A Multi-Modal Routing Approach Combining Dynamic Ride-Sharing and Public Transport

27. August 2015

{fahnenschreiber, guendling, keyhani, schnee}@cs.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Algorithms Group

Abstract

Ride-sharing facilitates cheap and eco-friendly mobility. In contrast to classical ride-sharing that basically works like a notice-board, dynamic ride-sharing allows a passenger to get a lift on a section of a driver's route and, if necessary, re-routes the driver. On a dynamic ride-sharing platform, drivers provide their routes and passengers specify queries consisting of departure and arrival location as well as a time for the journey. The platform computes suitable matches of driver routes and passenger queries, and proposes them to both parties.

State-of-the-Art platforms for public transport routing and dynamic ride-sharing provide unimodal connections but do not combine both transport modes. The challenge is that the driver routes are not static but could be changed significantly if the driver accepts the detour to pick up the passenger and drop him / her off at his / her destination. Thus, a driver's route may result in a number of dynamic ride-sharing offers, namely all possible connections between pick up and drop off points with an acceptable detour for the driver. In this paper, we present a solution that integrates dynamic ride-sharing into our existing multi-criteria intermodal travel information system. We solve two challenges: First, we allow dynamic ride-sharing between two train rides by connecting public transport stations by dynamic ride-sharing offers of drivers. For this, we integrate driver offers into our graph model, which represents the public transport timetable. Second, we find suitable dynamic ride-sharing offers of drivers who can take the passenger from his / her start location to a public transport station or from a station to the queried destination location. In our computational study, compared to unimodal train connections, we obtain a significant improvement of the results by combining public transport and dynamic ride-sharing.

1 Introduction

The algorithms group at the department of computer science of Technische Universität Darmstadt has a long track record in developing multi criteria timetable information systems [7][6][2]. Since 2008 our systems are capable of incorporating real time information. This allows us to support the traveler in case of delays [3]. Current research focuses on intermodal travel information systems incorporating public as well as private transportation [4].

Motivation

Current and future mobility requirements demand intelligent solutions. Ride-sharing is gaining in importance, and yields benefits compared to costly, eco-unfriendly individual car rides. Dynamic ride-sharing makes better use of existing resources by bringing travelers with matching routes together. Participating drivers and passengers can be matched on-demand or in advance. This is done by calculating the detour required for the driver to give the respective potential passenger a lift. If the result matches certain criteria, both the driver and the potential passenger receive an offer and decide whether to accept or decline.

Our existing multi-modal routing system computes door-to-door connections by incorporating public and private transportation. It optimizes the criteria travel duration and number of train and mode changes while the price is also taken into account. In this work, we present an approach to fulfill future individual traffic demands by proposing a journey information system, which adds dynamic ride-sharing to our routing system.

The motivation is that both these modes are complementary: On the one hand, public transport is quite sparse in some regions, but dynamic ride-sharing offers to reach areas with limited public transport connectivity. On the other hand, the number of dynamic ride-sharing offers relevant for a query increases when allowing routes which bring the passenger to a station where he can continue his journey using public transport.

This way, dynamic ride-sharing as well as public transportation benefit from this combination: by offering dynamic ride-sharing to travelers using public transportation, the car utilization can be improved (reducing costs for the driver). Furthermore, the result quality of timetable information systems can be enhanced by additional optimal connections which use dynamic ride-sharing.

Our contribution

In this work, we address two problems to combine dynamic ride-sharing and public transport: connecting public transport stations by dynamic ride-sharing connections and connecting start and destination of a query to public transport stations by dynamic ride-sharing routes. To solve the first challenge we add station-to-station connections that are derived from dynamic ride-sharing offers to our graph representation in a preprocessing step. These connections can be updated when matches are agreed upon, offers are withdrawn, or new ones are available. The second challenge requires the selection of suitable ride-sharing offers that allow for a connection from the start location to a station or from a station to the destination location. Since the source and destination adress are given in the individual queries and are not known beforehand, in contrast to the set of stations, this step needs to be carried out on-the-fly for each query.

State of the Art

In the literature different multi-modal search algorithms combining public and private transportation have been described, see the recent survey [1]. There are also commercial systems available[1]. To the best of our knowledge neither of these find optimal connections combining public transport and dynamic ride sharing.

1 Such as http://www.fromatob.de/ and https://www.qixxit.de/en/

Travel Information System

To enable the search for optimal connections combining public transportation as well as ride-sharing we extend our existing travel information system *TIS* which computes journeys for a given user query that contains start and destination location, start time interval, and transportation mode selection. The resulting journeys are Pareto-optimal regarding the optimization criteria of travel duration and number of interchanges. All computed journeys share the following structure: first, from the start location, private transportation is used to reach a station of the public transportation network. Second, with public transportation the passenger travels to a station near the destination. Last, from there the destination location is reached by private transportation again.

Primarily, TIS computes connections in the public transportation network including potential walks between stations which are close to each other. For the private transportation parts of the multi-modal connections, we use a third-party routing service, OSRM [5].

Here, we briefly explain the time-dependent graph model in TIS. This simplified description follows Disser et al. [2]. In the timetable, there is a set of transports $\mathscr{T}$ and a set of stations $\mathscr{S}$. Each transport $t \in \mathscr{T}$ consists of a set of elementary connections $\mathscr{E}(t)$. Hence, an elementary connection models a train run from a station to another station without intermediate stops. The set of all elementary connections of all trains is denoted by $\mathscr{E}$. In a time-dependent graph $G = (V, E)$, for each station $s \in \mathscr{S}$, there is a *station node* $v_s \in V$ in the graph. There is an edge $e = (v_a, v_b)$ between two station nodes $v_a, v_b \in V$ if there is at least one elementary connection $c \in \mathscr{E}$ from station $a \in \mathscr{S}$ to station $b \in \mathscr{S}$. For a correct modeling of interchanges the graph contains additional node and edge types which are skipped here. A full description can be found in [2].

Each edge $e = (v_a, v_b) \in E$ has a duration and a cost. The duration is time-dependent and is determined during the search: If the edge is used at time τ, the duration of the edge equals the difference between the arrival time (at the head station of the edge) of the first connection departing after τ and time τ.

There are also *foot* edges in the graph to allow walkings from a station to other stations within walking distance. These edges are not time dependent but return a constant edge cost representing the time required to reach the other station by foot.

Acknowledgements

This work was supported by German Railways Deutsche Bahn AG. Public transport schedules were kindly provided by German Railways Deutsche Bahn AG. Anonymized, real dynamic ride-sharing queries and offers were kindly provided by flinc AG.

2 Dynamic Ride-Sharing

In this section, we will introduce our profit model for drivers, and describe the scenarios “connecting public transport stations” and “connecting start and destination of a query to public transport stations” by dynamic ride-sharing routes. In the first scenario, stations in our public transportation graph are connected to each other. In the second scenario, an address provided by the user is connected with a public transport station.

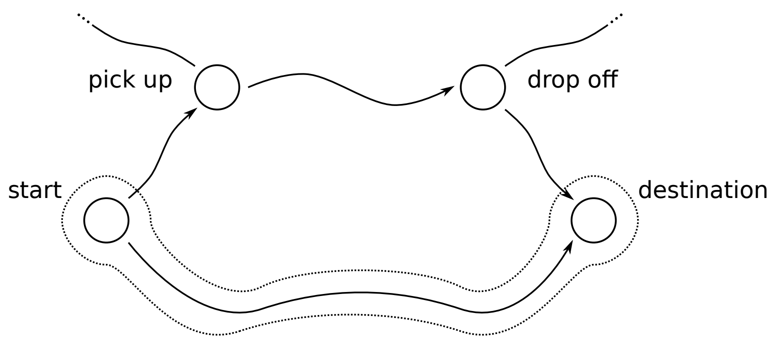

Figure 1: Rerouting a driver. The driver’s journey is from *start* to *destination*. A potential dynamic ride-sharing match reroutes the driver via *pick up* and *drop off*. Note, that pick up and drop off do not need to be located close to the original route (dotted area) of the driver as long as the constraints regarding detour and profit hold.

As shown in Figure 1, a driver provides a route from a *start* location to a *destination* location together with his departure time. Passengers are interested in a ride that does not necessarily start and end at the driver’s start and destination. Therefore, a passenger has to be *picked up* and *dropped off* at his / her start and destination location. The driver may accept the detour in order to give the passenger a lift. She/He is rerouted to visit the pick up and drop off locations. The resulting times at these locations determine the availability of the dynamic ride-sharing connection and act as a schedule. To account for delays caused by heavy traffic, a safety margin is added to the duration of all ride-sharing connections, where the drop off location is not the final destination of the passenger.

The passenger bears part of the driver’s costs. We assume a reasonable driver who maximizes the profit as defined in the following equation. By “dur_detour”, we denote the difference between the duration of the original journey (from driver’s start to driver’s destination) and the duration of the route via pick up and drop off. For the difference in distance we define “dist_detour” analogously. By M, we denote the cost per kilometer the driver has to pay for upkeep and gas. An hourly wage, denoted by W, compensates for the time required to take the detour.

$$\begin{aligned} \text{driver\_expenses} &= \text{dist\_detour} \cdot M\ \text{ct}/\text{km} + \text{dur\_detour} \cdot W\ \text{ct}/\text{min} \\ \text{profit} &= \text{passenger\_ contribution} - \text{driver\_expenses} \end{aligned}$$

We define an upper bound for the detour duration and distance and a lower bound for the driver's profit. Ride-sharing connections that do not satisfy these bounds are skipped since they are not attractive for drivers. In order to reduce the effort required to find reasonable reroutings, we conduct pre-computations using the straight line distance. Further calculations can be skipped if the results of those pre-computations already do not match the driver's criteria.

Station to Station

Our system enables ride-sharing connections from one public transport station to another. After this ride, the passenger continues her/his journey with another ride-sharing connection, public transportation or an individual transport to her / his destination location. Using an additional service "on-the-fly" during the search does not yield reasonable query times. For performance reasons the internal graph representation needs to be extended to contain all reasonable potential ride-sharing connections. Ride-sharing can therefore be seen as supplement to public transportation. In the following, this use case is referred to as *supplement* scenario. The driver has an own "schedule" and every potential pick up and drop off location combination can be seen as an elementary connection. Therefore, all reasonable routes from pick up to drop off stations can be added to the set of elementary connections $\mathcal{E}$. One advantage of the supplement case is that all data is available statically beforehand.

Therefore, we can preprocess all available driver's routes (start, destination, and time) in order to integrate all reasonable dynamic ride-sharing connections into the backbone graph. This saves valuable time when calculating optimal journeys for the user. The sheer amount of public transport stations leads to many potential pick up / drop off station pairs. Many smaller stations only lead to marginally different connections. Therefore, we decided to focus only on those stations that are served by trains (not only busses, trams, or subways).

Address to Station and Station to Address

In case of connections from or to user provided locations we need to gather ride-sharing matches on-the-fly. Those calculations cannot be carried out beforehand and are at the same time time-critical because the user is waiting for the system to respond. Since this is a special case of "*door* to *door*" routing we refer to this case as the *door* scenario.

The basic approach is to gather all potentially useful ride-sharing offers that could take the passenger from her / his provided start location to a public transport station or from a public transport station to her / his provided destination

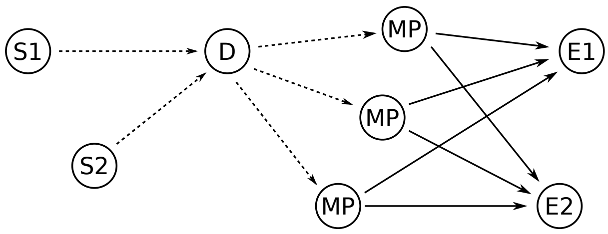

Figure 2: Rerouting the driver in the door scenario. Example situation for the passenger's departure **D**. Two offers (**S1** to **E1**, **S2** to **E2**) and three meeting points **MP** have been identified. Dotted distances need to be routed on the fly, solid ones are known from a preprocessing step.

location. All these connections are then temporarily integrated into the graph. The algorithm then finds connections that contain these rides if this yields an optimal connection.

It is beneficial for the system's performance to make use of data generated in the supplement preprocessing. We assume that the positions of all previously selected public transport stations are about evenly distributed. Therefore, we are able to select relevant ride-sharing offers very fast. Ride-sharing offers that could take the user from a public transport station to her / his destination location can easily be determined by gathering the rides with a drop off station in the area of the provided destination location. As shown in Figure 2 only the driving distances from the offer's start location via the user's location to the meeting point[2] need to be recomputed. Relevant ride-sharing connections from the user's start location to public transportation stations can be determined analogously.

The direct connection case can be seen as a special form of the *door* scenario. All ride-sharing offers that have pick up stations near the provided start location and drop off stations near the provided destination location are suitable to enable a direct ride-sharing connection if the departure time is contained in the user provided interval.

3 Computational Study

Our prototype of the information system is implemented in C++ and evaluated on an Intel Xeon 3.4 GHz[3] quadcore CPU with 32 GB main memory. The public transport schedule were kindly provided by the German railway company, Deutsche Bahn, and contains all long-distance and regional trains in Germany as well as busses, trams, and underground trains of selected transportation au-

[2] for example parking areas or easily accessible places
[3] Intel(R) Xeon(R) CPU E3-1245 V2 @ 3.40GHz

thorities. The schedule contains 106,000 stations which results in 1.6 million nodes in the graph. Furthermore, the graph contains 6 million edges, where 1.4 million model train operations and the rest enables interchanges or walks between stations. The 6,455 long distance travel stations[4] in the schedule are selected as meeting points. The overall time window for the schedule was restricted to two weeks.

In our evaluation, we used real dynamic ride-sharing offers and real customer queries kindly provided by the dynamic ride-sharing service flinc. The data is anonymized to protect customer privacy: all coordinates are rounded to two decimal places. The dataset contains about 2,500 offers and more than 2,000 queries. The offers and queries are either one-off or defined by an interval and a weekly recurrence pattern. For our evaluation, we mapped the queries and offers to the corresponding weekdays in our time period of two weeks. This results in 10,000 (temporally) unique offers and 7,000 queries. Street-routing is provided by OSRM [5], which operates on data from OpenStreetMap[5].

Preprocessing

The approach for the supplement scenario relies on preprocessing, where street-routing has the largest performance impact. However, we profit from the fact, that the backbone of our system, the public transport stations, is static. Therefore, we choose to compute all possibly required distances[6] in one go and perform the actual processing of route approval according to the driver's profit model and edge generation afterwards.

The street-routing step is completed in just under two hours for the complete evaluation dataset, while adding another offer would take only ten seconds. The actual preprocessing runs in one minute and ten seconds (thereof 25 seconds for routes approval, 42 seconds for edge generation).

Evaluation Approach

The computational study focuses on the extension of our existing information system with ride-sharing. As baseline, we configure our system with the public transport schedule and walkings up to 15 minutes from the start to the "first station" and from the "last station" to the destination. The evaluation is carried out in four different configurations: *no ride-sharing* at all; ride-sharing in the *supplement* scenario; ride-sharing in the *door* scenario; the *full system*, with both scenarios. Two questions are examined: How is the result quality affected in the various scenarios and what are the performance figures that directly impact the user experience?

4 every station served by trains; not only busses, trams, and subways

5 Project OpenStreetMap, `https://www.openstreetmap.org`

6 all offer starts to all meeting points, all meeting points to all offer ends, and the full meeting point distance matrix.

In order to assess the result quality, the query set is partitioned: The first subset contains all queries for which the baseline delivers a connection. These are 81.6% of all queries, and they will be used to judge whether combining ride-sharing with public transport results in connections that are better compared to connections that only use public transport. In the rest of this paper, we denote such better connections as *better alternatives*.

For other 14.7% of all queries, the baseline does not deliver a connection. For them, combining dynamic ride-sharing with public transport is a chance to *enable journeys* in the first place. However, for the remaining 3.7% of the queries no matching offers are available, which marks them as the unanswerable upfront. These queries are not considered in the rest of this study.

Better Alternatives

In this evaluation, the information system processes the first subset of the query set, queries that can be answered by the baseline. Table 1 presents the improvements, when ride-sharing is included in the different configurations.

It has to be noticed, that the door scenario yields significantly more improvements than the supplement scenario. Yet, even for the supplement scenario, we could lower the minimum number of required interchanges for over a quarter of the queries by 1. In the door and full scenarios the fastest connection could be improved in over a fifth of the queries, on average by more than 80 minutes. For over a third of the queries, we were able to save one interchange on the connections with the least number of interchanges.

On the one hand, the high saving potential results from a number of moderate improvements. On the other hand, some results, where the information system barely found a connection without ride-sharing[7], benefited significantly from the new possibilities, and were turned into highly attractive travel opportunities. This shows that a system incorporating dynamic ride-sharing improves the overall quality of the connections offered to the users.

Enable Journeys

In this evaluation, the information system uses ride-sharing to enable journeys, which are impossible with public transport alone (second subset of the query set). Table 2 confirms that this endeavor is successful in many cases. In the range-extending door-scenario 94% of these queries can now be answered. Unfortunately, the supplement scenario does not fare well at all.

To assure that the newly possible journeys are valuable to the passenger, their travel times were compared to twice the time required to travel from the start to the destination by car. For about a third of the queries the fastest connection does not exceed this bound and, thus, is considered high quality.

[7] Especially, when using door-scenario connections in the late hours of the day and to bridge areas with only sparse near distance transportation.

	criterion: duration		criterion: transfers	
	% better	avg. impr.	% better	avg. impr.
suppl.	6.0%	26.4 min	28.0%	1.18
door	21.3%	86.7 min	37.1%	1.27
full	22.3%	83.4 min	44.4%	1.33

Table 1: Improvements achieved by including dynamic ride-sharing. Rows show the improvements over the reference system without dynamic ride-sharing. Columns "% better" give the percentage of queries with better results. Columns "avg. impr." show the average improvement over these responses.

	new results	high quality
new in suppl.	2.2%	1.0%
new in door	94.1%	33.1%
new in full	94.3%	33.3%

Table 2: New results and their quality. Column "new results" shows the amount of newly enabled journeys. Column "high quality" gives the percentage of journeys with at most two times the car travel duration.

Information System Performance

Table 3 shows the processing times in the different configurations. Either the supplement or the door scenario roughly double the computation time. The supplemental scenario is noticeably faster. The combination of both is slightly slower than three times the time required by the baseline version.

All in all, the performance impact of the various scenarios is definitely noticeable by the user. However, the average processing time of under seven seconds for the full configuration stays within reasonable bounds.

	none	suppl.	door	full
average [s]	1.8	3.5	4.1	6.2
90% quantile [s]	2.5	7.2	6.1	11.4

Table 3: Computation times of the graph search in the various scenarios.

4 Conclusion

We presented and evaluated the integration of dynamic ride-sharing into our multi-modal travel information system.

The extended system is capable to find optimal connections combining public transport and dynamic ride-sharing. These connections use dynamic ride-sharing either between public transportation (supplement scenario) or from the start or to the destination address (door scenario). The computed connections are Pareto-optimal in terms of travel duration and number of interchanges, while the price is also taken into account. Within reasonable computation time, we find additional attractive connections. Either alternatives to connections consisting only of public transportation or new connections for queries that could not be answered successfully without dynamic ride-sharing. Many of these connections are able to compete with individual modes of transportation.

While we have shown that ride-sharing is a valuable addition, its success depends on the available offers. The following aspects warrant further investigation: First, including offers with via locations and allowing multiple passengers per car, which introduces new constraints on the driver's schedule and driver-passenger matching. Second, implementing an online mode, which re-routes the driver during an ongoing journey. This could increase coverage and even might allow to reroute passengers on their public transport journey in case of connection failures.

References

[1] Hannah Bast, Daniel Delling, Andrew V. Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks. *CoRR*, abs/1504.05140, 2015.

[2] Yann Disser, Matthias Müller-Hannemann, and Mathias Schnee. Multi-criteria shortest paths in time-dependent train networks. In *Experimental Algorithms*, pages 347–361. Springer, 2008.

[3] Lennart Frede, Matthias Müller-Hannemann, and Mathias Schnee. Efficient on-trip timetable information in the presence of delays. In *ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems, Karlsruhe, Germany*, 2008.

[4] Felix Gündling, Mohammad Hossein Keyhani, Mathias Schnee, and Karsten Weihe. Fully realistic multi-criteria multi-modal routing. *tuprints - TU Darmstadt publication service*, December 2014.

[5] Dennis Luxen and Christian Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 513–516, New York, NY, USA, 2011. ACM.

[6] Matthias Müller-Hannemann and Mathias Schnee. Finding all attractive train connections by multi-criteria pareto search. In *Algorithmic Methods for Railway Optimization*, pages 246–263. Springer, 2007.

[7] Matthias Müller-Hannemann, Mathias Schnee, and Karsten Weihe. Getting train timetables into the main storage. *Electronic Notes in Theoretical Computer Science*, 66(6):8–17, 2002.