that Learn From the Past

Kateina Bhmov, Mat Mihalk, Peggy Neubert, Tobias Prger

November 2014

Robust Routing in Urban Public Transportation: Evaluating Strategies that Learn From the Past

Kateřina Böhmová[1], Matúš Mihalák[1], Peggy Neubert[2], and Tobias Pröger[1]

[1] Department of Computer Science, ETH Zürich, Zürich, Switzerland
[2] Verkehrsbetriebe Zürich, Zürich, Switzerland

Abstract. Given an urban public transportation network and historic delay information, we consider the problem of computing reliable journeys. We propose new algorithms based on our recently presented solution concept (Böhmová et al., ATMOS 2013), and perform an experimental evaluation using real-world delay data from Zürich, Switzerland. We compare these methods to natural approaches as well as to our recently proposed method which can also be used to measure typicality of past observations. Moreover, we demonstrate how this measure relates to the predictive quality of the individual methods. In particular, if the past observations are typical, then the learning-based methods are able to produce solutions that perform well on typical days, even in the presence of strong delays.

1 Introduction

When using public transportation to travel from a stop s to a stop t, we may want to arrive at t no later than at time t_A. Determining the right moment to leave s is nontrivial: We want to reach t at time t_A at the latest, but we don't want to leave s much too early. In an ideal situation, every bus and every tram is on time, and it is sufficient to compute a journey that is planned to leave s as late as possible but still reaches t at the latest at t_A. However, in reality, traffic can be congested and we should expect delays. Thus, we are looking for a *robust* journey from s to t that arrives before time t_A, but still leaves s at a "reasonable" time. Notice that one may have additional preferences, such as low travel costs, which we don't consider for the sake of simplicity.

Many approaches to find a fastest journey in a given public transportation network were considered in the literature, see e.g., a recent survey by Bast et al. [1]. However, it follows from an empirical study performed by Firmani et al. [9] on the transportation network of Rome that the timetable information and real movement of the vehicles (based on GPS data) are only mildly correlated. Thus, routing based solely on a scheduled timetable without considering the occurrence of delays may lead to solutions of quite an unsatisfactory quality.

One approach to account for delays is using stochastic methods—the delays are typically modeled as random variables on the edges of the network [4, 10, 15], or on each vehicle [6, 7]. For a given fixed timetable, Disser et al. [8]

extended Dijkstra's algorithm for computing pareto-optimal multi-criteria journeys. Müller-Hannemann and Schnee [14] used a *dependency graph* to predict secondary delays caused by some current primary delays and gave routing strategy with respect to these delays. Bast et al. [2] studied the robustness of transfer patterns in the presence of delays. They argue that even when delays occur, a reasonably good path is still included in the pattern. Dibbelt et al. [7] modeled the delays using stochasticity and computed a *decision graph* with all the possibly relevant nodes and vehicles instead of a single path. Goerigk et al. [12] assumed that a set of delay *scenarios* is provided, and showed how to compute a journey that arrives on time in every scenario (strict robustness) or a journey with fewest number of unreliable transfers having an almost optimal travel time (light robustness). Goerigk et al. [11] considered journeys, within the setting of delay scenarios, that can be updated if delays occur (recoverable robustness).

In [3], we introduced a different approach for finding robust journeys that uses recorded observations from the past as input—we look for journeys that performed well in the given past observations. Since this approach requires journeys to be comparable in different past days, classical solutions concepts, such as a path in the time-expanded or the time-dependant graph, are not suitable.

In the present paper we shortly describe our solution concept and several methods for finding robust journeys, and perform an extensive experimental study to evaluate these methods and to study different aspects related to robust routing.

2 Model

Network Design. Let $\mathcal{S}$ be a set of stops. A *line* is an ordered sequence $\langle v_1, \dots, v_k \rangle$ of stops from $\mathcal{S}$, where v_i is visited directly before v_{i+1}. We explicitly distinguish two lines with the same stops but opposite directions. A sequence of lines $\langle l_1, \dots, l_\beta \rangle$ with $l_i \neq l_{i+1}$ is called an *st-route* if there exist $\beta + 1$ stops $v_0 := s, v_1, \dots, v_{\beta-1}, v_\beta := t$ where both v_{i-1} and v_i are stops on the line l_i, and the line l_i visits v_{i-1} (not necessarily directly) before v_i. We say that a *transfer* between the lines l_i and l_{i+1} occurs at v_i. Notice that there might be more than one possible transfer between two lines. For two stops $s, t \in \mathcal{S}$ and an integer $\beta \in \mathbb{N}$, let $\mathcal{R}_{st}^{\beta}$ denote the set of all *st*-routes with at most $\beta - 1$ transfers.

Trips and Timetables. While the only information associated with a line itself are its consecutive stops, it usually is operated multiple times per day. Each of these concrete realizations is called a *trip*. A *timetable* stores for every stop $v \in \mathcal{S}$ the arrival and departure times of every trip over a day. We have

1. a *planned* timetable T_{plan} which we assume to be periodic, i.e., every line realized by some trip τ will be realized by a later trip τ' again (not necessarily on the same day).
2. a set $\mathcal{T}$ of *recorded* timetables T_i that describe how various lines were operated during a given time period (e.g., on a concrete day). These recorded timetables are concrete executions of the planned timetable.

In the following, *timetable* refers both to the planned as well as to a recorded timetable. We assume that timetables respect the FIFO property, i.e. two buses or trams *of the same line* do not overtake each other.

Goal. Let $s, t \in \mathcal{S}$ be the departure and the target stop, and $\beta \in \mathbb{N}_0$ be the maximum number of allowed transfers. A *journey* consists of a departure time t_D, a route $\langle l_1, \ldots, l_\alpha \rangle \in \mathcal{R}_{st}^\alpha$ with $\alpha \leq \beta - 1$, and a sequence of transfer stops $\langle v_1, \ldots, v_{\alpha-1} \rangle$. Its intuitive interpretation is to leave stop s at time t_D, take the first arriving (trip of) line l_1, and for every $i \in \{1, \ldots, \alpha - 1\}$, leave l_i at stop v_i and immediately take the next arriving trip of line l_{i+1}. Our goal is to compute a recommendation in form of one or more (robust) journeys from s to t that will likely arrive on time (i.e., at time t_A or earlier) on a day for which the concrete travel times are not known yet.

3 Robustness

Overview. In this section we present some approaches for computing robust journeys. We note that, given a route r and a parameter $\gamma \in \mathbb{N}$, we can use the planned timetable T_{plan} to find a journey j along r that leaves s as late as possible, but not later than time $t_A - \gamma$. Thus, as soon as an algorithm identifies both route r and a parameter γ, it can also reconstruct a corresponding journey.

Transfer Buffers. An naïve strategy to increase reliability of a journey is to enforce an additional buffer time at each transfer or at the end of the trip. The Buffer-ξ approach uses T_{plan} to compute a journey that is planned to leave s as late as possible, arrives at t not later than time t_A, and that has an additional time of at least ξ at each transfer of the journey. This especially implies that if a line l_i is planned to arrive at a transfer stop v_i at time t_i, then the next line l_{i+1} of the journey can only be taken at time $t_i + \xi$ or later.

A Similarity-Based Approach. In [3], we described how a general approach to robust optimization designed by Buhmann et al. [5] can be applied for computing robust journeys. This section briefly recalls our method. Let $T \in \mathcal{T}$ be a timetable and $\gamma \in \mathbb{N}_0$. An *approximation set* $A_\gamma(T)$ contains all routes $r \in \mathcal{R}_{st}^\beta$ for which T contains a journey along r that leaves s at time $t_A - \gamma$ or later, and that arrives at t at time t_A or earlier. We assume that $A_\gamma(T)$ is a *multiset*: a route r is contained as often as it is realized by a journey starting at time $t_A - \gamma$ or later, and arriving at time t_A or earlier (see Figure 1 for an example). The parameter γ can be interpreted as the maximal time that we depart before t_A. If we consider approximation sets $A_\gamma(T_1), \ldots, A_\gamma(T_k)$ for the timetables $T_1, \ldots, T_k \in \mathcal{T}$, every approximation set contains only routes that are realized (by a journey) in the same time period $[t_A - \gamma, t_A]$, and that are therefore comparable among different approximation sets.

The approach in [3, 5] expects that exactly two timetables $T_1, T_2 \in \mathcal{T}$ are given. To compute a *robust* route when only two timetables are available, we

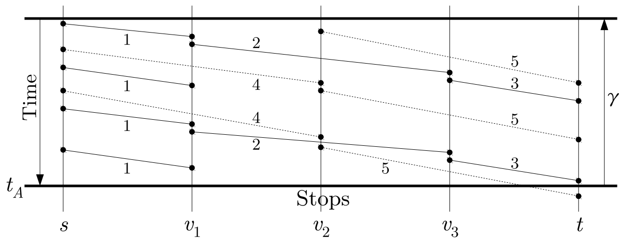


Fig. 1. A timetable with five lines $\{1, \dots, 5\}$ and two routes $r_1 = \langle 1, 2, 3\rangle$ (solid) and $r_2 = \langle 4, 5\rangle$ (dotted). The x-axis denotes the stops $\{s, v_1, v_2, v_3, t\}$, the y-axis the time. If a trip leaves a stop v_d at time t_d and arrives at a stop v_a at time t_a, it is indicated by a line segment from (v_d, t_d) to (v_a, t_a). $A_\gamma(T)$ contains r_1 three times and r_2 once.

consider $A_\gamma(T_1) \cap A_\gamma(T_2)$: the only chance to find a route that is likely to be good in the future is a route that was good in the past for *both* recorded timetables. The parameter γ determines the size of the intersection: if γ is too small, the intersection will be empty. If γ is too large, the intersection contains many (and maybe all) st-routes, and not all of them will be a good choice. Assuming that we knew the "optimal" parameter γ_{OPT}, we could pick a route from $A_{\gamma_{OPT}}(T_1) \cap A_{\gamma_{OPT}}(T_2)$. Buhmann et al. [5] suggest to set γ_{OPT} to the value γ that maximizes

$$S_\gamma = \frac{|\mathcal{R}_{st}^\beta||A_\gamma(T_1) \cap A_\gamma(T_2)|}{|A_\gamma(T_1)||A_\gamma(T_2)|}. \tag{1}$$

The value $S_{\gamma_{OPT}}$ measures how similar the timetables T_1 and T_2 are, so we call it the *similarity* of T_1 and T_2. The Similarity-Rand approach selects a route r from the intersection uniformly at random, while Similarity-MRR selects the most frequent route r from the intersection. For both approaches we recommend to depart at least γ_{OPT} units of time in advance. For more details, cf. [3, 5].

Function-Based Approaches. Let $T_i \in \mathcal{T}$ be a (recorded historic) timetable, $r = \langle l_1, \dots, l_\alpha\rangle \in \mathcal{R}_{st}^\alpha$ be a route, $\tau_1, \dots, \tau_k$ be the trips of line l_1 in T_i and $D(\tau_j, s)$ be the departure time of the trip τ_j at s. We define δ_i^r as

$$\min_{j \in [1,k]} \left\{ t_A - D(\tau_j, s) \;\middle|\; \begin{array}{l} \tau_j \text{ can be extended to a journey along } r \text{ that} \\ \text{arrives in } T_i \text{ at stop } t \text{ at time } t_A \text{ or earlier} \end{array} \right\}, \tag{2}$$

which intuitively can be interpreted as follows: to arrive on time using route r on the day at which T_i is realized, one has to leave s at least δ_i^r units of time before the latest allowed arrival time t_A. For a given function $f : (\mathbb{R}^+)^{|\mathcal{T}|} \to \mathbb{R}$, we search for a route $r \in \mathcal{R}_{st}^\alpha$ that minimizes $f(\delta_1^r, \dots, \delta_{|\mathcal{T}|}^r)$. In the following, we describe some possible choices for f, and we abbreviate $f(\delta_1^r, \dots, \delta_{|\mathcal{T}|}^r)$ by $f(r)$.

For a number $p \in [1, \infty]$, the Norm-p estimator has the objective function

$$f_{\|\cdot\|}^p(r) = \left\| \left(\delta_1^r, \dots, \delta_{|\mathcal{T}|}^r\right) \right\|_p . \tag{3}$$

It is easy to see that $f^1_{\|\cdot\|}$ selects all routes which in average (w.r.t. the recorded timetables in $\mathcal{T}$) depart as late as possible. Moreover, $f^\infty_{\|\cdot\|}$ selects all routes minimizing the maximum time between the departure and the latest allowed arrival time t_A. Such routes can alternatively be seen as routes maximizing the earliest departure time necessary to arrive on time in *all* timetables in $\mathcal{T}$. Thus, the Norm-∞ estimator is related to the similarity-based approach from the previous paragraph in the following way. Let $\gamma_{FI} = \min \left\{ \gamma > 0 \mid \bigcap_{i=1}^{|\mathcal{T}|} A_\gamma(T_i) \neq \emptyset \right\}$ be the smallest value for γ such that the intersection of all γ-approximation sets is non-empty. One can observe that every route r contained in $\bigcap_{i=1}^{|\mathcal{T}|} A_{\gamma_{FI}}(T_i)$ minimizes $f^\infty_{\|\cdot\|}$ and vice versa. We note that these methods relate to strict robustness [12], but are based on a different solution concept, and learn from past observations given as daily recorded timetables (instead of specifying a set of possible delays).

Now, let $p \in [1, \infty]$ be arbitrary and let r^p_j be a route minimizing $f^p_{\|\cdot\|}$. To determine how much in advance one has to depart when using r^p_j, we use our previous observations. For $p = 1$, it is reasonable to set $\gamma^p_j = f^1(r^p_j)/|\mathcal{T}|$ since $f^1_{\|\cdot\|}$ corresponds to averaging the departure times. For $p = \infty$, it is reasonable to set $\gamma^p_j = f^\infty(r^p_j)$. For every other $p \in (1, \infty)$, we simply scale the time linearly with respect to $p = 1$ and $p = \infty$. More concretely, we set

$$\gamma^p_j = f^\infty(r^p_j) - \left(\frac{f^p(r^p_j) - f^\infty(r^p_j)}{f^1(r^p_j) - f^\infty(r^p_j)} \right) \cdot \Big(f^\infty(r^p_j) - f^1(r^p_j)/|\mathcal{T}| \Big). \tag{4}$$

A different function-based estimator comes from the *mean-risk model* which was just recently used for finding robust routes in private transportation [13]. Let $c \in \mathbb{R}^+_0$ be the *risk-aversion coefficient*, where $c = 0$ corresponds to the situation where the risk is being completely ignored. The objective function associated with the Mean-Risk-c estimator is

$$f^c_{MR}(r) = \mathrm{Mean}\big(\delta^r_1, \ldots, \delta^r_{|\mathcal{T}|}\big) + c \cdot \sqrt{\mathrm{Variance}\big(\delta^r_1, \ldots, \delta^r_{|\mathcal{T}|}\big)}. \tag{5}$$

For a route r_j minimizing f^c_{MR}, we simply set $\gamma_j = f^c_{MR}(r_j)$ as the time one has to depart in advance. Notice that Mean-Risk-0 is equivalent to Norm-1.

4 Experimental Results

Experimental Setup. For an experimental evaluation of the methods proposed in Section 3 we used the tram and bus network of the city of Zürich, Switzerland, which has 401 stops and 292 lines. The recorded timetables $\mathcal{T} = \{T_1, \ldots, T_7\}$ were realized on seven consecutive Thursdays in the period from 4 April to 23 May 2013, ignoring 9 May (which was a public holiday and therefore had different traffic and a different planned timetable).

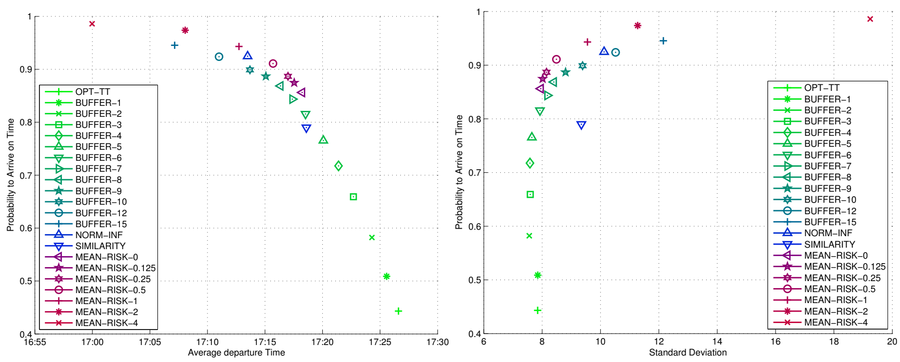
We observed that in reality many of the 292 lines have the same ID (such as, e.g., tram 6, bus 31, etc.). This is consequence of our modeling: not only do we distinguish lines travelling in opposite directions, but there are also special lines coming from or going to the depot, lines whose corresponding vehicle turns around in advance, and lines that do not visit certain stops in the evening. Since these special lines operate only on a low frequency and mostly only early in the morning or late in the evening, we ignored them and focused on the "standard" realizations. Hence we effectively used only 118 of the 292 lines.

For each of the following experiments, we generated 10,000 (30,000 for the experiments on the number of transfers) departure/target pairs $(s, t) \in \mathcal{S}^2$ with $s \neq t$ uniformly at random. For each such pair (s, t), we computed the smallest $\beta \in \mathbb{N}$ such that $\mathcal{R}_{st}^{\beta} \neq \emptyset$ and used this value for the maximum allowed number of transfers. Notice that we have $\beta = 1$ even if there exists a direct st-route with no transfers at all. In such a case, one might still prefer to take an alternative route with only one transfer, probably leading to a shorter travel time. After computing β and $\mathcal{R}_{st}^{\beta}$, we performed the corresponding experiment. We set the target arrival time t_A to 18:00 except for the experiments that study how the behavior of the methods changes during the day. Unless otherwise stated, the buffer methods used the planned timetable T_{plan} as input, the similarity-based methods used T_5 and T_6 (recorded on 2 May and 16 May), and the function-based methods used $T_1, \ldots, T_6$ (recorded between 4 April on 16 May). Timetable T_7 (recorded on 23 May) was used to assess the quality of the proposed journeys.

In our experiments we observed that the performance of Similarity-Rand and Similarity-MRR is nearly identical, so our figures show only the behavior of the latter variant, and for simplicity we refer to both variants by Similarity. Also, Norm-2 performs similarly to Norm-Inf, so our figures mostly omit Norm-2. Furthermore we observed that it rarely happened that a journey proposed by Buffer-ξ, Similarity, Norm-Inf or Mean-Risk-1 arrived much too early or much too late in the test instance. In all of these cases this was caused either because of a highly non-typical situation in the input or the test instance (e.g., an accident), or because a line was chosen that was not realized regularly (e.g., less than once per hour). Hence we ignored all pairs (s, t) for which at least one of the methods above computed a journey arriving more than one hour too early or too late.

The experiments were performed on one core of an Intel Core i5-3470 CPU clocked at 3.2 GHz with 4 GB of RAM running Debian Linux 7.8. For enumerating all st-routes in $\mathcal{R}_{st}^{\beta}$, we used the algorithm proposed in [3] which runs on average 35ms. After computing $\mathcal{R}_{st}^{\beta}$, the buffer strategies have an average running time 1ms or less, the similarity-based methods 8ms, and the function-based approaches 24ms. Notice that these running times are faster than the ones described in [3], because we used a smaller network (without the agglomeration).

Arrival Rate, Departure Time and Standard Deviation on the Arrival Time. Intuitively, an earlier departure time leads to a higher probability to arrive on time (arrival rate), and achieving a higher arrival rate in a network with delays entails a higher standard deviation on the arrival time. Figure 2 compares the proposed methods with respect to these aspects. It shows that, independently of the considered method, there is a clear trade-off between the departure time and the arrival rate (a) as well as between the standard deviation of the arrival time and the arrival rate (b).



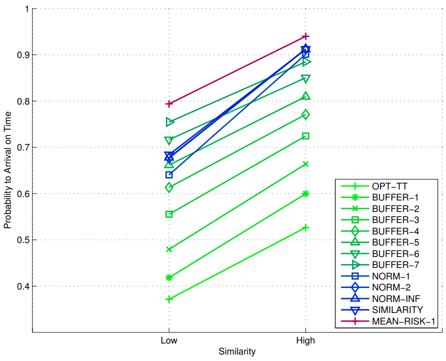
(a) (b)

Fig. 2. Comparison of various methods: arrival rate vs. average departure time (a), and arrival rate vs. standard deviation on the arrival time (b).

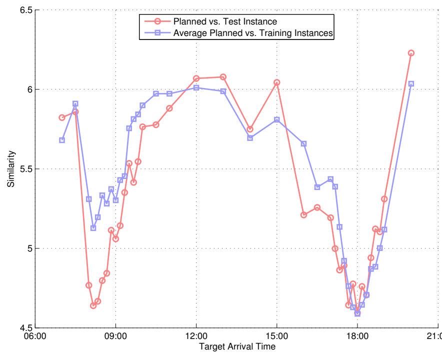
Both Buffer-ξ and Mean-Risk-c, methods that require a parameter, formed Pareto optimal fronts in both (a) and (b). Clearly, Mean-Risk-c benefits from the additional information from the input instances $T_1, \ldots, T_6$ and it dominated Buffer-ξ in both (a) and (b). The Similarity method needs no parameter tweaking, it is based only on two past observations, and still proposed a solution which departs not too early and gives a reasonable arrival rate. Notice that Norm-Inf (the generalization of Similarity) also benefits from the knowledge of the six past observations and with no parameter tweaking it produces a solution which gives a very reasonable trade-off between departure time, arrival rate and the standard deviation on the arrival time. Moreover, the solutions proposed by Norm-Inf performed rather well compared to all the competitors (which require parameter tweaking).

Influence of the Similarity between Input and Test Instances. We just saw that journeys proposed by the similarity-based approaches performed rather poorly, with respect to both arrival rate as well as standard deviation on the arrival time. However, we have to take into account that these methods use only two recorded timetables as input: if both differ substantially from the test instance, then in general there is very little one can do. The generic approach by Buhmann et al. [5] works well if both the input and the test instances are typical, i.e., if their mutual similarity is high. Thus we investigate the impact of high and low mutual similarities on the quality of the predictions.

First we note that the similarity $S_{\gamma_{OPT}}$ does not only depend on the two input instances but also on the origin s and the destination t, and on the target arrival time t_A. Thus, in the following experiments, we do not always use the same timetables T_5, T_6 as input and T_7 for testing, but select for every (s, t) the timetables whose mutual similarities are as high or as low as possible. Let Υ be the set of all triples of recorded timetables $(T_i, T_j, T_k) \in \mathcal{T}^3$ where i, j, k are mutually different. For a given pair (s, t) and two timetables $T_i, T_j \in \mathcal{T}$, let S^{st}_{ij} be the similarity of T_i and T_j with respect to s and t. We selected triples whose



(a)



(b)

Fig. 3. Influence of low/high similarity on the arrival rate: comparing various methods (a). Influence of the target arrival time on the similarity of the planned timetable and the test instance T_7, and on the average similarity between the planned timetable and each of the input instances $T_1, \ldots, T_6$ (b).

minimum (or maximum, resp.) pairwise similarity is as high or or low as possible,

$$(T_1^h, T_2^h, T_3^h) = \underset{(T_i,T_j,T_k)\in\Upsilon}{\arg\max}\ \min\left\{S_{ij}^{st}, S_{ik}^{st}, S_{jk}^{st}\right\} \tag{6}$$

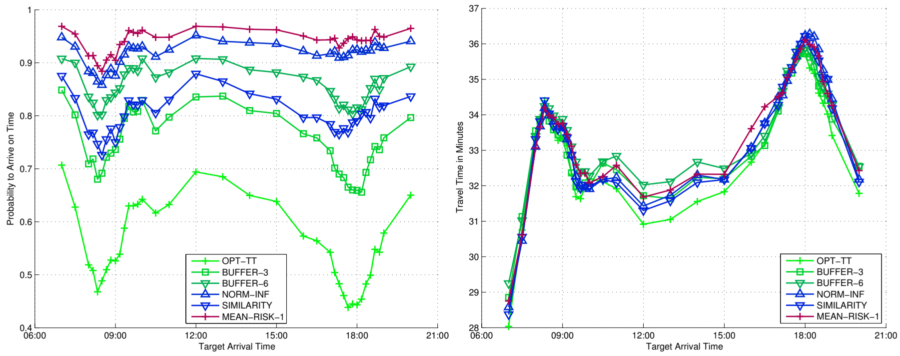
$$(T_1^l, T_2^l, T_3^l) = \underset{(T_i,T_j,T_k)\in\Upsilon}{\arg\min}\ \max\left\{S_{ij}^{st}, S_{ik}^{st}, S_{jk}^{st}\right\} \tag{7}$$

and used T_1^h and T_2^h as input and T_3^h for testing, and for comparison, used T_1^l and T_2^l as input and T_3^l for testing. Even though Mean-Risk-c and Norm-p could handle more instances, they were given just the two mentioned instances.

Figure 3(a) shows that all methods benefit when the similarity of the three instances is high. The arrival rates of both Norm-p and especially Similarity increase significantly. We observed that Similarity outperforms Norm-p when the similarity is low, which is reasonable: for a low similarity, the routes in the first intersection of the approximation sets as well as the route that maximizes the average departure time are too much influenced by the noise in the input instances. However, Similarity can still let the approximation sets grow beyond the first intersection so that more stable solutions are contained (which Norm-p can not). On the other hand, if the similarity is high, then there is so few noise in the data that S_γ is maximized already at the first γ for which the intersection is non-empty, thus Similarity and Norm-p are nearly identical.

Of course these results cannot directly be used for designing an algorithm, since the test instance is unknown. Nevertheless we believe that the results are interesting because they demonstrate the power of the similarity-based approach.

Influence of the Target Arrival Time. Figure 4 shows how the behavior of the methods, in terms of the arrival rate (a) and travel time (b), changes over the day. In particular, we can observe a clear influence of the morning and evening



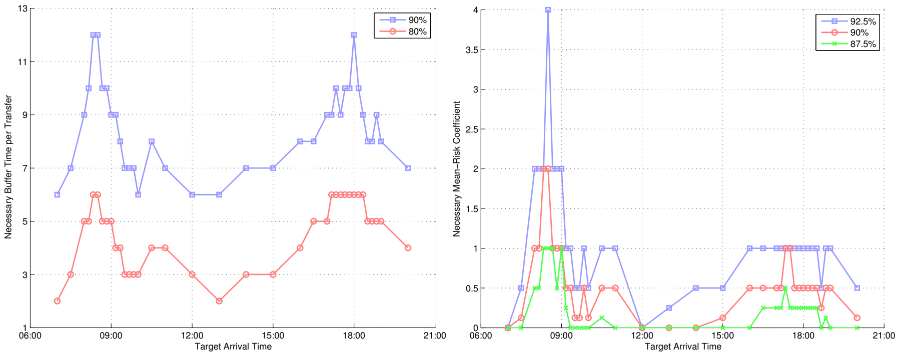
(a) (b)

Fig. 4. Comparison of various methods: Arrival rate vs. target arrival time (a), and travel time vs. target arrival time (b).

rush hours. Interestingly, Figure 4(a) shows that the probability to arrive on time of different methods is affected differently by the morning and evening rush hour. Specifically, Buffer-ξ, which is based solely on the planned timetable T_{plan}, seems to be greatly affected by both morning and afternoon rush hour. On the other hand, the methods that use recorded timetables for suggesting a journey are less affected by the afternoon rush hour than by the morning rush hour.

To understand this behavior, let us first look at Figure 3(b). The red curve shows how the value of similarity of T_{plan} and the test instance T_7 changes during the day. In particular, we see a significant drop of similarity during rush hours. Notably, the two peaks corresponding to morning and evening rush hour are of the same height. This suggests that on the day corresponding to T_7, during the morning rush hour, there was a similar amount of irregularities, with respect to T_{plan}, as during the evening one. On the other hand, the blue curve in Figure 3(b) shows the changes during the day of the averaged value of similarity of T_{plan} and each of the training instances $T_1 - T_6$. Also there the similarity drops during rush hours, but we clearly see that the morning peak is significantly lower than the evening one. This suggests that in the six recorded timetables $T_1 - T_6$ used as training instances for the algorithms, the amount of irregularities (with respect to T_{plan}) during morning rush hour was significantly lower than during evening rush hour. Thus, when comparing the two curves, we see a significant gap between them during the morning rush hour, but a relative match during the rest of the day. This suggests that the test instance T_7 contained during the day a similar amount of irregularities as it is expected on a typical day (represented by $T_1 - T_6$), with the only exception of the morning rush hour, where it was much more irregular than on a typical day.

Let us now relate what we observed in Figures 3(b) and 4(a). Since Buffer-ξ strategies are based solely on T_{plan}, any irregularities with respect to T_{plan} occurring in T_7 (captured by the red curve in Figure 3(b)) affect its arrival rate. This explains why the arrival rate of Buffer-ξ (in Figure 4(a)) drops both in



(a) Buffer-ξ (b) Mean-Risk-c

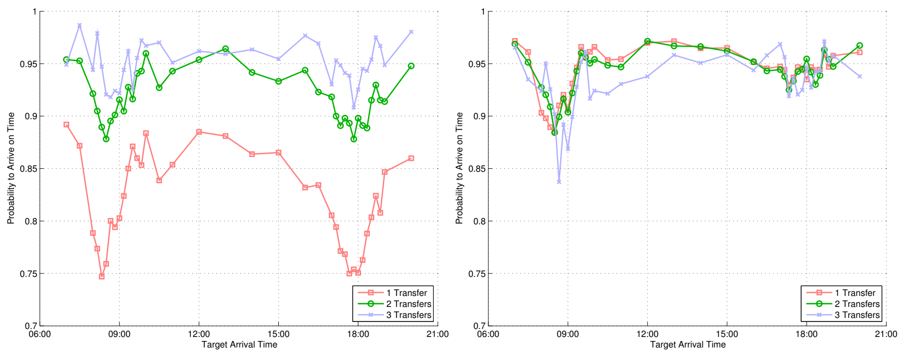
Fig. 5. Necessary parameter to achieve a specified arrival rate in T_7 depending on the target arrival time.

the morning and evening rush hour and exhibits two peaks of nearly the same height. On the other hand, the methods that use the information from the past observations (e.g., Mean-Risk-c) are trained to account for a certain amount of irregularities (blue curve in Figure 3(b)). Since the situation in T_7 in the evening is typical, the solutions proposed by these methods are prepared for it and their arrival rate (in Figure 4(a)) is almost not affected by the evening rush hour. In contrast, morning rush hour causes their arrival rate to drop significantly and this maps to the discrepancy of the red and blue curve in Figure 3(b).

In Figure 4(b) we observe that during peak hours, the travel time increases. Interestingly, the required travel time does not depend on the method nor whether it is on time or not. Thus, to achieve higher probability to arrive on time, one has to depart earlier (as seen in Figure 2(a)), but does not need to increase the time spent traveling.

Choice of the Parameters for Buffer-ξ *and* Mean-Risk-c*.* Figure 5(a) displays the minimum value of the parameter ξ of Buffer-ξ that would be necessary to achieve arrival rates of 80%, and 90% of the cases in T_7, and how this value changes in the course of a day. We see that, affected by the daily rush hours, this parameter varies significantly, suggesting that the Buffer-ξ strategy needs a non-trivial amount of parameter tweaking. We observe that the peaks corresponding to morning and evening rush hours are of the same height. Again, we can directly relate this behavior with the observed similarity of T_{plan} and T_7 (captured by the red curve in Figure 3(b)).

Similarly, Figure 5(b) displays the value of the coefficient c of Mean-Risk-c that would be necessary to achieve arrival rates of 78.5%, 90%, and 92.5% in T_7, and its development during the day. We observe that this value is greatly affected by the morning rush hour. On the other hand, the peak corresponding to the evening rush hour is visible, but not too significant. Again, we link this behavior of the value to the observed similarity of the training/test instances with the planned timetable—the two curves captured by Figure 3(b). Recall that in the



(a) Buffer-ξ (b) Mean-Risk-c

Fig. 6. Influence of the number of transfers on the arrival rate.

morning rush hour there is a gap between the two curves in Figure 3(b) indicating that the situation in T_7 was not typical with respect to previous observations. As we see in Figure 5(b), quite large value of the coefficient c would be necessary to compensate for the unexpected irregularities. In contrast, in a situation that is typical (i.e., when the two curves in 3(b) approximately match), the Mean-Risk-c method performs well and fine-tuning of the parameters is not crucial. For instance, a coefficient c set to 1 leads to reasonably robust solutions.

Influence of the Number of Transfers. Figure 6(a) shows that the arrival rate of Buffer-ξ is quite sensitive to the number of transfers. This suggests that the number of transfers is another aspect (of possibly many aspects) which has to be taken into account when searching for the best parameter for Buffer-ξ. In contrast, Figure 6(b) shows that the influence of the number of transfers on the arrival rate of the Mean-Risk-c method is almost negligible. Thus, there is no need to fine-tune the coefficient c to compensate for this aspect. We remark that we generally observed that the arrival rate of the methods based on the past observations is not very sensitive to the number of transfers.

5 Conclusion

We observed a clear trade-off: to achieve higher probability to arrive on time in a network full of delays, one has to depart earlier and expect higher standard deviation on the arrival time. On the other hand, the average travel time itself does not change with robustness or the choice of a routing method. The methods based solely on the planned timetable, where the robustness is achieved by adding buffer times, need a non-trivial parameter tweaking for which many aspects need to be considered (time of the day, number of transfers, etc.). The methods that learn from past benefit from the additional knowledge: If the test instance is typical with respect to the past observations, these strategies perform well, Mean-Risk-c does not need much fine-tuning, and Norm-Inf with no parameter tweaking proposes a highly competitive solution with a reasonable trade-offs. We have seen that similarity gives a good measure of the amount of irregularities in the network and can help to detect typical situations. Notably, it considers complex solutions (journeys), and thus it has a potential to capture behavior that cannot be observed only locally. We believe that this measure is worth further exploring, and by considering various aspects (e.g., how would different approaches benefit if we use similarity to preselect typical instances for training?) it can bring us even closer to the goal of robust routing.

Acknowledgements. This work has been partially supported by the Swiss National Science Foundation (SNF) under the grant number 200021 138117/1, and by the EU FP7/2007-2013 (DG CONNECT.H5-Smart Cities and Sustainability), under grant agreement no. 288094 (project eCOMPASS). Kateřina Böhmová is a recipient of a Google Europe Fellowship in Optimization Algorithms, and this research is supported in part by this Google Fellowship. We wish to thank the Verkehrsbetriebe Zürich (VBZ) for providing historic real-world data.

References

1. Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.: Route planning in transportation networks (2014)
2. Bast, H., Sternisko, J., Storandt, S., et al.: Delay-robustness of transfer patterns in public transportation route planning. In: ATMOS. pp. 42–54 (2013)
3. Böhmová, K., Mihalák, M., Pröger, T., Šrámek, R., Widmayer, P.: Robust routing in urban public transportation: How to find reliable journeys based on past observations. In: ATMOS. pp. 27–41 (2013)
4. Boyan, J., Mitzenmacher, M.: Improved results for route planning in stochastic transportation. In: SODA. pp. 895–902 (2001)
5. Buhmann, J.M., Mihalák, M., Šrámek, R., Widmayer, P.: Robust optimization in the presence of uncertainty. In: ITCS. pp. 505–514 (2013)
6. Dibbelt, J., Pajor, T., Strasser, B., Wagner, D.: Intriguingly simple and fast transit routing. In: SEA. pp. 43–54 (2013)
7. Dibbelt, J., Strasser, B., Wagner, D.: Delay-robust journeys in timetable networks with minimum expected arrival time. In: ATMOS. pp. 1–14 (2014)
8. Disser, Y., Müller-Hannemann, M., Schnee, M.: Multi-criteria shortest paths in time-dependent train networks. In: Experimental Algorithms, pp. 347–361. Springer (2008)
9. Firmani, D., Italiano, G.F., Laura, L., Santaroni, F., et al.: Is timetabling routing always reliable for public transportl. In: ATMOS. pp. 15–26 (2013)
10. Frank, H.: Shortest paths in probabilistic graphs. Operations Research 17(4), 583–599 (1969)
11. Goerigk, M., Heße, S., Müller-Hannemann, M., Schmidt, M., Schöbel, A.: Recoverable robust timetable information. In: ATMOS. pp. 1–14 (2013)
12. Goerigk, M., Knoth, M., Müller-Hannemann, M., Schmidt, M., Schöbel, A.: The price of robustness in timetable information. In: ATMOS. pp. 76–87 (2011)
13. Lim, S., Sommer, C., Nikolova, E., Rus, D.: Practical route planning under delay uncertainty: Stochastic shortest path queries. In: Robotics: Science and Systems VIII (2012)

14. Müller-Hannemann, M., Schnee, M.: Efficient timetable information in the presence of delays. In: Robust and Online Large-Scale Optimization. pp. 249–272. Springer (2009)
15. Nikolova, E., Kelner, J.A., Brand, M., Mitzenmacher, M.: Stochastic shortest paths via quasi-convex maximization. In: ESA. pp. 552–563 (2006)