

A Heuristic for Multi-modal Route Planning

Dominik Bucher, David Jonietz and Martin Raubal

Abstract Current popular multi-modal routing systems often do not move beyond combining regularly scheduled public transportation with walking, cycling or car driving. Seldom included are other travel options such as carpooling, carsharing, or bikesharing, as well as the possibility to compute personalized results tailored to the specific needs and preferences of the individual user. Partially, this is due to the fact that the inclusion of various modes of transportation and user requirements quickly leads to complex, semantically enriched graph structures, which to a certain degree impede downstream procedures such as dynamic graph updates or route queries. In this paper, we aim to reduce the computational effort and specification complexity of personalized multi-modal routing by use of a preceding heuristic, which, based on information stored in a user profile, derives a set of feasible candidate travel options, which can then be evaluated by a traditional routing algorithm. We demonstrate the applicability of the proposed system with two practical examples.

Keywords Multi-modal routing · Routing algorithms · Personalization

1 Introduction

Whereas the computation of an optimal driving path through a static road network is a well-researched topic, there has been far less work on the still challenging tasks of integrating personalization options and multiple modes of transportation (Sanders and Schultes 2007; Bast et al. 2015). Particular problems result from the fact that

D. Bucher (✉) · D. Jonietz (✉) · M. Raubal (✉)

Institute of Cartography and Geoinformation, ETH Zurich, Stefano-Franscini-Platz 5,
8093 Zurich, Switzerland
e-mail: dobucher@ethz.ch

D. Jonietz
e-mail: jonietzd@ethz.ch

M. Raubal
e-mail: mraubal@ethz.ch

different means of transport operate on specific sub-networks, e.g. a walking path versus a road or train network, which have to be interconnected via distinct transfer nodes which are partly time-dependent, for instance in the case of public transportation or car-pooling (Delling et al. 2014; Raubal et al. 2007). Apart from an increasing complexity of the subsequent graph, which can lead to higher computational effort required for routing queries, a typical result are drastic inequalities with regards to node degrees, which in turn reduce the effectivity of several algorithms developed to speed up graph network computations (Bast 2009). In addition, the dynamic nature of modalities such as car-pooling or public transport imply further growth of the graph and the attributes of its elements as well as call for more complex updating and routing algorithms (cf. Bast et al. 2015 for a list of algorithms).

Personalization of a routing service, i.e. tailoring the results to the specific requirements of a particular user, is especially relevant in the context of multi-modal routing (e.g. Arentze 2013; Liu et al. 2014; Funke and Storandt 2015), but further complicates graph operations. The recommended route is typically individualized by means of a set of user-specific numerical values, which serve as parameters of link cost functions. In our view, this method has the following drawbacks:

- Semantically richer user profiles lead to more complex (and computationally intensive) edge cost functions.
- While user preferences can be easily and intuitively expressed by means of weight coefficients, they are not well-suited for representing hard restrictions (e.g., by means of setting the weight coefficients to ∞).
- Although due to hard restrictions, some travel options might not be feasible for a specific user, they are normally still checked by the routing algorithm.
- The system cannot easily be adapted to new requirements, e.g. additional user characteristics.

With regards to these issues, in this paper we propose an alternative approach for personalized multi-modal route computation, namely to use a rule-based heuristic which precedes the actual routing algorithm, and which, based on a user profile and the situational context, identifies a preliminary set of feasible candidate travel options. In a second step, rather than analyzing the entire graph for the optimal multi-modal route, a routing algorithm focuses merely on the set of pre-computed alternatives, thereby computing detailed routes and assessing their total cost. This procedure, we argue, can dramatically reduce the computational load of the route calculation process, allows for the incorporation of semantically rich user profiles with preferences and restrictions, and can be easily extended. Finally, we demonstrate the applicability of our method with a practical scenario in an urban area, and discuss the resulting routes.

This paper is structured as follows: In Sect. 2, we present relevant prior work related to multi-modal routing and the personalization of route recommendation systems. Our method is described in Sect. 3, starting with a formal description of the rule-based heuristic and moving on towards the general system architecture and details of the implementation. In Sect. 4, we show a practical example before the

results of this study are presented in Sect. 5. The final Sect. 6 provides a discussion of the results and concludes this paper.

2 Related Work

This section presents prior work of relevance in the areas of route planning and personalization, in particular with regards to techniques which address the challenges of multi-modal route planning discussed previously.

2.1 *Techniques for Multi-modal Route Planning*

Routing (i.e., finding the shortest path between nodes) on graphs is a well-researched area (cf. Bast et al. 2015). For that purpose, the famous algorithm of Dijkstra (1959) can be used on a broad range of graphs (directed and undirected, as well as cyclic and acyclic), but is only suitable for graphs with positive edge weights. In case of negative edge weights, the algorithm of Bellman-Ford (1956) is widely used as an alternative.

An application of such algorithms to transportation route planning, however, often requires specific means to speed up processing times, which is due to domain-specific aspects such as large, dynamic road networks and complex calculation criteria. For this reason, various algorithms have been developed. The A* algorithm, for instance, uses a simple heuristic to direct the search towards the target, e.g. by incorporating the Euclidean distance to the destination as an additional factor (Hart et al. 1968). Although this technique has proven to be very useful for routing through road networks, it still requires large parts of the graph to be traversed. Faster speedup techniques include for example Contraction Hierarchies (Geisberger et al. 2008) and related algorithms, which, however, require extensive preprocessing of the graph, during which shortcuts in the graph are inserted whenever a connection between two nodes is suitable. A graph which has been reduced in this manner allows quick routing, and can be unfolded after a route has been identified. Contraction Hierarchies work best for balanced graphs with evenly distributed nodes and vertices (such as a street network). When modeling a typically rather unbalanced public transport network, however, Contraction Hierarchies have to be applied with care (Bast 2009). A similar, but even faster method is proposed with the Transit Node algorithm, which, however, requires more intensive preprocessing and auxiliary space (Bast et al. 2007).

In general, the choice of a suitable routing algorithm requires a trade-off between query processing time, preprocessing time, and graph size, ranging from no preprocessing at all (using Dijkstra or similar), to exhaustive pre-calculation of the optimal route between all vertex pairs, thereby reducing route queries to a simple table lookup.

Naturally, these techniques can be used for multi-modal routing. However, since the underlying graph can change its structure dynamically, query times can increase dramatically, depending on the deployed unfolding technique (Bast 2009). While graph unfolding is successfully applied to a variety of modal combinations (Bast et al. 2015; Dibbelt et al. 2015; Sanders and Schultes 2007), there are also other approaches: For example, RAPTOR (Round-Based Public Transport Routing) does not use a graph at all, but instead relies on data structures that hold individual public transport trips with stops, departure times, and transfer possibilities (Delling et al. 2014). In each iteration, newly reached stops are computed without a change of the transport vehicle. In the next round, the means of transport is changed in order to reach new stops. This procedure continues until a route has been identified. To the best of our knowledge, the (hypothetical) expansion of RAPTOR to multi-modal routing comes closest to our approach, however, without any means of personalization or acknowledgement of the different requirements posed by the various means of transport.

Another noteworthy and comparable approach combines spatial analysis and network routing to reach the desired result (Eiter et al. 2012): In a first step, possible intermediate stops are evaluated based on a spatial join, a standard analysis method to be found in many geographic information systems (GIS). These are then connected, using “traditional” routing algorithms. In contrast to our work, Eiter et al. (2012) do not use this technique for the routing itself, but rather to find places that adhere to some user preferences or choices. Finally, they only consider a single intermediate stop.

2.2 *Personalization and User Profiling*

The concepts of personalization and user profiling are very closely related. Thus, a user profile, according to the European Telecommunications Standards Institute (ETSI 2005, p. 17), “the total set of user-related information, preferences, rules and settings, which affects the way in which a user experiences terminals, devices and services”, represents a prerequisite for successfully personalizing systems or applications (Foerster et al. 2012; Aoidh et al. 2009). Personalization, in turn, describes the tailoring and adjustment of such services to the specific needs and preferences of a particular user, and might serve to filter irrelevant content, alter data visualization methods, make suitable recommendations or adapt the functionality or visual appearance of the program itself (Aoidh et al. 2009; Aissi and Gouider 2012; Raubal 2009).

User profiling and personalization are not only of relevance for routing applications, but for a wide range of Information and Communications Technology (ICT) applications, including Location Based Services (LBS) (Foerster et al. 2012; Raubal 2011). In a routing context, apart from calculating the shortest path, as has traditionally been a key functionality of GPS-enabled navigation devices, alternative approaches have developed algorithms to compute the fastest, the least

complicated in terms of wayfinding, the most attractive, the safest, or the easiest route (Golledge 1995; Huang et al. 2014; Krisp and Kehler 2015). A more substantial personalization, however, is achieved when individual preferences and constraints are incorporated into the routing algorithm, which are usually represented as parameters of link costs, stored as a vector of individual weight coefficients (e.g. Arentze 2013; Liu et al. 2014; Funke and Storandt 2015; Nuzzolo et al. 2015; Yang et al. 2015; Campigotto et al. 2016).

The user profile, therefore, occupies a central role in the process of computing personalized routes. It is usually represented as a set of numerical values, which serve as parameters for the route or edge cost calculation. Arentze (2013), for instance, acknowledges individual preferences with regards to travel-time (mode-dependent), financial travel costs (situation-dependent, e.g. fuel costs are generally valued less than ticket costs), preferred traffic mode and service quality (e.g. seat availability and avoiding transfers). In their work on a personalized routing system for tourists, Liu et al. (2014) represent preferences with regards to the degree of interest in specific points of interest (POI), maximum distance, fee, minimum road conditions and traffic conditions. The user profile derived by Nuzzolo et al. (2015) relates to differences between actual and desired arrival times, waiting times, sum of access and egress times, on-board time spent, on-board crowding degree, preference for transit service, number of transits. Campigotto et al. (2016) distinguish between features of routes related to the distances covered and time spent with regards to each individual mode of transportation, financial cost, and others, including the number of transportation means changes, the amount of uphill cycling, as well as the number of public transport stops.

In order to create a user profile, information related to aspects such as personal needs, preferences, interests or habits must be collected. This can occur either by explicitly requesting a user's feedback, or implicitly by inferring such unobservable information from observable behavior (Aoidh et al. 2009) or a combination of both (e.g. Allemann and Raubal 2015). In the context of routing applications, users are often explicitly asked for their general preferences prior to the route recommendation, however, in some cases their actual choice behavior is also monitored and taken into account (e.g. Nuzzolo et al. 2015; Campigotto et al. 2016). There are also studies which use GPS-trajectories from previous travels to extract user preferences and restrictions, however mostly focusing on unimodal transportation (e.g. Letchner et al. 2006; Dai et al. 2015; Yang et al. 2015; Jonietz 2016; Broach et al. 2012; Tsui and Shalaby 2006).

3 The Pre-processing Heuristic

In this section, we propose a heuristic as part of a larger multi-modal personalized routing system. The heuristic allows to pre-calculate a set of plausible routes (where each route is defined by a list of route segments, consisting of start and end point, and a transport modality), which can then be evaluated in more detail by a

conventional route planner, in order to check for actual feasibility and user acceptance. In contrast to conventional route planning, we argue, our heuristic is easily adaptable to user preferences (e.g. with regards to modes of transport) or routing constraints (e.g., critical threshold values, such as maximal walking distances, or excluded modalities), and, other than algorithms based on extensive graph pre-processing, does not require algorithm-dependent updates of the transport graph.

In fact, the proposed travel options could already be useful for presenting a user the different possibilities to reach the destination on a less detailed level, for instance by taking a particular train line in combination with a tram, or going by bus to a car-sharing station. Nevertheless, on this general level, our heuristic does not consider the actual road network or public transport timetables. Thus, in a second step, the individual segments of a route should be analyzed and validated by use of a conventional route planner, for instance based on an existing solution such as the Google Maps Direction API¹ or ESRI routing services,² and run independently of the heuristic itself.

Exemplary use cases for application of our heuristic are the following:

- A user is interested in evaluating different options to get from Zurich to Munich, without any particular constraints.
- A user has a bike at home, and is interested in getting to work in an energy-efficient manner (i.e., emitting the least amount of CO₂).
- A backpacker tourist would like to use car-pooling or any other cheap means of transport.

Naturally, such use cases could also be addressed via traditional routing systems, however, such queries would necessarily involve multiple web services and platforms, and require the user to manually assess the feasibility of the derived route options. This is partially due to the fast increasing complexity of multi-modal route requests, which, by use of our heuristic, can be decreased by the possibility to formulate pre-set queries, build a dynamic transport graph, and prune large parts of this graph before evaluating it in detail.

3.1 Formalizing Route Calculations

Formally, a route request R can be defined as a tuple consisting of an origin O , represented by a coordinate pair (O_{lat}, O_{lon}) and a set of labels O_L , which denote selected characteristics of the location, such as the number of bikes at the location or the train lines that stop at the location, and a destination D of the same structure as O . More complex requests (which we cover in this paper) also contain a set of

¹<https://developers.google.com/maps/documentation/directions>.

²<https://developers.arcgis.com/features/directions>.

user preferences P , a set of user constraints A and a description of the context C , i.e., the current state of the world, such as traffic conditions or the weather. Both preferences and constraints can have effect on a variety of route characteristics, such as mode choice, number of inter-modal transfers, time or financial cost, walking distances, or energy consumption.

The expected result of a route request is a list of routes L_R , ordered by some criteria, for instance a combination of various route characteristics, such as the number of mode transfers and energy consumption. A single route contains all intermediate trip segments S_i as well as summary values V_i , such as distance covered with the different means of transport, overall and detailed energy consumption, or the incurring monetary cost of the route. Summarizing, the function of calculating routing results can be denoted as follows:

$$\{(S_{0,i}, V_0), (S_{1,j}, V_1), \dots\} = f_{routing}(O, D, P, A, C)$$

The aim of our heuristic is not to produce an exact description of the route (i.e., a detailed path on the transport network), but instead a list of trip segments denoted by a start and end point, and an associated mode of transport. For instance, the returned result could tell a user to take the bike from home to a particular train station, then a train line to another station, a tram to a stop close to the destination, and walk the final part of the journey. At this stage, however, it is important to note that the results are approximate, and not yet verified by a route planner. Thus, in order to ensure that the proposed trip is actually feasible, a traditional route planner needs to be deployed, which takes the exact transport network into account.

3.2 Rule Base for Different Modes of Transport

The proposed heuristic evaluates the feasibility of different trip segments and used modes of transport based on their individual preconditions in combination with user preferences and constraints. Thus, for example, driving is only possible if a car is available at the current location of the user; a train ride between two stations depends on the existence of at least one connecting line.

Since all modes of transport are marked by an individual set of preconditions and outcomes, we formulate individual rules of the general form:

$$O[condition] \rightarrow M[condition] \rightarrow D[condition]: [outcomes]$$

where both the origin O and the destination D locations have to fulfill some conditions in order for the mode M to be potentially available for a user. Examples for location conditions would be the availability of a car or the labelling as a train station. The mode condition, in contrast, concerns the action of traveling per se. Walking between locations, for instance, is only possible if the total distance is below the maximum acceptable walking distance for a particular user. If a user

moves between two locations, the resulting outcomes mostly affect herself (e.g. she will have walked a certain distance if the mode is “WALK”), but can also influence the context (time will inevitably pass, and the weather might change). Naturally, a further outcome is that the user changed its position to the destination location.

To state an example, consider the following rule, simplified for better reading (for a comprehensive list of all rules, we refer to Sect. 4):

$$\begin{aligned}
 & A[\emptyset] \rightarrow \\
 & \text{WALK}[(\text{user}[\text{distWalked}] + \text{dist}(A, B) < \text{user}[\text{maxDist}]) \\
 & \quad \text{AND } (\text{NOT } \text{context}[\text{rainyWeather}])] \\
 & \text{AND } (\text{context}[\text{currentTime}] \text{ IN } \text{user}[\text{acceptableTimeIntervalWalk}])] \\
 & \rightarrow B[\emptyset]: \\
 & \text{user}[\text{distWalked} += \text{dist}(A, B)], \text{context}[\text{time} += \text{time}(A, B)]
 \end{aligned}$$

The rule describes the locational and mode preconditions as well as outcomes for walking between an origin and a destination. A lack of locational preconditions denotes that in principle, it is always possible to walk between locations. With regards to the mode conditions of walk, however, one can see that there are three preconditions which need to be fulfilled for the journey to be possible: Firstly, the maximal distance covered by walking must not exceed a certain threshold, namely the maximum acceptable walking distance for a particular user. It would, of course, also be thinkable to incorporate other aspects here, such as the steepness of terrain, which, however, is omitted at the present stage of our work. The second and third preconditions are related to the context, thus walking is only possible if the weather is not rainy and the current time is within an acceptable time interval for walking set by the user (e.g., no walking at night time due to fear of crime). With regards to the outcome of walking from A to B , the walked distance needs to be added to the total walked distance of a particular user. Further, the passed time has to be noted. As it has been stated earlier, for these calculations, the heuristic does not consider the exact street network distance between A and B , but rather refers to the Euclidean distance. This approximation eliminates the need to have an underlying transport graph, and makes the system very flexible with regards to inserting and updating locations. A similar approach has been used in transport planning, e.g., by Raubal et al. (2007).

3.3 Procedure

Based on rules such as the one discussed in the previous section, an ever-increasing graph of modal change nodes can be computed via several functions. The first function, called *checkReachability*, computes the modes of transport that can potentially be used to reach a set of destination locations from a set of origin locations, if any are available. This is done by iterating through the set of origin locations, and, for each mode of transport, determining whether all origin

conditions are fulfilled. The destination locations are analyzed in a similar manner. Afterwards, the mode conditions are tested for valid origin and destination pairs. The rule for WALK, for instance, allows all locations as valid origins and destinations, and then checks for every combination whether the destination is reachable with regards to the walking distance of the origin, the weather conditions, and the time of day.

Given an origin (or destination) location, and a mode of transport, the second function, called *expand*, is used to produce a new set of locations. For example, applying the function to a location which is connected to the public transport network, and setting public transport as mode, would yield all other locations connected by the same lines (respecting one-way properties of individual lines), i.e., those which are reachable without a transfer.

Based on these two functions, the following procedure can produce potential routes:

1. An initial check tests whether the destination can be reached directly (i.e., without any transfers) from the origin location. This is done by calling the *checkReachability* function with origin O and destination D . If potential routes are detected, for each mode M that allows a direct transition between origin and destination, a triple (O, D, M) is added to the set of solutions. In most cases, this will include taking the car (if a user does not explicitly discourage this solution, or does not own a car), as most locations can be reached by driving.
2. The function *expand* is applied to the origin O , and, based on all modes M of transport allowed by the user, calculates a set of reachable locations $\{L_1, L_2, \dots\}$. This implies a user to travel to L_i by using the respective mode. For this reason, the quadruple (O, L_i, M, S) is stored with the location L_i , which will, in a later step, allow the algorithm to backtrack and build an optimal route. The state S contains more information on the route to reach L_i , such as the number of mode transfers, or the distance walked. This information is necessary since a location L_i can usually be reached by more than one mode.
3. For each member of this new set of starting nodes $\{L_i\}$, we check if the destination D can be reached, using *checkReachability*. If true, the triples (L_i, D, M) are added to the set of preliminary results. These need to be backtracked in a later step in order to calculate the optimal route.
4. Starting from the destination location, *expand* is used to identify a set of new reachable locations L_i and their corresponding modes. This step is similar to 2, but takes place in reversed order. The found quadruples (L_i, D, M, S) are again added to the respective location L_i , to allow backtracking later on.
5. *checkReachability* is applied to this new set of nodes in order to determine possible starting points from within the previously defined set of reached locations when having started from the origin. Similar to 3, found triples (L_i, L_j, M) are added to the set of preliminary solutions. The algorithm then continues at point 2 until a maximum number of transfers are reached, or another stopping criteria has been met (e.g., a valid route has been found, or a route which is substantially better than simply using the car).

6. The preliminary set of solutions is used to calculate a final set of solutions, which includes all trip segments. For this, a third function *unfold* is needed. Here, the stored quadruples are used to backtrack and find a way to the origin, respective to the destination. At the current stage, the tuple with the least number of transfers is chosen, since this is generally assumed to be a useful proxy for route quality (users are often interested in transferring as little as possible, because transfers always lead to slack times). However, more elaborate backtracking functions are imaginable, which take into account the energy efficiency or the travelled distance.

The presented approach dynamically constructs a mode transfer graph which is personalized with regards to user preferences and constraints. Technically, steps 4 and 5 are not required, but instead one could continue expanding the graph from one side. However, comparable to bidirectional Dijkstra search strategies (cf. Bast 2009), this leads to a smaller set of locations needed to be checked, which in turn reduces computational complexity. Also, due to the fact that travel routes usually follow a hierarchical scheme (for instance one needs to walk to a higher-order hub of a means of mass-transit, use this mode, and finally cover the “last-mile” by walking again), bidirectional search simplifies the incorporation of strategies which favor such hierarchical route calculation. For instance, long segments associated with “WALK” could only be allowed at origin and destination.

The constructed graph is drastically reduced in comparison to the full transportation network graph, which is due to the *expand* function, which only considers locations that can be reached via the allowed modes of transport, and by considering user preferences, the context, and the state stored in the quadruple (O, D, M, S) . For example, if a user has reached her maximum walking distance, further nodes that would involve walking are not considered anymore. The same is true if the weather conditions, as stored in the context, are rainy.

4 Implementation

In this study, the proposed method was implemented as a preprocessing step for a routing system aimed at supporting sustainable ways of traveling. This is embedded in the larger context of *GoEco!*, a project which aims at developing a mobile app designed to provide feedback to users with regards to energy-efficient ways to alter their mobility behavior (Cellina et al. 2016; Bucher et al. 2016). This includes the computation of multi-modal, personalized routes which favor sustainable mobility options such as walking, biking or public transport.

The described concepts were implemented in Python, using a PostGIS database for storing and querying spatial data. The database contains all transfer locations, with respective labels denoting properties of the location. Querying this database

Table 1 A selection of rules implemented in our prototype system

Mode	Rule	Description
Walk	$d_w = user[distWalked] + dist(A, B)$ $A[\emptyset] \rightarrow WALK(d_w < user$ $[maxDist])$ and (NOT context $[rainyWeather]$ OR ($d_w < user$ $[maxDistRain])$) and (context $[currentTime]$ IN user $[acceptableTimeIntervalWalk])$ $\rightarrow B[\emptyset]$: $user[distWalked += dist(A, B)]$, $context[time += time(A, B)]$	Every node provides walking, however, a user can only walk up to a maximal distance (which gets decreased if it is raining), and if the current time is within an accepted time interval for walking. As a result of walking, the total walked distance is updated as well as the context
Bike	$A[user[bikeLocation] = A] \rightarrow BIKE$ $[(NOT context[rainyWeather])$ AND $(context[currentTime]$ IN user $[acceptableTimeIntervalWalk])$ $\rightarrow B$ $[bikeParking = true]: user$ $[bikeLocation] = B, user[distBiked]$ $+= dist(A, B), context[time += time$ $(A, B)]$	A user can only take the bike, if her bike currently is at the location. Further, the destination needs to have a bike parking spot available. Concerning contextual variables similar to walking
Car	$A[user[carLocation] = A] \rightarrow CAR$ $[\emptyset] \rightarrow B[\#parkingSpots > 0]: user$ $[carLocation] = B, context[time$ $+= time(A, B)]$	Taking the car is only possible from the location where the user currently has parked her car to locations with a parking spot available. As a result, the car is at location B
Train	$A[connectsLineX = true] \rightarrow TRAIN$ $[\emptyset] \rightarrow B[connectsLineX = true]:$ $[\emptyset], context[time += time(A, B)]$	Similar to bus
Tram	$A[connectsLineX = true] \rightarrow TRAM$ $[\emptyset] \rightarrow B[connectsLineX = true]:$ $[\emptyset], context[time += time(A, B)]$	Similar to bus
Carshare	$A[carSharing = true,$ $\#cars > 0] \rightarrow CARSHARE[\emptyset] \rightarrow B$ $[\#parkingSpots > 0]: A[\#cars$ $= 1], context[time += time(A, B)]$	Carsharing is possible from carsharing locations, where enough cars are available. The destination needs to have free parking spots
Carpool	$A[intersects(A,$ $C) = true] \rightarrow CARPOOL[\emptyset] \rightarrow A$ $[intersects(B, D) = true]: [\emptyset]$	Carpooling is possible from locations that intersect with a spatio-temporal corridor of a carpooler
Bike-Share	$A[bikeSharing = true,$ $\#bikes > 0] \rightarrow BIKESHARE$ $[context[weather] \neq "rain"] \rightarrow B$ $[bikeParking = true]: A[\#bikes$ $= 1], user[distBiked] += dist(A,$ $B), context[time += time(A, B)]$	Bikesharing is possible from bikesharing locations, where enough bikes are available

for appropriate locations, our heuristic dynamically builds up the transport graph. Table 1 shows the different modes of transport supported by our prototype system, and the according rules currently implemented. For practical reasons, this list does

not include the entirety of the updated variables (i.e., the outcomes of the actions). Thus, for example, the distances traveled with the respective mode are always updated (which also makes it easy to introduce rules depending on those distances, e.g., a maximal car distance), as well as the number of transfers taken.

The modes car and carpool represent special cases, as the simple application of the above heuristic would quickly lead to a very large set of nodes due to the fact that by car, all locations are reachable. While this makes sense for trips of the pattern WALK → CAR → WALK, it seems an edge case to assume that a user would for instance drive the major part of the total distance to the destination, only to switch to carpooling for the last part of the trip. To reduce the search space, we constrain the maximal distance traveled by car in the *expand* function to half the total distance between the car location and destination, even if the user specified a larger maximal car distance (note that the patterns WALK → CAR → WALK, or even WALK → CAR → TRAIN → WALK, as commonly found in Park and Ride systems, are not excluded from the solutions here, as they are still identified during *checkReachability*). In general, carpooling, meaning shared journeys in one privately owned car, is implemented as follows: assuming that a driver is willing to drive short detours to pick up other people, and drop them off somewhere along her original route, we model a detour corridor by buffering the original route. Any location within this buffer is a possible pick-up spot, whereas any location downstream becomes a valid drop-off location.

In our implementation, upon calculation of a set of viable routes, a weighting and ranking by approximate energy consumption is applied (using a distance based energy consumption model³). Of course, for other applications, a weighting by distance walked, number of transfers taken, or a combination thereof could be more suitable. Finally, from these sorted route alternatives, a predefined number of the top distinct ones are selected and presented to the user, where distinct means they at least differ in the mode of one segment. This distinction is necessary, because it is often possible to change the transport vehicle at different locations, even when using the same sequence of transport means (e.g., changing from tram number X to tram number Y might be possible at different stops, because they share a part of their routes).

In our approach, the search space is therefore reduced as follows: First, the heuristic only has to consider a graph consisting of transfer nodes, which is mostly made up from public transport stops (a reduction of two to three orders of magnitude). Second, for validation of individual route segments, only the respective sub-graphs have to be queried (e.g., the street network, or the public transport network). Because the linking of these sub-graphs causes many speedup techniques to fail, it is beneficial to query them separately.

³We use the energy consumption values for different modes provided by <http://www.mobitool.ch>.

5 Case Studies

In this section, we describe an exemplary application of the proposed heuristic to two case studies set within the city of Zurich. To illustrate the potential of our method for personalization, for each scenario, two users with differing preferences and constraints are defined:

1. Users A and B would both like to travel from Seebach (a suburb in the northeast of Zurich) to a place close to the city center at daytime. The weather is fine at the time the users leave but heavy rain is forecasted for a later point in time (15 min after leaving). User A has a car, but no bike available and would like to walk no more than 1 km in total. User B has no car and no bike but is willing to walk the longer distance of 3 km. For both, the maximum walking distance decreases by a factor 5 during rain. If multiple alternatives are available, they would like to choose the ones with the smallest amount of CO₂ emissions at the expense of longer walking or biking distances.
2. Users C and D would both like to travel from Altstetten, located on the periphery of Zurich, to the city center during late evening hours, and fine weather conditions. User C only has a bike (no car), but is afraid of walking or biking at nighttime (so C tries to keep the respective distances small). User D also has a bike, and has no problem with walking or biking at night. User D likes to do sports, tries to save energy, and does not mind biking for longer distances.

We stored all public transport stations together with the served lines (in the vicinity of Zurich, taken from the GTFS⁴ feed from the Swiss federal railways⁵), all carsharing stations (obtained from <http://www.mobility.ch>) and bikesharing stations (obtained from the city of Zurich) in a PostGIS database, and ran all queries on an Intel Core i7-4910MQ CPU (2.9 GHz), on a laptop with 16 GB of RAM. Query responses of our Python implementation were in a range of approximately 100 ms–3 s, depending on the number of routes found, and the number of modal changes involved. After the heuristic had been completed, the routes were refined using a routing system comparable to Google Maps Directions. Individual route segments (start and end location, as well as modality) were fed into the system, and the returned routes drawn on an OpenStreetMap base map, using the Leaflet library⁶.

For case study one, we additionally assume that user A has parked her car approximately 400 m down the road. The heuristic is able to pick up arbitrary car and bike locations, similar to carsharing, or bikesharing stops. Figure 1 shows the routing result for user A. While the left part shows the raw output of our preprocessing heuristic, we refined the route by querying a conventional route planner for better visualization, as described previously. The route planner computes a path for

⁴General Transit Feed Specification, see <https://developers.google.com/transit/gtfs>.

⁵<http://www.fahrplanfelder.ch>, retrieved via <http://gtfs.geops.ch>.

⁶<http://www.openstreetmap.org>, <http://leafletjs.com>.

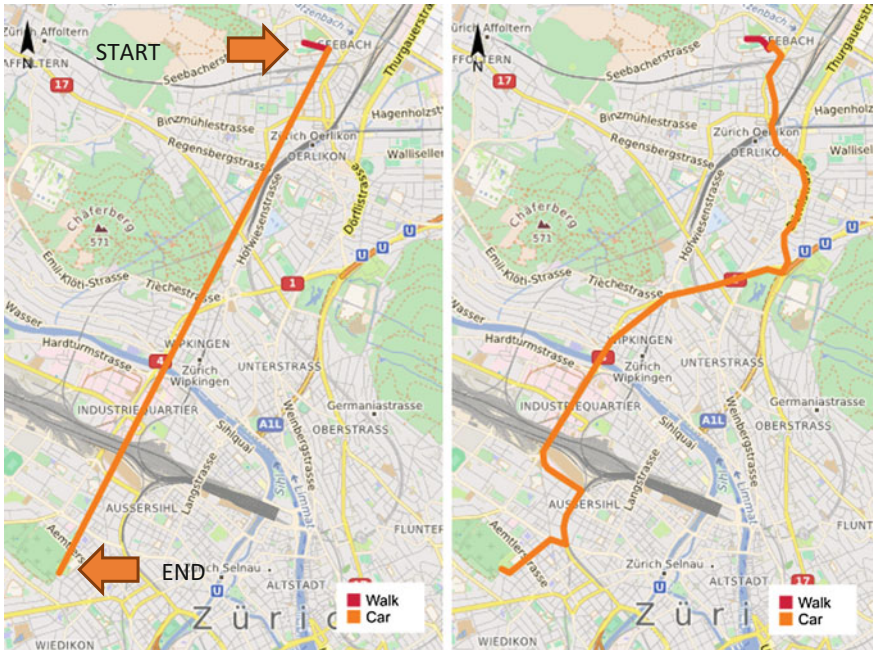


Fig. 1 User A of case study one. The *left* figure shows the raw output of the heuristic, the *right side* the exact route. The *red* WALK segment in the beginning shows the distance walked to the car

each segment, using the mode of transport proposed by the heuristic. The output is shown in the right part of the figure.

It can be seen that the heuristic identifies only the car as a viable travel option, primarily because of the user’s reluctance to walk for longer distances. Even though public transport stops can be found approximately 450 m from the start location, the maximum walking distance of only 1,000 m, which is further reduced due to the forecasted rainy weather conditions (max. walking distance during rain 200 m), conflicts with the fact that the closest public transport stop to the destination is around 300 m away, which ultimately leads to the sole preference of the car.

Figure 2 shows the result for user B (max. walking distance of 3,000 m, which during heavy rain is reduced to 600 m). Whereas the left figure shows the results under consideration of the forecasted rainy weather, this is neglected on the opposite map. It is clearly visible that the upcoming rain limits the options of user B, primarily because it leads to a reduced number of modes considered, namely solely the ones which do not require a long final walk to the destination.

It is noteworthy, however, that the right figure does not depict the routes with the smallest walking distances. This is due to the fact that for the sake of energy efficiency, user B is willing to walk or bike longer distances (as long as it is not raining). In comparison to user A, user B has a completely different set of options, resulting from the lack of a car, and the willingness to walk further. A final note

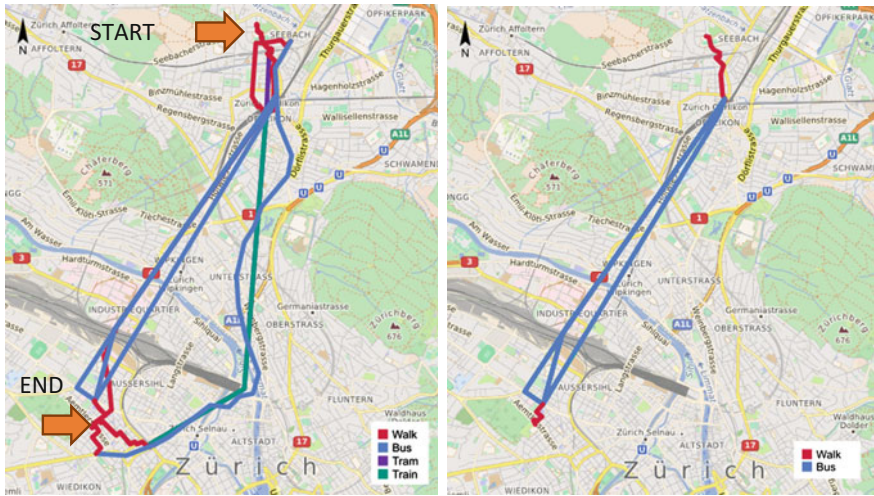


Fig. 2 User B of case study one. The *left* figure shows possible routes without consideration of weather restrictions. The *right* figure assumes good weather in the beginning, but high chances of rain at a later point

concerns the different route alternatives: In the left figure, it is easy to identify different ways of reaching the destination. User B could for example cover most of the route with the train, or use a combination of train, tram, or bus to reach a variety of stops close to the destination. Of course, it is not guaranteed that all of these routes are still feasible once time schedules are considered. However, such output could already assist a user in planning a route.

Figure 3 compares the results of users C and D. User C’s options are very limited: Since C would like to minimize walking during the night (the maximal walking distance during the night was set to 200 m), the only viable travel option computed by the heuristic includes renting a car from a nearby carsharing station.



Fig. 3 User C (*left side*) and user D (*right side*) of case study two. While C is restricted from walking too far during the night by fear, D has many more options

Using this car, user C can drive to the city, park it, and walk the last meters to the destination. User D, on the other hand, has plenty of options: Apart from using the same carsharing station, D could directly ride the bike to the city center (blue line), or take a variety of combinations of public transport (green lines). Note that the order of the returned alternatives for user D is “*bike, public transport (route 1–3), carsharing*”, as D would like to use an energy efficient route (where bike is rated as the most energy efficient, and carsharing as the least efficient alternative).

The presented case studies demonstrate the applicability of the heuristic to a variety of personalized use cases, resulting in various multi-modal routes to be proposed. Depending on the preferences of users and the available transport modes, a completely different set of route alternatives are suggested. While the routes which have actually been computed by a conventional route planner resemble solutions known from various routing systems available to the public, the heuristic is able to drastically reduce the search space prior to detailed computations on the transport graph, and is able to suggest less conventional alternatives, such as using various shared means of transport, or routes depending on the weather and other context variables.

6 Conclusion

In this paper we proposed the use of a preset rule-based heuristic for the computation of personalized multi-modal routes through complex, dynamic transportation networks. We discussed how the user profile and the situation-specific context can be deployed in order to reduce the search space of the route planner to a subset of pre-defined feasible origin-destination pairs and the according modes of transport. The functionality and practical value of our method was demonstrated on the example of different users traveling in the city of Zurich.

The use of a preceding heuristic has several advantages, including a reduction of computational load needed for the route planning process. By providing the route planner with a set of pre-defined start- and end-nodes, as well as the mode-specific part of the transportation network to be used for this trip segment, the search options necessary to be checked are minimized. Further, this strategy allows for parallelizing the route planning process. Thus, while the heuristic is still being executed, the actual route planner can already evaluate the resulting route options. Also, individual trip segments might be part of several different trip options, but only be evaluated once (for example, a journey by bike from home to a train station needs to be evaluated only once, even though the overall journey might continue differently). Additionally, the procedure allows for more detailed and dynamic personalization due to the fact that different user profiles merely require the set of rules in the heuristic to be updated, whereas altering of the graph itself is not needed.

There are, however, also some shortcomings of our approach. For example, with more complex personalization, the rules can get complicated and lead to an

increased computational overhead. This can partly be addressed by implementing a “transfer graph” (i.e., a network connecting different transfer points, such as public transport stops, car- or bikesharing facilities, etc.) as part of the heuristic which could improve the efficiency, however, at the cost of reverting to a more rigid structure. Further, the scalability of such a system was only briefly discussed in this paper. The heuristic, at its present stage, can quickly generate a high number of possibly reachable states. However, by restricting modes to maximal distances, this can be avoided to some degree (e.g., if both origin and destination are far away from an intermediate stop, it does not make sense to consider using a tram at this stop, except to reach a “hub” for faster transportation).

Finally, our work can possibly be extended by expanding the heuristic to include other points of interest (POIs) or general user needs (cf. Eiter et al. 2012; Raubal et al. 2004; Bucher et al. 2015). For example, a user might be interested in a journey that includes a stop at a pharmacy, or knowing possibilities for meeting friends during a journey. Using concepts from time geography, we can compute sets of intermediate reachable POIs, and integrate them into the rule-based approach.

Acknowledgments This research was supported by the Swiss National Science Foundation (SNF) within NRP 71 “Managing energy consumption” and by the Commission for Technology and Innovation (CTI) within the Swiss Competence Center for Energy Research (SCCER) Mobility.

References

- Aissi S, Gouider MS (2012) Personalization in geographic information systems: A survey. *International Journal of Computer Science Issues (IJCSI)* 9(4): 291–299.
- Allemann D, Raubal M (2015) Usage differences between bikes and E-bikes. In Bacao F, Santos M, Painho M (eds) *Geographic information science as an enabler of smarter cities and communities*, Springer, Heidelberg, 201–217.
- Aoidh EM, McArdle G, Petit M, Ray C, Bertolotto M, Claramunt C, Wilson D (2009) Personalization in adaptive and interactive GIS. *Annals of GIS* 15(1): 23–33.
- Arentze TA (2013) Adaptive personalized travel information systems: a Bayesian method to learn users’ personal preferences in multimodal transport networks. *IEEE Transactions on Intelligent Transportation Systems* 14(4): 1957–1966.
- Bast H (2009) *Car or public transport—two worlds*. Springer, Berlin, Heidelberg, 355–367.
- Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck RF (2015) *Route planning in transportation networks*. Technical Report, Microsoft Research.
- Bast H, Funke S, Sanders P, Schultes D (2007) Fast routing in road networks with transit nodes. *Science*, 316(5824): 566–566.
- Bellman R (1956) On a routing problem (No. RAND-P-1000). Rand Corp, Santa Monica Ca.
- Broach J, Dill J, Gliebe J (2012) Where do cyclists ride? A bicycle route choice model developed with revealed preference GPS data. *Transportation Research Part A: Policy and Practice* 46 (10): 1730–1740.
- Bucher D, Cellina F, Mangili F, Raubal M, Rudel R, Rizzoli A.E, Elabed O (2016). Exploiting Fitness Apps for Sustainable Mobility – Challenges Deploying the GoEco! App. *ICT4S 2016*.

- Bucher D, Weiser P, Scheider S, Raubal M (2015). Matching complementary spatio-temporal needs of people. In Online proceedings of the 12th international symposium on location-based services, Augsburg, Germany.
- Campigotto P, Rudloff C, Leodolter M, Bauer D (2016) Personalized and situation-aware multimodal route recommendations: the FAVOUR algorithm. *IEEE Transactions on Intelligent Transportation Systems*, [arXiv:1602.09076](https://arxiv.org/abs/1602.09076).
- Cellina F, Rudel R, De Luca V, Rizzoli A E, Botta M, Raubal M, Bucher D, Weiser P (2016) Eco-feedback and gamification elements for sustainability: the GoEco! living lab experiment. Poster presented at the 6th European Transport Research Conference (TRA 2016). Warsaw, Poland.
- Dai J, Yang B, Guo C, Ding Z (2015) Personalized route recommendation using big trajectory data. In: *IEEE 31st International Conference on Data Engineering*, Seoul, 543–554.
- Delling D, Pajor T, Werneck RF (2014) Round-based public transit routing. *Transportation Science* 49(3): 591–604.
- Dibbelt J, Pajor T, Wagner D (2015) User-constrained multimodal route planning. *Journal of Experimental Algorithmics (JEA)* 19: 3–2.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1 (1): 269–271.
- Eiter T, Krennwallner T, Prandtstetter M, Rudloff C, Schneider P, Straub M (2012) Semantically enriched multi-modal routing. *International Journal of Intelligent Transportation Systems Research* 14(1): 20–35.
- ETSI (2005) Human factors (HF): user profile management. ETSI, Nice.
- Foerster T, Stoter J, van Oosterom P (2012) On-demand base maps on the web generalized according to user profiles. *International Journal of Geographical Information Science* 26(1): 99–121.
- Funke S, Storandt S (2015) Personalized route planning in road networks. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '15)*. ACM, New York.
- Geisberger R, Sanders P, Schultes D, Delling D (2008) Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In: *Experimental Algorithms*, Springer, Heidelberg, 319–333.
- Golledge R (1995) Path selection and route preference in human navigation: A progress report. In: Frank A, Kuhn W (eds) *Spatial Information Theory-A Theoretical Basis for GIS*, Vol. 988, Springer, Berlin, Heidelberg, New York, 207–222.
- Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2): 100–107.
- Huang H, Klettner S, Schmidt M, Gartner G, Leitinger S, Wagner A, Steinmann R (2014) AffectRoute – considering people’s affective responses to environments for enhancing route-planning services. *IJGIS* 28(12): 2456–2473.
- Jonietz D (2016) Personalizing Walkability: A Concept for Pedestrian Needs Profiling Based on Movement Trajectories. In: Sarjakoski T, Santos MY, Sarjakoski LT (eds) *Geospatial data in a changing world*. Springer, Heidelberg, 279–297.
- Krisp J, Keler A (2015) Car navigation – computing routes that avoid complicated crossings. *IJGIS* 29(11):1988–2000.
- Letchner J, Krumm J, Horvitz E (2006) Trip router with individualized preferences (TRIP): incorporating personalization into route planning. In: Porter B (ed) *Proceedings of the 18th conference on Innovative applications of artificial intelligence - Volume 2 (IAAI'06)*, AAAI Press, 1795–1800.
- Liu L, Xu J, Liao SS, Chen H (2014) A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication. *Expert Systems with Applications* 41(7): 3409–3417.
- Nuzzolo A, Crisalli U, Comi A, Rosati L (2015) Individual behavioural models for personal transit pre-trip planners. *Transportation Research Procedia* 5: 30–43.

- Raubal M (2009) Cognitive engineering for geographic information science. *Geography Compass* 3(3): 1087–1104.
- Raubal M (2011) *Cogito ergo mobilis sum: The impact of Location-Based Services on our mobile lives*. In: Nyerges T, Couclelis H, McMaster R (eds) *The SAGE Handbook of GIS and Society*, Sage Publications, Los Angeles, London, 159–173.
- Raubal M, Miller H, Bridwell S (2004) User-Centred Time Geography For Location-Based Services. *Geografiska Annaler B*, 86(4), 245–265.
- Raubal M, Winter S, Teßmann S, Gaisbauer C (2007) Time geography for ad-hoc shared-ride trip planning in mobile geosensor networks. *ISPRS Journal of Photogrammetry and Remote Sensing* 62(5): 366–381.
- Sanders P, Schultes D (2007) Engineering fast route planning algorithms. In: *Experimental Algorithms*, Springer, Berlin, Heidelberg, 23–36.
- Tsui A, Shalaby A (2006) Enhanced system for link and mode identification for personal travel surveys based on global positioning systems. *Transportation Research Record: Journal of the Transportation Research Board* 1972: 38–45.
- Yang B, Guo C, Ma Y, Jensen CS (2015) Toward personalized, context-aware routing. *The VLDB Journal* 24(2): 297–318.