Is Timetabling Routing Always Reliable for Public Transport?

Donatella Firmani[1], Giuseppe F. Italiano[1], Luigi Laura[2], and Federico Santaroni[1]

1 **Department of Civil Engineering and Computer Science Engineering University of Rome "Tor Vergata", Rome, Italy**
firmani@ing.uniroma2.it, italiano@disp.uniroma2.it, santaroni@ing.uniroma2.it

2 **Department of Computer, Control, and Management Engineering and Research Centre for Transport and Logistics – Sapienza University of Rome, Italy**
laura@dis.uniroma1.it

Abstract

Current route planning algorithms for public transport networks are mostly based on timetable information only, i.e., they compute shortest routes under the assumption that all transit vehicles (e.g., buses, subway trains) will incur in no delays throughout their trips. Unfortunately, unavoidable and unexpected delays often prevent transit vehicles to respect their originally planned schedule. In this paper, we try to measure empirically the quality of the solutions offered by timetabling routing in a real public transport network, where unpredictable delays may happen with a certain frequency, such as the public transport network of the metropolitan area of Rome. To accomplish this task, we take the time estimates required for trips provided by a timetabling-based route planner (such as Google Transit) and compare them against the times taken by the trips according to the actual tracking of transit vehicles in the transport network, measured through the GPS data made available by the transit agency. In our experiments, the movement of transit vehicles was only mildly correlated to the timetable, giving strong evidence that in such a case timetabled routing may fail to deliver optimal or even high-quality solutions.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Shortest Path Problems, Route Planning, Timetable-based Routing, Public Transport Networks

Digital Object Identifier 10.4230/OASIcs.ATMOS.2013.15

1 Introduction

In the last years we have witnessed an explosion of exciting research on point-to-point shortest path algorithms for road networks, motivated by the widespread use of navigation software. Many new algorithmic techniques have been introduced, including hierarchical approaches (e.g., contraction hierarchies) [15, 26], reach-based approaches [18, 19], transit node routing [6], and hub-based labeling algorithms [1]. (Delling et al. [11] gives a more detailed overview of the literature.) The algorithms proposed in the literature are of great practical value, as on average they are several orders of magnitude faster than Dijkstra's algorithm, which is too slow for large-scale road networks: on very large road networks, such as the entire Western Europe or North America, the fastest algorithms are able to compute point-to-point distances in few microseconds on high-performance computing platforms and in hundred

13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'13).
Editors: Daniele Frigioni, Sebastian Stiller; pp. 15–26
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

milliseconds on mobile devices (see e.g., [17]). Computing the actual shortest paths (not only distances) requires slightly more time (i.e., few order of magnitudes), but it is still very fast in practice. We remark that this algorithmic work had truly a big practical impact on navigation systems: some of the ideas introduced in the scientific literature are currently used by Apple, Bing and Google Maps. Furthermore, this research on point-to-point shortest path algorithms generated not only results of practical value, but also deep theoretical questions that gave rise to several exciting results: Abraham et al. [2] gave theoretical justifications of the practical efficiency of some of those approaches under the assumption of low highway dimension (HD) of the input graph, which is believed to be true for road networks, and even showed some amazing relationships to VC dimension [1].

Although most algorithmic techniques designed for road networks can be immediately transferred to public transport networks, unfortunately their adaptation to this case is harder than expected, and they fail to yield comparable speed-ups [5, 14]. One of the reasons, as explained in the excellent work of Bast [4], is that most public transportation networks, like bus-only networks in big metropolitan areas, are far more complex than other types of transportation networks, such as road networks: indeed, public transport networks are known to be less hierarchically structured and are inherently event-based. Thus, it seems that, in order to achieve significant speed-ups on public transport networks, one needs to take into account more sophisticated and larger scale time-dependent models [9, 14, 24, 25] or to develop completely different algorithmic techniques, such as either the transfer patterns introduced by Bast et al. [5], the approach based on dynamic programming by Delling et al. [10] or the connection scan by Dibbelt et al. [12].

Current route planning algorithms for public transport networks are mostly based on timetable information, i.e., they compute shortest routes under the assumption that all transit vehicles (e.g., buses, subway trains) will start their trip exactly at the planned time and that they will incur in no delays throughout their journey. However, in our daily experience buses often run behind schedule: unavoidable delays occur frequently and for many unplanned reasons, including traffic jams, accidents, road closures, inclement weather, increased ridership, vehicle breakdowns and sometimes even unrealistic scheduling. As a consequence, widely used timetable routing algorithms may suffer from several inaccuracies: the more buses run behind schedule, the more is likely that routing methods based on timetabling will not be able to estimate correctly the waiting times at bus stops, thus failing to deliver optimal solutions, i.e., the actual shortest routes. Indeed, in the recent past, a lot of effort has been put in developing either robust models able to efficiently cope with delays and cancellation events [8, 13, 16, 7], or dynamic delay propagation models for the design of robust timetables and the evaluation of dispatching proposals [23]. These approaches yield interesting insights into the robustness of the solutions offered against small fluctuations.

In this framework, it seems quite natural to ask how much timetabling-based routing methods are effectively able to deliver optimal solutions on actual public transport networks. To address this complex issue, in this paper we try to measure the quality of the solutions offered by timetabling routing in the public transport network of the metropolitan area of a big city, where unpredictable delays, unplanned disruptions or unexpected events seem to happen with a certain frequency. As a first step, we consider the public transport network of Rome: we believe that fluctuations on the transit schedule are not limited to this case, but they happen often in many other urban areas worldwide. In more detail, we performed the following experiment. On a given day, we submitted to Google Transit, the well known public transport route planning tool integrated in Google Maps, many queries having origin and destination in the metropolitan area of Rome: in this case, the journeys computed by

Google Transit are based on the timetabling data provided by the transit agency of Rome[1]. Besides its origin and destination, each query q_i is characterized by the starting time τ_i from the origin. For each query q_i, on the same day we followed precisely the journeys suggested by Google Transit, starting at time τ_i, by tracking in real time the movement of transit vehicles in the transport network through the GPS data made available by the very same transit agency. In order to do that efficiently, we collected the GPS data on the geo-location of all vehicles on the very same day, by submitting queries every minute to the transit agency of Rome [28]. With all the data obtained, we built a simulator capable of following precisely each journey on that given day, according to the GPS tracking of transit vehicles in the transport network. Finally, we computed the actual total time required by each journey in our simulator and compared it against its original estimate given by Google Transit. We believe that the simulator built for this experiment was not only instrumental for its success, but it can also be of independent interest for other investigations in a public transport network.

Our experimental analysis shows that in the public transport network considered the movement of transit vehicles was only mildly correlated to the original timetable. In such a scenario, timetabled-based routing methods suffer from many inaccuracies, as they are based on incorrect estimations of the waiting/transfer times at transit stops, and thus they might fail to deliver an optimal or even high-quality solution. In this case, in order to compute the truly best possible routes (for instance, shortest time routes), it seems that we have to overcome the inherent oversights of timetable routing: toward this end, we advocate the need to design new route planning algorithms which are capable of exploiting the real-time information about the geo-location of buses made available by many transit authorities.

2 Preliminaries

In the following we introduce some basic terminology which will be useful throughout the paper. Our public transport networks consist of a set of *stops*, a set of *hops* and a set of *footpaths*:

- A *stop* corresponds to a location in the network where passengers may either enter or exit a transit vehicle (such as a bus stop or a subway station).
- A *hop* is a connection between two adjacent stops and models a vehicle departing from stop u and arriving at stop v without intermediate stops in between.
- A *trip* consists of a sequence of consecutive hops operated by the same transit vehicle. Trips can be grouped into *lines*, serving the exact same sequence of consecutive hops.
- A *footpath* enables walking transfers between nearby stops. Each footpath consists of two stops and an associated (constant) walking time between the two stops.
- A *journey* connects a source stop s and a target stop t, and consists of a sequence of trips and footpaths in the order of travel. Each trip in the journey is associated with two stops, corresponding to the pick-up and drop-off points.

[1] Roma Servizi per la Mobilità [28].

3 Experimental Setup

3.1 Experiments

In our experiments, we considered the public transport network of Rome, which consists of 309 bus lines and 3 subway lines, with a total of 7,092 stops (7,037 bus stops and 55 subway stops). We generated random queries, where each query q_i consisted of a triple $\langle s_i, t_i, \tau_i \rangle$:

- s_i is the start stop;
- t_i is the target stop;
- τ_i is the time of the departure from the start stop.

Our experiments were carried out as follows. Each start and target stop s_i and t_i was generated uniformly at random in the metropolitan area of Rome, while the departure time τ_i was chosen uniformly at random between 7:00am and 9:00pm. We selected Thursday June 6, 2013 as a day for our experiments, and in this day we did not observe any particular deviation form the typical delays in the trips. We submitted each query q_i to Google Transit on the very same day (June 6, 2013), and collected all the journeys suggested in return to the query and their predicted traveling times. In the vast majority of cases, Google Transit returns 4 journeys, but there were queries that returned less than 4 public transit journeys; this might happen, for instance, when one of the journeys returned is a footpath. This produced a total of $4,018$ journeys. Note that, since Google Transit is based on the timetabled data provided by the transit agency of Rome, the predicted traveling time of each journey is computed according to the timetable.

We next tried to measure empirically the actual time required by each such journey in the real public transport network. We performed this as follows. On June 6, 2013 we submitted queries every minute to the transit agency of Rome [28], in order to obtain (from GPS data) the instantaneous geo-location of all vehicles in the network. Given that stream of GPS data, we built a simulation system capable of following precisely each journey from a given starting time, according to the GPS tracking of transit vehicles in the transport network. We describe this process in more detail in Section 3.2. Finally, we computed the actual total time required by each route in our simulator and compared it against its original estimate given by Google Transit.

3.2 Simulation system

Our system makes it possible to simulate closely the experience of a user traveling according to each input journey, after leaving the origin at the corresponding time. For each trip in the journey, the pick-up and drop-off times are computed according to the position of transit vehicles in the public transport network. A user can be picked-up or dropped-off either earlier or later than originally scheduled, and if a delayed transit vehicle misses a connection then the next trip of the same line is chosen. To obtain the real-time position of ground vehicles (such as buses, trains or trams) we used streamed GPS data, while for trips which do not provide vehicle live positions (such as saubway train trips) we employed their original estimate given by Google Transit. This allows us to follow input journey containing both ground and underground trips as well. We remark that all of the journeys produced in our experiments contained at least one trip operated by ground vehicles. Finally, we used Google Maps to compute the times needed by footpaths.

4 Experimental Results

In this section we report the results of our experiments. We compare the *estimated time* $t_e(j)$ required by each journey j according to the timetable (as reported by Google Transit), and its *actual time* $t_a(j)$ computed from the vehicle real-time positions given by the stream of GPS data (as contained in our simulation system). More specifically, we define the *error coefficient* of journey j to be $t_a(j)/t_e(j)$. Note that the error coefficient measures the distance between the time predicted by timetabling routing and the actual time that journey j will incur in reality. It will be equal to 1 whenever the actual journey will be in perfect agreement with the times predicted by timetabling routing. It will be larger than 1 whenever the actual journey will be slower than what was predicted by timetabling routing (increased waiting times at a bus stop for a delayed connection). It will be smaller than 1 whenever the actual journey will be faster than what was predicted by timetabling routing (smaller waiting times at a bus stop, which can happen in the case a previous connection, which was infeasible by timetabling, was delayed and can become a viable option in the actual journey). Obviously, the more the error coefficient will deviate substantially from 1 (especially in the case where it is larger than 1), the less accurate will be the time estimations of timetabling routing and the more likely is that timetabling routing will fail to compute the shortest journeys.

4.1 Measured error coefficients

To report the distribution of the error coefficients as a function of the journey time, we proceed as follows. For each journey j, the journey time is taken as the estimated time $t_e(j)$ according to the timetable. Since there can be multiple journeys sharing the same value of $t_e(j)$, we group those journeys into *time slots* within a 3-minute resolution. More formally, we measure $t_e(j)$ in minutes and the k-th time slot σ_k contains all journeys j such that $t_e(j) \in [3k, 3(k+1)]$. For each time slot, we look at the proximity of the obtained error coefficient distribution to the constant 1, which represents the ideal scenario where the times of actual journeys are in perfect agreement with the times predicted by timetabling routing. To this end, we compute the metrics below:

- **Average.** We measure the average of the error coefficient in each time slot.
- **Percentiles.** Analogously, for each time slot σ, we measure the 10th percentile and the 90th percentile of the error coefficients.
- **Minimum-Maximum.** Finally, we measure $\min_{j\in\sigma}\{\frac{t_a(j)}{t_e(j)}\}$ and $\max_{j\in\sigma}\{\frac{t_a(j)}{t_e(j)}\}$

We define $t_e(\sigma_k) = 3k + 1.5$ and plot both the evolution of these statistics and the error coefficient, as functions of t_e. This also enable us to distinguish between *short distance* journeys, i.e., journeys j with $t_e(j)$ smaller than 30 minutes, *medium distance* journeys, i.e., journeys j with $t_e(j)$ between and 30 and 60 minutes, and *long distance* journeys, i.e., journeys taking more than 60 minutes.

Figure 1 plots the error coefficient for each journey and illustrates the average of the error coefficients for each time slot obtained in our experiments. Note that the error coefficients fluctuate wildly, ranging from 0.15 to 4.44, and the reader may ask how actual trips with extremely small or extremely high error coefficients look like. To this end, we provide more details on two extreme cases, which are a short journey with minimum error coefficient and a long journey with maximum error coefficient, denoted by j_m and j_M respectively:

- j_m consists of a single short distance trip, $t_a(j_m) = 2$ minutes, $t_e(j_m) = 13$ minutes and error coefficient ≈ 0.15;
- j_M consists of 3 short distance trips and 1 medium distance trip, $t_a(j_M) = 3$ hours and 49 minutes, $t_e(j_M) = 1$ hour and 24 minutes and error coefficient ≈ 2.72.

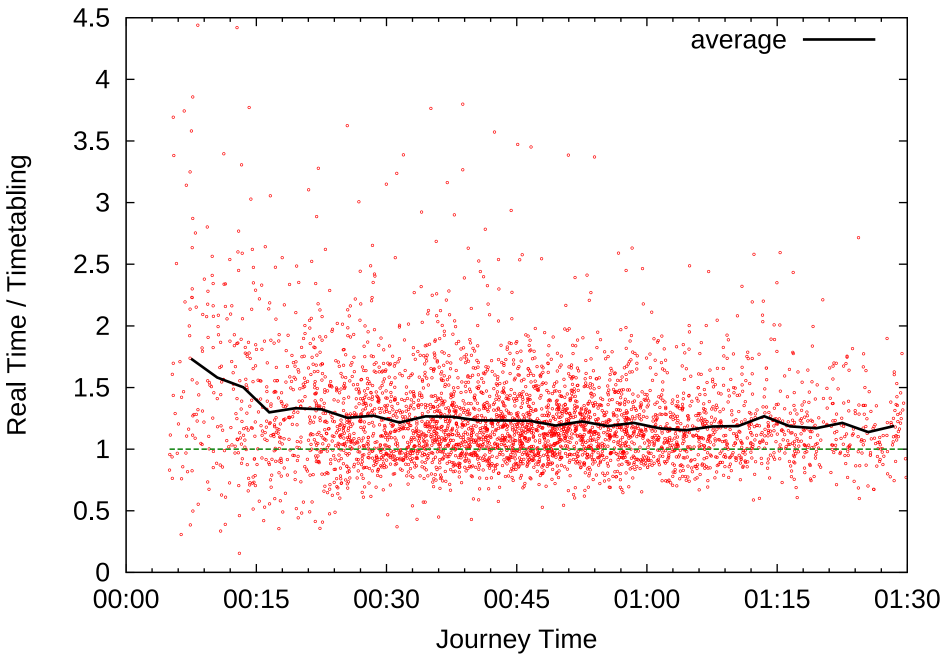


Figure 1 Distribution of the error coefficients as a function of the journey times (better viewed in color).

The short journey connects two stops which are rather close to each other, and only require a 1-minute bus trip: in this case, the discrepancy between the estimated and the actual travel time is induced by the waiting time at the bus stop. The long journey connects two stops which are rather far away: the journey itself consists of four trips (three short distance and one long distance trip), operated by ground vehicles through intense traffic areas. This results in moderate delays on the short distance trips and a much higher delay on the medium distance trip due to intense traffic.

While high fluctuations are possible, the average error coefficient lies in the interval $[1.13, 1.73]$, which implies that on the average the actual journey times are between 13% and 73% slower than the times used by timetabling routing! In detail, the average error coefficient falls between 1.27 and 1.73 for short journeys, and between 1.13 and 1.26 for long journeys. The fact that the error coefficients appear to be substantially larger for short journeys is not surprising, as short journeys are likely to be more affected (in relative terms) by fluctuations on the schedule. On the other side, larger errors might be less tolerable on short journeys from the users' perspective.

Figure 2 shows the 10th and the 90th percentiles of the distribution of the error coefficients. For the sake of comparison, for each time slot we report also the minimum and the maximum error coefficient. This gives us an interesting insight on a typical user experience: in 80% of the short journeys computed by a timetable-based method, the actual time required ranges from 0.72 to 3.14 of the time estimated with timetabling. Analogously, the same percentage of long journeys takes up to 2 times more than the estimated time. As for the first and last deciles, we observe higher variability in the short journeys rather than in the long journeys. Finally, we observe that 10% of the journeys taking from 15 to 45 minutes are distributed over a long tail in the range $[1.6, 3.8]$. Roughly speaking, 1 such journey out of 10 will take more than twice the scheduled time!

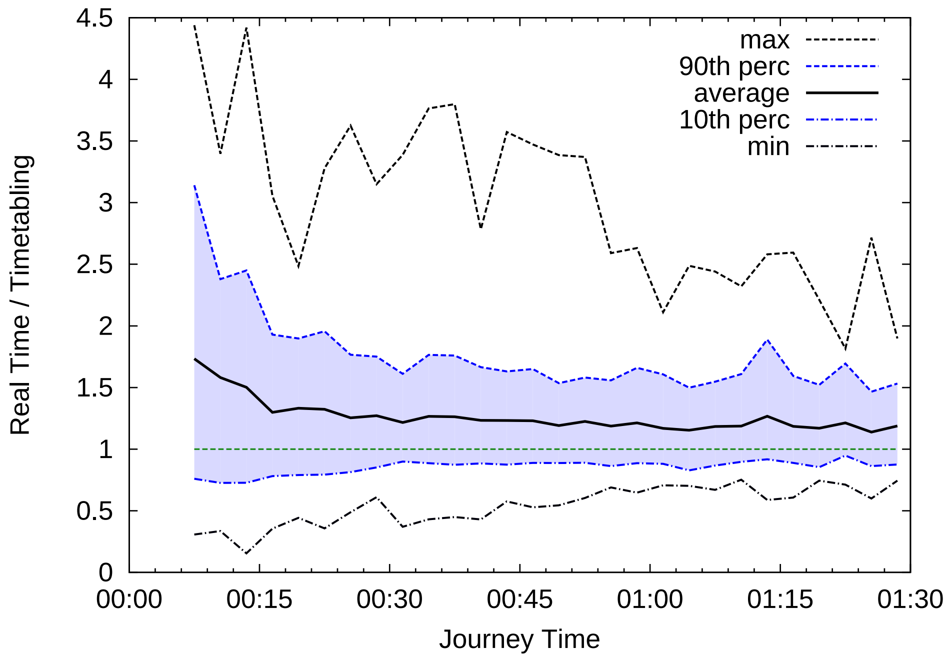


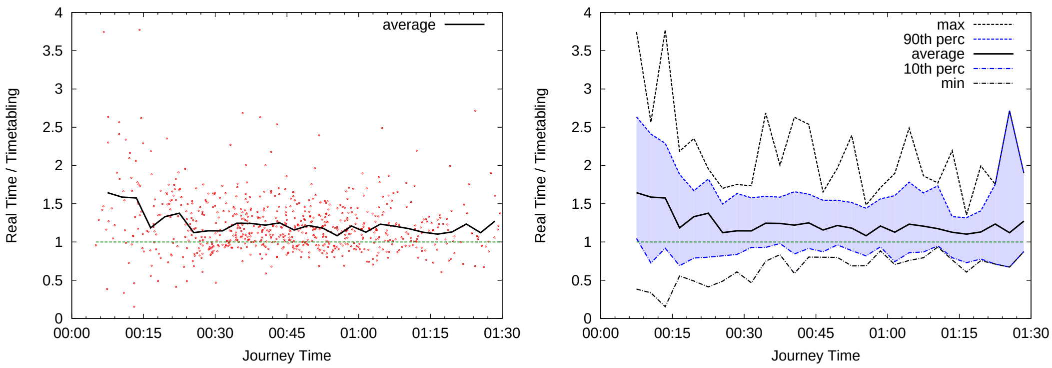
Figure 2 The 10th and the 90th percentiles of the distribution of the error coefficients (better viewed in color).

It is natural to ask in this scenario whether different discrepancies between the estimated and the actual travel times could be observed under different traffic conditions. As illustrated in Figures 3–5, the distribution of the error coefficients is slightly affected by the different times of the day, which mainly differ for the traffic conditions. This is not surprising, as our queries are generated at random and do not follow the traffic patterns. Since in the morning rush hours there is more traffic towards the city center, while in the evening rush hours the traffic flows out of the city center, only a small percentage of random queries are likely to be affected by those traffic patterns. In the full paper, we will report the result of other experiments that will highlight this phenomenon.

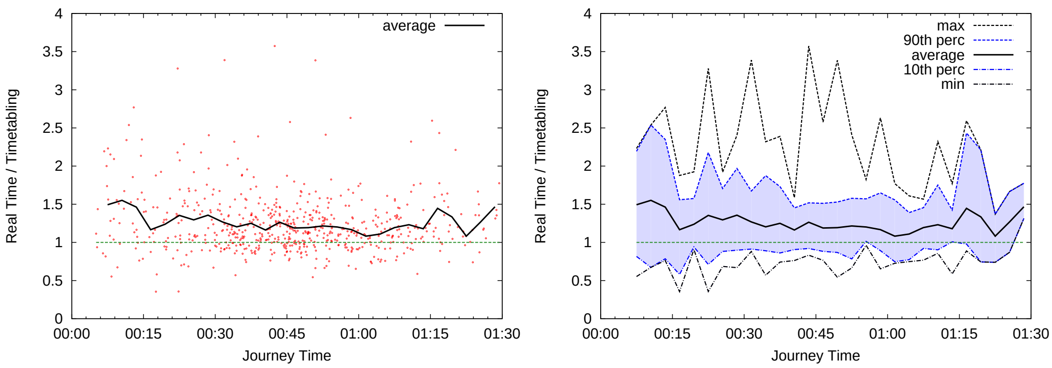
4.2 Correlations in ranking

In order to get deeper insights on the differences between the time estimates provided by timetabling and the actual times obtained by tracking transit vehicles in the network, we next investigate the relative rankings of journeys. Namely, for each query we take the four journeys provided by Google Transit and compare their relative rankings in the lists produced by two methods, according to the travel times. If the ranking of the four journeys agree (say, the shortest journey for timetabled routing is also the shortest journey in our real-time simulation with GPS data, the second shortest journey for timetabled routing is also the shortest journey in our real-time simulation, etc...) then there is a strong correlation between the two rankings, independently of the values of the journey times.

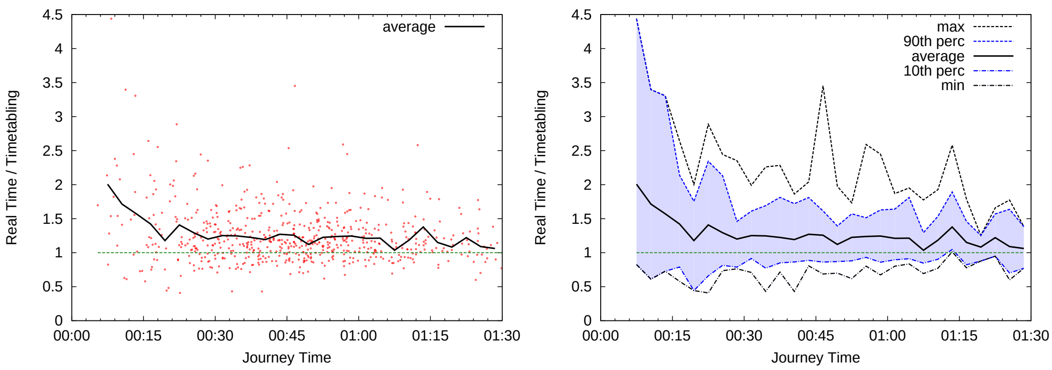
To assess the degree of similarity between the two rankings, we use the Kendall Tau coefficient [22]. This is a rank distance metric that counts the number of pairwise disagreements between two ranking lists: the larger the distance, the more dissimilar the two lists



■ **Figure 3** Distribution of the error coefficients in journeys with time of the departure from 7:30am to 9:30am (better viewed in color).



■ **Figure 4** Distribution of the error coefficients in journeys with time of the departure from 11:30am to 1:30pm (better viewed in color).



■ **Figure 5** Distribution of the error coefficients in journeys with time of the departure from 5:00pm to 7:00pm (better viewed in color).

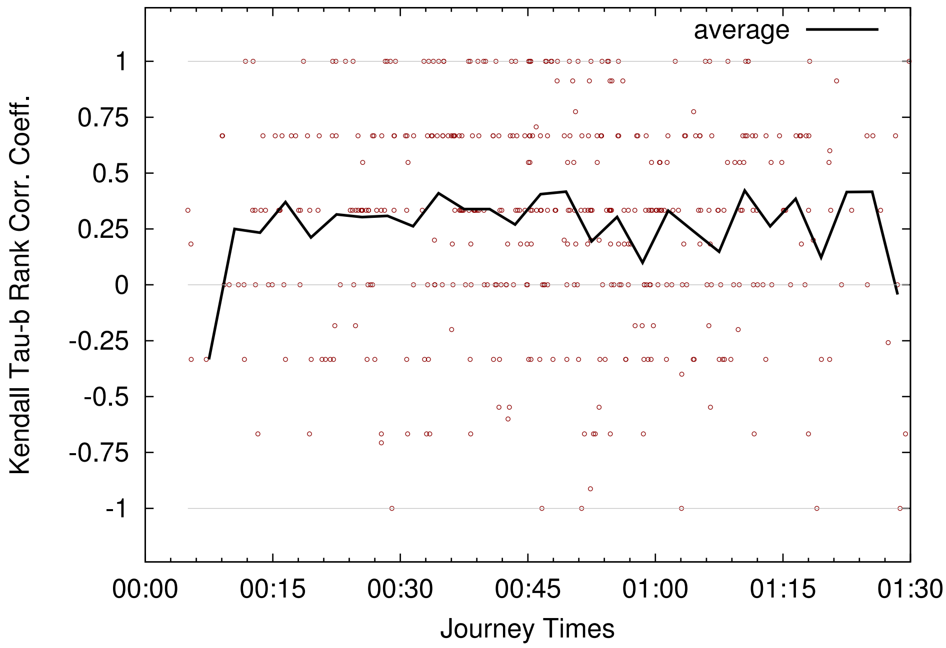


Figure 6 Kendall Tau-b coefficients for the queries in our experiment (better viewed in color).

are. In particular, we use the Tau-b statistic, which is used when ties exist [3]. The Tau-b coefficient ranges from -1 (100% negative association, or perfect inversion) to $+1$ (100% positive association, or perfect agreement): a value of 0 indicates the absence of association (i.e., independence of the two rankings).

Figure 6 shows values of the Kendall Tau-b coefficient for the queries considered in our experiment, plotted against the journey times. As one could expect, in many cases there is a positive correlation between the time estimates provided by timetabling and the actual times obtained by tracking transit vehicles. However, there are also values close to 0, and even worse, there are many negative Tau-b coefficients. The average Tau-b coefficient for each time slot is close to 0.25, which implies only a mildly positive correlation between the two rankings considered. In particular, the average Tau-b coefficient has smaller values for very short journeys and for long journeys: those cases appear to be more vulnerable to fluctuations in the schedule, and thus there seems to be a larger error on the time estimates provided by timetabled routing. In general, the rank correlation analysis given by the Kendall Tau-b statistics shows even more convincing arguments that, according to our experiments in the public transport network considered, timetabled routing fails to deliver optimal or even high-quality solutions.

5 Final Remarks

In this paper we measured empirically the quality of the solutions computed by timetabling routing in a real public transport network: for many queries, we compared the time estimate provided by Google Transit with the actual times, computed using the real-time GPS data of the transit vehicles. Our analysis shows that widely used timetable routing algorithms suffer from many inaccuracies, as they are based on incorrect estimations of the waiting/transfer times at transit stops, and thus they might fail to deliver an optimal solution.

The main question that arises naturally in this scenario is how to exploit the real-time information about the geo-location of buses to overcome the inherent oversights of timetable routing and to compute the truly best possible (under several optimization criteria) point-to-point routes, such as shortest routes, routes with minimum number of transfers, etc. As shown recently [20, 21, 27], geo-location data could in fact provide a more accurate and realistic modeling of public transport networks, as they are able to provide better estimates on many variables, such as bus arrival times, the times needed to make a transfer, or the times needed to travel arcs in the transport network. In particular, we expect that this more accurate modeling will make it possible to compute solutions of better quality overall.

Another important issue to investigate is how to compute robust routes, e.g., routes with more backup options (again, based on the current geo-location of buses) and thus less vulnerable to unexpected events. We remark that, whichever is the optimization criterion, route planning with real-time updates on the location of buses appears to be a challenging problem. This is because one has to deal with the sheer size of the input network, augmented with the actual location of buses and combined with a huge bulk of real-time updates, and the fact that such updates provide accurate information only about the past and the current state of the network, while, in order to answer effectively routing queries, one still needs to infer some realistic information about the future. Perhaps, this explains why a solution to these problems has been elusive, despite the fact that geo-location data have been already available for many years.

Acknowledgements We wish to thank the transit agency of Rome, Roma Servizi per la Mobilità, and in particular Luca Allulli, Carlo Gentile, Damiano Morosi, and Fabio Schiavo, for their kind assistance with the GPS data used to track the transit vehicles in the transport network of Rome.

References

1 Ittai Abraham, Daniel Delling, Amos Fiat, Andrew V. Goldberg, and Renato Fonseca F. Werneck. VC-dimension and shortest path algorithms. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *Proc. of the 38th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 690–699. Springer Berlin Heidelberg, 2011.

2 Ittai Abraham, Amos Fiat, Andrew V. Goldberg, and Renato Fonseca F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 782–793. SIAM, 2010.

3 A. Agresti. *Analysis of Ordinal Categorical Data*. Probability and Statistics. Wiley, 2010.

4 Hannah Bast. Car or public transport—two worlds. In Susanne Albers, Helmut Alt, and Stefan Näher, editors, *Proceedings of the 8th Symposium on Experimental Algorithms (SEA)*, volume 5760 of *Lecture Notes in Computer Science*, pages 355–367. Springer Berlin Heidelberg, 2009.

5 Hannah Bast, Erik Carlsson, Arno Eigenwillig, Robert Geisberger, Chris Harrelson, Veselin Raychev, and Fabien Viger. Fast routing in very large public transportation networks using transfer patterns. In Mark de Berg and Ulrich Meyer, editors, *Proceedings of the 18th annual European conference on Algorithms (ESA): Part I*, pages 290–301. Springer Berlin Heidelberg, 2010.

6 Holger Bast, Stefan Funke, Domagoj Matijevic, Peter Sanders, and Dominik Schultes. In transit to constant time shortest-path queries in road networks. In *Proceedings of the 9th Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 2007.

7 Annabell Berger, Daniel Delling, Andreas Gebhardt, and Matthias Müller-Hannemann. Accelerating time-dependent multi-criteria timetable information is harder than expected. In Jens Clausen and Gabriele Di Stefano, editors, *Proceedings of the 9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany, 2009.

8 Annabell Berger, Andreas Gebhardt, Matthias Müller-Hannemann, and Martin Ostrowski. Stochastic delay prediction in large train network. In Alberto Caprara and Spyros C. Kontogiannis, editors, *Proceedings of the 9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS)*, pages 100 – 111. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany, 2011.

9 Daniel Delling, Bastian Katz, and Thomas Pajor. Parallel computation of best connections in public transportation networks. *ACM Journal on Experimental Algorithmics*, 17:(4.4), October 2012.

10 Daniel Delling, Thomas Pajor, and Renato Fonseca Werneck. Round-based public transit routing. In David A. Bader and Petra Mutzel, editors, *Proceedings of the 14th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 130–140. SIAM, 2012.

11 Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering route planning algorithms. In Jürgen Lerner, Dorothea Wagner, and Katharina A. Zweig, editors, *Algorithmics of Large and Complex Networks*, pages 117–139. Springer Berlin Heidelberg, 2009.

12 Julian Dibbelt, Thomas Pajor, Ben Strasser, and Dorothea Wagner. Intriguingly simple and fast transit routing. In Vincenzo Bonifaci, Camil Demetrescu, and Alberto Marchetti-Spaccamela, editors, *Proceedings of the 12th Symposium on Experimental Algorithms (SEA)*, volume 7933 of *Lecture Notes in Computer Science*, pages 43–54. Springer Berlin Heidelberg, 2013.

13 Yann Disser, Matthias Müller–Hannemann, and Mathias Schnee. Multi-criteria shortest paths in time-dependent train networks. In Catherine C. McGeoch, editor, *Proceedings of the 7th Workshop on Experimental Algorithms (WEA)*, pages 347–361. Springer Berlin Heidelberg, 2008.

14 Robert Geisberger. Contraction of timetable networks with realistic transfers. In Paola Festa, editor, *Proceedings of the 9th Symposium on Experimental Algorithms (SEA)*, volume 6049 of *Lecture Notes in Computer Science*, pages 71–82. Springer Berlin Heidelberg, 2010.

15 Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In Catherine C. McGeoch, editor, *Proceedings of the 7th Workshop on Experimental Algorithms (WEA)*, pages 319–333. Springer Berlin Heidelberg, 2008.

16 M. Goerigk, M. Knoth, M. Müller–Hannemann, M. Schmidt, and A. Schöbel. The price of robustness in timetable information. In Alberto Caprara and Spyros C. Kontogiannis, editors, *Proceedings of the 9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS)*, pages 76 – 87. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany, 2011.

17 Andrew V. Goldberg. The hub labeling algorithm. In Vincenzo Bonifaci, Camil Demetrescu, and Alberto Marchetti-Spaccamela, editors, *Proceedings of the 12th Symposium on Experimental Algorithms (SEA)*, page 4. Springer Berlin Heidelberg, 2013.

18 Andrew V. Goldberg, Haim Kaplan, and Renato F. Werneck. Reach for A*: Efficient point-to-point shortest path algorithms. In Rajeev Raman and Matthias F. Stallmann, editors, *Proceedings of the 8th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 129–143. SIAM, 2006.

19 Ronald J. Gutman. Reach-based routing: A new approach to shortest path algorithms optimized for road networks. In Lars Arge, Giuseppe F. Italiano, and Robert Sedgewick, editors, *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics*, pages 100–111. SIAM, 2004.

20 Timothy Hunter, Ryan Herring, Pieter Abbeel, and Alexandre Bayen. Path and travel time inference from GPS probe vehicle data. In Daphne Koller, Yoshua Bengio, Léon Bottou, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 21*. Nips Foundation, 2009.

21 Erik Jenelius and Haris N. Koutsopoulos. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological*, 53(0):64 – 81, 2013.

22 M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1–2):81–89, 1938.

23 Matthias Müller-Hannemann and Mathias Schnee. Efficient timetable information in the presence of delays. In Christos D. Zaroliagis Ravindra K. Ahuja, Rolf H. Möhring, editor, *Robust and Online Large-Scale Optimization*, pages 249–272. Springer Berlin Heidelberg, 2009.

24 Matthias Müller-Hannemann, Frank Schulz, Dorothea Wagner, and Christos D. Zaroliagis. Timetable information: Models and algorithms. In Frank Geraets, Leo Kroon, Anita Schoebel, Dorothea Wagner, and Christos D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, pages 67–90. Springer Berlin Heidelberg, 2007.

25 Evangelia Pyrga, Frank Schulz, Dorothea Wagner, and Christos D. Zaroliagis. Efficient models for timetable information in public transportation systems. *ACM Journal on Experimental Algorithmics*, 12:(2.4), 2008.

26 Peter Sanders and Dominik Schultes. Highway hierarchies hasten exact shortest path queries. In Gerth Stølting Brodal and Stefano Leonardi, editors, *Proceedings of the 13th annual European conference on Algorithms (ESA)*, pages 568–579. Springer Berlin Heidelberg, 2005.

27 Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 316–324, New York, NY, USA, 2011. ACM.

28 Agenzia Roma servizi per la Mobilità. Muoversi a Roma. `http://muovi.roma.it/`, 2013. [Online; accessed June-2013].