Fully Realistic Multi-Criteria Multi-Modal Routing*

Felix Gündling, Mohammad H. Keyhani,
Mathias Schnee, and Karsten Weihe
Technische Universität Darmstadt, Algorithmik†

December 10, 2014

Abstract

We report on a multi-criteria search system, in which the German long- and short-distance trains, local public transport, walking, private car, private bike, and taxi are incorporated. The system is fully realistic. Three optimization criteria are addressed: travel time, travel cost, and convenience.

Our algorithmic approach computes a complete Pareto set of reasonable connections. The computational study demonstrates that, even in such a large-scale, highly complex scenario, appropriate speed-up techniques yield an acceptable query response time.

1 Introduction

Most journeys do not start and end at public transportation stations, but at two different street addresses, and require several modes of transportation: walking, driving a car, riding a bike, and the like, on one hand; local and long-distance public transportation (bus, streetcar, train, etc.) on the other hand. Connections that combine several modes of transportation are usually called *multi-modal* in the literature. Three optimization criteria are commonly regarded as relevant: total duration of the journey, convenience, and total cost of the journey.

Our Contribution We present empirical results from an algorithmic approach along with some speed-up techniques for the multi-modal routing problem, more specifically, for the fundamental case that long-distance and local public transportation is the backbone of a connection. In our data, this backbone comprises all short- and long-distance trains operated by Deutsche Bahn, the national German train company, as well as a large number of buses, streetcars, city railways, subways, etc., from many regions of Germany. Moreover, our prototype allows the following *private* transport modes: walks, rides with own car or bike, taxi rides, and courtesy lifts by a friend or relative. It is fully realistic in the sense that all choices and options offered by the travel information system of Deutsche Bahn to specify a query, are basically offered by our prototype as well: The users can specify a start location, a target location, and a departure time; the modes of private transportation may be deliberately excluded by the user. Moreover, the above-mentioned optimization criteria, travel duration, travel cost, and convenience (number of vehicle changes) are incorporated. Note that even the travel cost alone is a highly complex criterion in its own right.

For a query, our prototype computes a set of Pareto optima that is complete in the following sense: For any weighting of the three criteria, the best reasonable solution can be found in the output. Here, we regard a solution S as unreasonable if there is another solution S' such that: Compared to S', solution S is cheaper but has a longer duration,

*This work was partially supported by German Railways Deutsche Bahn AG (RIS).
†{guendling,keyhani,schnee,weihe}@algo.informatik.tu-darmstadt.de

and the cost difference is negligible compared to the difference of the durations. This design decision only excludes connections that are indeed unreasonable from any user's perspective, and has reduced the query response times significantly.

The scenario we address is by far more complex than any multi-modal scenario that we have seen in the literature. We see the scientific value of this paper in the empirical proof that appropriate algorithmic techniques compute a set of reasonable Pareto optima with a reasonably short runtime of our *backbone algorithm*, even in such a complex scenario. This is a major step towards our goal to disprove the claim in [5], that multi-criteria search is generally too slow. More specifically, given a set of start stations and a set of target stations, and given two sets of private transportation routes from the start address to the start stations and, respectively, from the target stations to the target address, our backbone algorithm performs a multi-criteria multi-source multi-target search and delivers a set of Pareto optima. In Section 4.2, we will argue that even the problem of finding private transport routes can be managed with an acceptable runtime.

Related Work To our knowledge, all scenarios in the literature are by far less complex in some way or other. Some work focuses on metropolitan areas [2, 5, 12, 15]; in work beyond metropolitan areas, the networks are also smaller by orders of magnitude [3, 9].

Most approaches only optimize one criterion, usually the travel duration [3, 7, 10]. Some other approaches replace the simultaneous optimization of several criteria by optimizing a weighted sum of the criteria, where the weightings of the criteria are fixed [1, 12, 15]. The solution computed by such an approach need not even be Pareto optimal (we refer to [4]). The approach presented in [6] excludes the private modes of computation (car, bike, taxi, etc.) from the multi-criteria search and thus computes durations only for these parts of a journey. On the other hand, the approach presented in [9] separates the considered modes of transportation from each other. Moreover, in contrast to multi-criteria approaches like [2, 5, 6, 7], we consider realistic prices as an additional Pareto criterion.

Bast et al. presented methods to filter large sets of Pareto optimal solutions [2]. Their method as well as the fuzzy filtering method presented by [5] can be applied to filter solutions delivered by our approach.

Our Concept We have already presented promising approaches to perform multi-criteria search on large train networks [8, 13]. The work presented now extends that approach to the multi-modal case. In particular, a time-dependent graph model [8] is the basis. We compute multi-modal connections which consist of three parts: First, a *private* mode of transportation in order to get to a station. Next, in the middle, a succeeding *public transportation* connection as the main part of the connection. Finally, a private mode of transportation in order to get to the target address. By *private* transportation, we mean transportation modes like car, taxi, bike, and walking.

We do not allow private modes of transportation in the middle of public transportation. This design decision does not reduce generality. In fact, walking from one station to another one between two rides of public transport is the only non-negligible case, and we cover this case by introducing foot edges into our graph model for the backbone.

Overview In Section 2, we present our model to combine public transportation with private transportation. We also explain our price model. Our algorithm and speed up techniques are explained in Section 3. In Section 4.1, we evaluate our prototype to compute multi-modal connections. Finally, the paper finishes with a conclusion and a motivation for future work.

2 Model

The backbone of our multi-modal routing prototype is *TIS*, a fully realistic timetable information system [8]. 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 [11].

2.1 Graph Model for the Backbone Part

Here, we briefly explain the time-dependent graph model in TIS. This description follows Disser et al. [8]. In the timetable, there is a set of trains $\mathcal{T}$ and a set of stations $\mathcal{S}$. Each train $t \in \mathcal{T}$ consists of a set of elementary connections $\mathcal{E}(t)$. 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 $\mathcal{E}$. In a time-dependent graph $G = (V, E)$, for each station $s \in \mathcal{S}$, 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 $c \in \mathcal{E}$ from station $a \in \mathcal{S}$ to station $b \in \mathcal{S}$.

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 t, the duration of the edge equals the difference between the "earliest arrival time at the head station of the edge" and t. The earliest arrival time is the arrival time of the elementary connection from station a to station b which has the earliest departure time later than t.

There are also *foot* edges in the graph to allow walkings from a station to other stations within walking distance. Our graph model incorporates minimal transfer times between trains at the same station as follows. For each station, a general minimal time is defined, which estimates the time required to walk from one platform to another one. Moreover, we support special transfer time rules between trains as predefined in the timetable. Such a special transfer time rule can allow shorter transfer times or force longer transfer times depending on the true distance between two platforms. For further details, we refer to [8].

So, our model is fully realistic. In particular, there are no simplifying assumptions such as periodicity; in fact, in our timetables there are many long-distance trains which run only once or a few times per day. Trains do not even operate every day but some trains only operate on specific days. The frequencies of many lines depend on the time of day. Many lines cease operating in the evening or night hours. We handle all these irregularities.

2.2 Supporting Various Modes of Private Transportation

The following private modes of transport are integrated:

- **Own Car:** The traveler can use her/his own car to drive from her/his start position to a station. Besides the driving time, we take into account a time penalty to find a parking facility and a fixed realistic price per kilometer. Clearly, the own car is only available at the starting point of the journey.

- **Courtesy Lift:** The traveler may get a lift by a friend or relative in order to move from her/his start address to a start station or, respectively, to move from a target station to her/his target address. Here, the traveler does not need to find a parking place. But at least for our prototypical work, it is certainly reasonable to simply assume double driving cost since in most cases, the driver has to return by car to the start address.

- **Taxi:** A taxi can be used before or after the public transportation connection. A taxi does not need a parking facility but is more expensive. In addition, a taxi has a base price, which is independent of the driving distance.

Transport Mode	Time Penalty	Base Price	Price per km	Avg. Speed (km/h)
Own Car	8	0	30	80
Courtesy Lift	0	0	60	80
Taxi	0	250	180	80
Bike	5	0	0	15
Walk	0	0	0	5

Table 1: Supported modes of private transportation. Times are given in minutes and prices are given in euro cent. "Time penalty" is the time required to park a car/bike and walk to the station.

- **Own Bike:** The traveler can ride her/his own bike to get to a start station. Riding a bike is costless, but a parking place is needed again. Analogously to the own car scenario, the own bike is available only at the beginning of a multi-modal connection.
- **Walking:** It is possible to walk to stations which are within walking distance. Walking is costless and no parking is needed.

For each private transportation mode we define a maximum travel duration (separately for the start and for the end part of a journey). This threshold can be set by the user. In our experiments, we use a maximum travel duration of 15 minutes for all private modes of transportation. Parking, base price, price per km, maximum duration, average speed, etc., are parameters whose values can be specified for each transport mode separately. In our computational study, we choose reasonable values for these parameters (see Table 1).

The routes for the private transportation modes are found in the road network using Open Source Routing Machine (OSRM) [11]. Based on OpenStreetMap data [14], OSRM computes routes from a start position to a target position.

2.3 Price Model

Our price model approximately equals the real tariffs. As a basis, we use distance based prices for public and private transportation. Here, the prices depend on the train categories for public transportation and on the transport mode for private transportation. Moreover, we support special tickets for fast trains and local transportation: for instance in Germany, there are special tickets which allow a cheaper travel when using specific train categories.

Our price model can be configured with arbitrary parameters and prices. For or computational study in this paper, we use 15 / 18 / 22 cents per kilometer for local / long-distance / high-speed trains. The used prices for the supported private transport modes are illustrated in Table 1.

2.4 Query

A query to our prototype has to specify the start and target location (addresses or GPS coordinates) for the search. In addition, the query has to specify a time interval for the start time of the connections. Each private transport mode may be excluded by the user. For every non-excluded private transportation mode, an individual limit on the maximum travel duration is to be defined by the user as well. Finally, the user has to specify values for the above-mentioned parameters such as average speed, time for parking, base price, distance based price, etc. (clearly, in a real-world application, all of these values would be pre-defined by an operator; the passenger may or may not be granted the right to adjust some of the values to her/his personal situation).

3 Algorithm

As mentioned in Section 1, a multi-modal connection computed by our algorithm has the following structure: It consists of a public transportation connection, enclosed by two private transportation connections before and after the public transportation part. Given a query as specified in Section 2.4, we search for multi-modal connections as will be explained in this section.

We start with a brief outline: At first, a set of start stations as well as a set of target stations are preselected using Euclidean distances. Then, we compute private transportation routes from the start address to each preselected start station. Analogously, we compute routes from each preselected target station to the target address. Based on that, the set of all start stations and the set of all target stations that are actually relevant for the query are determined. Now all data for the multi-criteria search for the backbone connection is available. The multi-criteria search is then performed by TIS, which receives the auxiliary data and delivers a Pareto optimal set of reasonable multi-modal connections.

3.1 Determining the Potentially Relevant Start and Target Stations

We need all stations that can be reached from the start address with private transportation within the maximal travel duration of the individual mode of transportation. Analogously, we need all stations from which the destination may be reached within the maximal travel duration.

For this purpose, at first, we select all stations such that the Euclidean distance from the start address (analogously, to the destination) divided by a conservative estimation of the speed does not exceed the maximal travel duration. This way, for each private transport mode, we get a set of stations which are potentially relevant. In a second step, we select those which are actually relevant for the search, as explained in Section 3.2.

3.2 Private Transportation Routes

Once the potentially relevant start and target stations are preselected (for each transport mode), we compute the private transportation routes. These are the routes from the start address to each start station and from each target station to the target address. Note that each route has to obey the maximum travel duration for the mode of private transport which the route uses. Since the stations are preselected via Euclidean distances, there could be routes which have an actual duration longer than allowed. These routes will be dropped.

The routes are computed using OSRM [11], which delivers the duration and the distance for each route (see Section 2.2). The costs for the routes are computed in a subsequent step. The data about the durations and costs of all computed private transportation routes is delivered to our timetable information system TIS.

3.3 Multi-Criteria Search using TIS

We use TIS to perform the multi-criteria search from the start address to the target address. As auxiliary data, TIS gets for each start and target station a list of tuples of duration and price, representing the private transportation routes. Using this data and based on the public transportation network, TIS computes Pareto optima among all reasonable connections.

Let N and M denote the sets of start and target stations. Let T_{start} and T_{target} denote the number of supported private transport modes at start and target. A naive approach to find multi-modal connections is to perform a search from the start address to the target address via each possible combination of start and target stations as well as each possible mode of private transport. This approach requires up to $T_{\text{start}} \cdot T_{\text{target}} \cdot |N \times M|$ searches. Especially, in cities with dense public transportation networks a large number of start and target stations has to be considered. Consequently, this naive approach is very inefficient and even impracticable for most queries.

The first step to reduce the complexity is to create at each start station, one start label for each private transport route. For instance, if a start station can be reached from the start address using a car or a bike, we create two start labels at this station: one start label for using a car and one start label for using a bike. The labels used during the search have to carry information about the private transport route which has been used to get to the start station. When labels are compared and filtered during the search, the data about the private transportation of each label has to be taken into account. This technique reduces the complexity, but there are still up to $T_{\text{target}} \cdot |N \times M|$ searches necessary. Therefore, we used further speed-up techniques as follows.

Multi-Source Search Instead of performing a new search per start station, we extend the single-source Dijkstra algorithm to a multi-source search. Start labels are created at all start stations for each transport mode. All these start labels are inserted into the priority queue. No modification of the search algorithm is necessary to handle labels with different start stations. A multi-source search has more start labels and is more expensive than a single source search. But this approach reduces the number of necessary searches to at most $T_{\text{target}} \cdot |M|$.

Multi-Target Search We use a further technique to extend TIS by multi-target search. First, we create a virtual node representing the target address. Then, all target stations are connected to this virtual node by creating a virtual edge for each private transportation route (only for routes from the target stations to the target address). Each virtual edge can be used at any time, and its duration and price equals the duration and price of the route which it represents.

Using virtual edges a multi-target search is possible. A multi-source multi-target search obtains the same set of Pareto optimal solutions as when performing single-source single-target searches. Self-evidently, it is more complex and sophisticated. But since the number of searches is reduced to 1, this approach results in obviously better runtimes. Computational results are presented in Section 4.

Note: The idea to introduce virtual edges can also be used in order to realize multi-source search. But this approach would imply an undesired effect: All connections would directly start with the virtual edges and have the same departure time. Consequently, instead of optimizing the travel duration, the algorithm would solve the earliest arrival problem given a fixed departure time. Furthermore, compared to our presented method for multi-source search, virtual edges at start have no advantages w.r.t. to the runtime. After processing the virtual edges, at each source station and for each private transportation route, a label would be created. This is a state which we obtain with our method without virtual edges.

4 Evaluation

4.1 Dataset and Test Queries

Dataset For street routing, we use OpenStreetMap data of the German road network [14]. The public transportation network data is provided by German Railways Deutsche Bahn AG (RIS) and contains all long-distance and city railways. Furthermore, the dataset contains local public transport like buses and streetcars of many local transport associations (RMV, VBB, VRR, VRM, etc). These cover nearly all metropolitan regions of Germany. The timetable schedule contains about 630k trains and 81k stations. The graph comprises 1,257,921 nodes and 3,563,427 edges including about 30k *foot* edges connecting stations that are relatively close to each other.

Test Queries In order to evaluate our approach, we generated routing queries based on real customer requests. These real customer requests are provided by German Railways, Deutsche Bahn AG (RIS), and are station to station (pure railway) requests. For each customer request, for the start as well as for the target station, we picked random coordinates from a 30km radius around the station. If both, source and target, are located in a region where data about local public transportation is available, the new query is added to the set of our test queries. The departure time interval is always one hour. The start time of the interval originates from the base customer request.

We generated three different query sets Q_i where $i \in \{1, 2, 3\}$. Here, i is the number of different modes of private transportation which are supported at the start of the journeys. The transportation modes are selected randomly from the modes introduced in Section 2.2. They all have different trade-offs. E.g. a courtesy lift is faster than driving with the own car, because it is not required to search for a parking place. On the downside, it requires the driver to head back and is therefore twice as expensive as using the own car. Each of the query sets contains 1,946 queries.

For the end of journeys, we randomly selected up to 2 different transport modes. The reason is that private transportation at target is modeled using virtual edges, and the number of virtual edges is negligible compared to the size of the graph. Therefore, in our computational study we focus on evaluating the algorithmic complexity of handling different private transport modes at the beginning of journeys.

4.2 Computational Results

In this section, we evaluate the performance of our C++ implementation of the core algorithm. The code was compiled with GCC (version 4.7.2) with the `-O3` and `-march-native` flags enabled. The evaluation was carried out on a Intel(R) Xeon(R) E3-1245 V2 CPU (4 cores but only one core in use), running at 3.40GHz. The machine has 32GB of main memory.

In our main evaluation, we have computed Pareto optimal sets of multi-modal connections for all queries in query sets Q_1,Q_2, and Q_3 (5,838 queries in total). The performance of our algorithm is presented in Table 2. Here, we only consider the runtimes of our own core algorithm. The reason is that we want to evaluate the quality of our multi-source multi-target multi-criteria search.

On average our implementation required 3,322 street routing requests for one routing query. In this evaluation, the OSRM street routing service was used as a web service. Network latencies can be eliminated by using the OSRM algorithm as a library. Since these computations are independent from each other they can be carried out in parallel. According to [11], one street routing request can be processed within 12ms (on average for the Germany dataset). For instance, when using 8 cores, the average runtime for the routing requests will be less than 5 seconds.

Note: Some different private transport modes like taxi, car, or "'courtesy lift"' use the same route. Therefore, many routing requests w.r.t. to the different private transport modes would result in similar requests to the routing service. Here, a reuse of street routing results can decrease the number of routing requests and obviously improve the overall runtime.

As depicted in Table 2, our efficient implementation of the core algorithm requires 5.76s on average to compute one routing result. This is quite reasonable with respect to more than 2,5k different possibilities to start the journey. Our evaluation showed that the core algorithm can answer 90% of all queries within less than 11.2s.

4.2.1 Number of Supported Modes of Private Transportation

In this section, we evaluate the influence of an increased number of different modes of private transportation at the start of the journeys. For this study, we generated two

	Average	90% Quantile
Runtime (ms)	5,761	11,200
# Street Routing Requests	3,322	6,775
# Labels Created	717,538	5,705,654
# Start Options	491	1,257
# Start Labels	2,548	7,397
# Edges at the Destination	328	845

Table 2: Statistics for queries with one, two, and three options for getting from the source location to a departure station. "Start Options" denotes the total number (sum for each station) of possible options to start the journey. E.g. if station A can be reached by foot and by car and station B can only be reached by car, there are three start options.

additional query sets with eight different transport modes, denoted by $Q_{8\mathrm{r}}$ and $Q_{8\mathrm{a}}$. Query set $Q_{8\mathrm{r}}$ contains queries which allow eight realistic options (see Table 3). For $Q_{8\mathrm{r}}$, in addition to the modes introduced in Section 2.2, we use the following modes for our evaluation:

- **Bike Sharing:** This option simulates the usage of a bike sharing service. Since the traveler first needs to walk to the bike sharing station and would not be able to park the bike at the train track, we add a time penalty of 15 minutes. We assume that the service provider charges 10 cent per kilometer. In our computational study, we use the mode "bike sharing" instead of using the mode "own bike".
- **Motorbike:** On the one hand, the motorbike is slower than the car. On the other hand, it is cheaper and has a lower time penalty.
- **"Own Car (slow)" and "Courtesy Lift (slow)":** Compared to the modes "Own Car" and "Courtesy Lift" as defined in Section 2.2, these options represent other routes or an environment-friendly style of driving. Many navigation systems do not just offer one possible route but present different possibilities to the user (e.g. a cheap route, a fast one and one that is in between).

Query set $Q_{8\mathrm{a}}$ contains queries which allow eight artificial trade-offs based on the initial car type. To generate these, we gradually increase the speed and the price. The exact values are depicted in Table 4.

Each of the query sets $Q_{8\mathrm{r}}$ and $Q_{8\mathrm{a}}$ contains 1,000 queries. For all queries in query sets Q_1, Q_2, Q_3, $Q_{8\mathrm{r}}$, and $Q_{8\mathrm{a}}$, we have computed multi-modal connections. The results of this computational study are presented in Table 5. A comparison of the results of the different query sets shows, that adding new modes of private transportation is not very expensive. The average runtime of queries with eight realistic modes is only 224ms higher than the average runtime of queries with three modes. This shows that our prototype can support reasonable numbers of modes and has no restrictions in this context.

Multi-Slot Labels For queries with more than two allowed modes of private transportation and if many of the possible modes are used to get to most of the start stations, we may improve processing times by condensing labels. Instead of using one label for each different mode of private transportation, we use only one label with multiple slots: one slot per mode of private transportation. Dominance between multiple slot labels is then executed as follows: Any slot of one label may dominate any slot of another. If all slots are dominated in a label this label is discarded.

We propose using the standard variant by default and only switching to the multiple slot variant if the criteria mentioned above are met. Thus, we may improve running times by 5% to 26%.

Transport Mode	Time Penalty	Base Price	Price per km	Max. Duration	Avg. Speed (km/h)
Foot	0	0	0	15	5
Bike Sharing	15	0	10	15	15
Motorbike	6	0	15	15	30
Own Car	8	0	30	15	80
Own Car (slow)	8	0	25	15	60
Courtesy Lift	8	0	60	15	80
Courtesy Lift (slow)	8	0	50	15	60
Taxi	0	250	180	15	80

Table 3: Eight realistic modes of private transportation used in query set Q_{r}.

Transport Mode	Time Penalty	Base Price	Price per km	Max. Duration	Avg. Speed (km/h)
Car 1	8	0	33	15	85
Car 2	8	0	30	15	80
Car 3	8	0	28	15	75
Car 4	8	0	25	15	70
Car 5	8	0	23	15	65
Car 6	8	0	20	15	60
Car 7	8	0	18	15	55
Car 8	8	0	15	15	50

Table 4: Eight artificial modes of private transportation used in query set Q_{a}.

4.2.2 Multi-Source Multi-Target

In this section, we show the run time improvement achieved using the multi-source multi-target search, as introduced in Section 3.3. For this purpose, we evaluated 4 different approaches: The first approach performs single-source single-target searches via all start and target stations and all supported private transport modes. The second approach combines multi-source search with single-target search. The third approach combines single-source search with multi-target search. Finally, the fourth approach realizes the multi-source multi-target search. The performance of each of these approaches are shown in Table 6.

For this evaluation, we have selected a number of 1,500 queries from Query sets Q_1, Q_2, and Q_3. Since the single-source and the single-target search require a large number of requests, for each analyzed approach, we have processed a subset of the queries. For that, we determined the set of start and target stations for each query. Then, we skipped queries which had a number of source and target stations larger than a threshold. Let M denote the set of start stations and N the set of target stations. The threshold for each approach can be seen in the second column of Table 6. Since the start and target addresses, and the supported transport modes are selected randomly (see Section 4.1), for some queries there are no connections for the requested time interval. Therefore, we also skipped queries for which we could not find any connections. The third column contains the number of evaluated queries.

# Transport Modes in Query	1	2	3	8 (r)	8 (a)
Routing Time (ms)	4,492	5,902	6,599	6,823	7,579
# Street Routing Requests	2,045	3,189	4,274	7,516	9,772
# Start Slots	179	461	734	1,485	1,617
# Destination Slots	340	325	324	329	329

Table 5: The influence of the number of supported transport modes.

Approach	Threshold for Num. of Stations	Num. of Queries	Runtime in s (Avg)	Runtime in s (0.9 Quantile)
Single-Source Single-Target	$\lvert M\rvert + \lvert N\rvert \leq 100$	89	1,810.970	6,017.232
Multi-Source Single-Target	$\lvert N\rvert \leq 100$	317	990.382	2,645.779
Single-Source Multi-Target	$\lvert M\rvert \leq 100$	89	172.519	537.678
Multi-Source Multi-Target	no threshold	1,112	5.578	10.628

Table 6: The improvement obtained by multi-source multi-target search.

The last two columns show the runtime improvement achieved by multi-source multi-target search. While the single-source single-target approach needs about 30 minutes in average, a multi-source multi-target search is averagely performed in 5.578 seconds. The obtained speed-up factor is 324.66.

5 Conclusion and Outlook

We have shown that a complete set of reasonable Pareto optima for the relevant optimization criteria can be computed even in a highly complex, fully realistic scenario with an acceptable query response time (w.r.t. our backbone algorithm).

We expect that we will get real-world data for car sharing and bike sharing in the future. Integrating these modes of transportation and keeping the query response times acceptable will be another serious challenge.

References

[1] Leonid Antsfeld and Toby Walsh. Finding multi-criteria optimal paths in multimodal public transportation networks using the transit algorithm. In *Intelligent Transport Systems World Congress*, page 25, Vienna, October 2012.

[2] Hannah Bast, Mirko Brodesser, and Sabine Storandt. Result diversity for multimodal route planning. In Daniele Frigioni and Sebastian Stiller, editors, *ATMOS*, volume 33 of *OASICS*, pages 123–136. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[3] Maurizio Bielli, Azedine Boulmakoul, and Hicham Mouncif. Object modeling and path computation for multimodal travel systems. *European Journal of Operational Research*, 175(3):1705–1730, 2006.

[4] Daniel Delling, Julian Dibbelt, Thomas Pajor, Dorothea Wagner, and Renato F. Werneck. Computing and Evaluating Multimodal Journeys. Technical Report 2012-20, Faculty of Informatics, Karlsruhe Institut of Technology, 2012.

[5] Daniel Delling, Julian Dibbelt, Thomas Pajor, Dorothea Wagner, and Renato F. Werneck. Computing multimodal journeys in practice. In *SEA*, pages 260–271, 2013.

[6] Daniel Delling, Thomas Pajor, and Dorothea Wagner. Accelerating multi-modal route planning by access-nodes. In Amos Fiat and Peter Sanders, editors, *ESA*, volume 5757 of *Lecture Notes in Computer Science*, pages 587–598. Springer, 2009.

[7] Julian Dibbelt, Thomas Pajor, and Dorothea Wagner. User-constrained multimodal route planning. In David A. Bader and Petra Mutzel, editors, *ALENEX*, pages 118–129. SIAM / Omnipress, 2012.

[8] Yann Disser, Matthias Mller-Hannemann, and Mathias Schnee. Multi-criteria shortest paths in time-dependent train networks. In Catherine C. McGeoch, editor,

WEA, volume 5038 of *Lecture Notes in Computer Science*, pages 347–361. Springer, 2008.

[9] Andrew Ensor and Felipe Lillo. Partial order approach to compute shortest paths in multimodal networks. *CoRR*, abs/1112.3366, 2011.

[10] Dominik Kirchler, Leo Liberti, Thomas Pajor, and Roberto Wolfler Calvo. UniALT for regular language contrained shortest paths on a multi-modal transportation network. In Alberto Caprara and Spyros Kontogiannis, editors, *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, pages 64–75, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[11] 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.

[12] Paola Modesti and Anna Sciomachen. A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. *European Journal of Operational Research*, 111(3):495–508, 1998.

[13] Matthias Müller-Hannemann and Mathias Schnee. Finding all attractive train connections by multi-criteria pareto search. In *ATMOS*, pages 246–263, 2004.

[14] OpenStreetMap http://www.openstreetmap.de/.

[15] Haicong Yu and Feng Lu. Advanced multi-modal routing approach for pedestrians. In *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, pages 2349 – 2352. IEEE, 2012.