# On the Way to a Multi-Modal Energy-Efficient Route

Matthias Prandtstetter, Markus Straub, Jakob Puchinger
AIT Austrian Institute of Technology GmbH
Mobility Department – Dynamic Transportation Systems
1210-Vienna, Austria, Giefinggasse 2
Email: {matthias.prandtstetter|markus.straub|jakob.puchinger}@ait.ac.at

***Abstract*—Within this paper, we present a flexible and expandable routing framework capable of finding multi-modal and inter-modal energy-efficient routes incorporating, among others, transportation modes such as public transport, electric vehicles, car-sharing, bike-sharing and walking. In contrast to conventional trip planning services, the proposed framework can evaluate routes not only with respect to travel distance or travel time but also with respect to energy used. In addition, range limitation by electric vehicles is incorporated into the routing request such that range-safety can be provided.**

## I. Introduction

Research in sustainable and energy efficient mobility is currently strongly motivated by increasing concerns about climate change and rising green house gas emissions. Electric powered mobility (e-mobility) is one of the directions currently taken in order to address those concerns. The market take-up of electric powered vehicles is slowed down due to the limited range of the vehicles and relatively long recharging times. Addressing those concerns is one of the major challenges of today's research in this area. However, the introduction of a new generation of vehicles can also be seen as opportunity for achieving sustainable changes in individual mobility behavior. In this paper we will try to address parts of theses challenges by proposing multi-modal routing methods taking into account the limited range of electric powered vehicles and their integration into a multi-modal traffic system.

Starting in the early 2000s online trip planning services became more and more available and quickly started to be an integral part of route planning—especially for private users. Such online trip planning services typically focus on computing the shortest and/or fastest route between two points. Sometimes other objectives like "the most beautiful" route can be requested. Furthermore, routing services specialized for other modes of transportation (MOTs) than *car* are offered which consider special constraints when planning a trip. E.g. a route planner for bicycle tours might prefer cycle paths instead of simply finding the shortest route which might lead over busy streets. However, to our best knowledge, no (commercial) route planning services exist which offer to compute *intermodal* routes incorporating *car*, *bicycle*, *public transportation*, *e-mobility*, *car-sharing*, *bike-sharing*, and *walking*. Obviously, only a reasonable subset of these MOTs is chosen when suggesting a route, but—theoretically—all of them could be used within a single route. In addition, to provide routes incorporating e-mobility, it is necessary to be able to (a) estimate the energy to be used along one specific route and to (b) compute the most energy-efficient route for a given origin-destination pair. Among others, both objectives are (major) goals of the Austrian research project *EMPORA*.

This work is partially funded by the Austrian Climate and Energy Fund within the "Electric Mobility Flagship Projects" programme (project emporA 2 under grant number 829096).

### A. Short overview of the EMPORA project

The EMPORA project brings together Austria's leading businesses from the automobile industry, infrastructure technology, energy supply and research sectors in order to integrate sub-systems, which are either new or currently in development, within innovative complete systems for electric mobility in a user-oriented and internationally coordinated way. The result is a national system solution, as well as the development and introduction of new technologies by Austrian companies.

One of the main objectives of the EMPORA project is to enable customers to integrate electric mobility seamlessly with other, foremost public mobility modes. Due to the intrinsically limited autonomous driving range of electric vehicles the combination of individual transport with public transport is more important compared to conventionally powered vehicles. This is achieved by providing users with an information system on hand held devices such as smart phones that allows them to plan multi-modal routes combining privately owned vehicles with public transport and car- and bike-sharing offers. The system supports pre-trip an on-trip planning and will consider real-time information about the current traffic situation, weather condition, user position and availability of shared vehicles and more. Route recommendations will also take into account user-specific preferences and the system will be able to learn individual route choices. In a final step the results will be demonstrated in Austrian cities including various real-world data sources and modes of transport.

The first step towards this goal was the development of multi-modal routing algorithms embedded in a flexible and expandable routing framework that will be presented in this paper.

### B. Structure of the Paper

In the next section we first present the general focus of this paper and continue in Section III with the presentation of methods for estimating energy consumption along a route. Section IV introduces a flexible method for representing multi-modal maps. Sections V and VI document experiments and complete the paper with some conclusions.

## II. Requirements for the Trip Planning Service

Since the goal was to provide a flexible yet efficient, intermodal trip planning service, the following requirements were determined:

- Multi-modality, i.e. different modes of transport (MOTs), have to be supported. In general, this means that the user should have the possibility to decide for which MOT a trip should be planned.
- Beside the computation of co-modal routes, i.e. provision of multiple routes with each incorporating exactly one MOT, the planning of *intermodal* trips should be provided. This means the resulting routes incorporate a subset of one or several MOTs from all MOTs that were pre-defined by the user.
- Furthermore, the user should have the possibility to define, which *characteristics* a route should comply to. E.g., when planning a bicycle tour, the user might be interested in the shortest distance route, the fastest route or the safest route. In addition, it should be possible to compute an *energy-efficient* route for motorized individual traffic, i.e. a trip which minimizes energy consumption of electric cars as well as conventional vehicles (in contrast to fastest or shortest routes).
- All computations performed should be as *efficient* as possible, meaning that although the requests by the users could be relatively complex, responses should be available in at most few seconds. At the same time, expansion of the service, e.g. introduction of additional MOTs or weighting functions, should be possible at any time—i.e., the system has to be *flexible*.
- Finally, the system should be *independent* from a particular map representation, i.e., maps should be easily exchangeable.

Based on this list of requirements, it can be seen that the development of such a system is a rather complex task including many uncertainties with respect to future demands.

Let us highlight the requirement to compute energy-efficient routes: Since the trip planning service shall be capable of incorporating e-mobility, it is important to address *range anxiety*, i.e. the fear of e-car drivers that the power available in the battery will not suffice to reach the target. In the worst case, the car could stop somewhere on the track without any possibility to reload the battery. Having methods for estimating the energy necessary for a planned trip, including methods for proposing suitable charging stations—if necessary—a feeling of security and assurance can be given to the driver and the acceptance of e-mobility can be increased.

### *Related Work*

Obviously, Dijkstra's Algorithm [1] for computing shortest routes in networks forms a basis for developments in the area of route computations. However, there are many other algorithms focusing on the computation of paths in networks: For example, in [2] the authors focus on speed up techniques for accelerating the computation of single routes. Although the computation times of individual routes can be significantly reduced in comparison to previously published work, extra pre-processing steps are necessary with the disadvantage that (a) they are rather time-expensive and (b) (unforeseen) changes in the network (e.g. traffic-dependent closings of roads) cannot be regarded (due to the pre-processing).

In [3], the authors recently focused on the computation of multi-modal journeys. In contrast to the classical shortest path problem, the authors try to incorporate multiple optimization criteria leading to an approach computing a Pareto set which is then sorted using an evaluation function originated in fuzzy logic.

Finally, in [4] route planning for electric vehicles is presented. In this work, all basic questions with respect to finding energy-efficient routes are addressed. However, weights of the arcs in the network (i.e. needed energy for traversing an road segment) are assumed to be given (and constant). That is, no traffic dynamics are respected by the proposed approach.

## III. Computation of Energy Consumption

As outlined in the previous section, it is important to have methods for estimating the power consumption when trying to incorporate e-mobility into trip planning services, cf. also [5], [6], [7]. For this purpose, a (basic) power consumption function is needed which can be built upon basic energy functions known from physics. Five main components can be determined for estimating power consumption in e-cars:

- **kinetic energy**: Obviously, the energy for gaining the (maximum) speed is necessary. We know from physics that this energy is saved in an object of mass $m$ as *kinetic energy* $E_{kin}$, which can be computed by
$$E_{kin} = \frac{1}{2} \cdot mv^2 \tag{1}$$
with $v$ being the speed of the object (car).
- **potential energy**: In addition, *potential energy* is stored in each object according to its height. Since streets are typically not level but incline (either up, down or both), the energy for raising the car needs to be spent according to
$$E_{pot} = mGh \tag{2}$$
with $m$ being the mass of the object, $h$ being the height the object is raised and $G$ denoting acceleration due to gravity.
- **drag**: Based on the shape of the car and its size, *drag* occurs. The energy needed to overcome this drag can be computed by
$$E_{drag} = c_W A \frac{1}{2} \rho v^2 \cdot l \tag{3}$$
where $c_W$ denotes the drag coefficient (i.e. the shape of the car), $A$ is the cross-sectional area (i.e. the size of the car), $\rho$ corresponds to the density of the air and $v$ is the current speed of the car. Variable $l$ finally introduces the length of the route into the formula.
- **rolling friction**: Depending on tire type and pressure, the *rolling friction* needs to be considered, where
$$E_{roll} = c_R F_N \cdot l \tag{4}$$

corresponds to the energy needed for overcoming the rolling friction, with $c_R$ denoting the rolling resistance coefficient and $F_N$, the normal force, can be estimated with $F_N = mG$. Variable $l$ denotes, again, the length of the route.

- **auxiliary consumers**: Finally, several *auxiliary consumers* like air condition, heating, head lights, or radio have to regarded, as they also reduce the range of an e-car:

$$E_{aux} = \sum_{\text{dev}} P_{\text{dev}} \cdot t \tag{5}$$

Here, the generic variable $dev$ denotes an auxiliary consumer and $P_{\text{dev}}$ the power of $dev$ and $t$ the travel time along the route.

Obviously, the total energy consumed along a route computes as the sum over the main components:

$$E_{\text{tot}} = E_{kin} + E_{pot} + E_{drag} + E_{roll} + E_{aux} \tag{6}$$

However, be aware that $h$ (used in $E_{pot}$) needs to be computed based on the topography of the route (i.e. all slopes have to be regarded). In addition, $E_{\text{kin}}$ will be a sum over individual terms where each term corresponds to a speed profile: E.g., when at the beginning the car is accelerated to $30\ \text{km/h}$, then slowed down to $20\ \text{km/h}$, sped up to $40\ \text{km/h}$ and then stopped ($0\ \text{km/h}$), we need to compute the kinetic energy necessary for accelerating to $30\ \text{km/h}$ and to accelerate from $20\ \text{km/h}$ to $40\ \text{km/h}$, i.e.

$$E_{kin} = \frac{m}{2} \cdot \Big[(30\ \text{km/h})^2 + \Big((40\ \text{km/h})^2 - (20\ \text{km/h})^2\Big)\Big] \tag{7}$$

for this specific example.

In addition, when using an electric car, there is the possibility to use recuperation for recovering energy when driving downhill or breaking. Therefore, to be consistent with the previous example, we need to introduce $E_{kin}^{-}$ denoting the kinetic energy which could be recovered, e.g.

$$E_{kin}^{-} = \frac{m}{2} \cdot \Big[\Big((20\ \text{km/h})^2 - (30\ \text{km/h})^2\Big) + (-40\ \text{km/h})^2\Big] \tag{8}$$

for our example. In addition, the energy recovered due to downhill roads need to be analogously regarded via a term $E_{pot}^{-}$, resulting in

$$E_{\text{tot}} = E_{kin} - E_{kin}^{-} + E_{pot} - E_{pot}^{-} + E_{drag} + E_{roll} + E_{aux} \tag{9}$$

for expressing the complete energy consumption along a specific route.

Unfortunately, the efficiency of the drive system of an e-car is not 100%, meaning that some energy is lost on its way from the source to the road. This happens mainly due to friction. According to experimentally determined values provided by project partners, efficiency for e-cars is on average about 80% during drive and only 60% during recuperation such that the expression for estimating the total energy needed along a route must be extended to

$$E_{\text{tot}} = \frac{E_{kin} + E_{pot} + E_{drag} + E_{roll} + E_{aux}}{0.8} - \left(E_{kin}^{-} + E_{pot}^{-}\right) \cdot 0.6 \tag{10}$$

However, we have to keep in mind that the superior goal is to reduce range anxiety. Although—using the above formulas—it is possible to theoretically estimate the energy consumption along a route, it is necessary to be on the safe side. Therefore, two assumptions are made which over-estimate the total energy consumption. For this purpose, we first assume that no recuperation is available. This assumption is essential since strong decelerations reduce the efficiency of recuperation, i.e. *less* energy would be recovered. Second, we assume that the car can instantly accelerate (and decelerate) to (and from) the maximum speed, leading to an over-estimation of $E_{drag}$.

Please note, that beside the fact that the energy consumption is over-estimated by these two assumptions, we do not have to consider negative energy weights on the edges (which could have happened when incorporating recuperation). This leads to the possibility to apply the well-known *Dijkstra's algorithm* [1] for computing shortest paths in a network with non-negative edge-weights.

## IV. Map Representation

As outlined above, one goal—among others—is to build a flexible, multi-modal trip planning framework which can be easily extended. From a theoretical point of view, trip planning is typically performed on a (directed) network $N(D, \mathrm{w})$ where $D(V, A)$ represents a digraph, i.e. a directed graph, with $V$ denoting the set of nodes (or vertices) and $A$ representing the set of directed edges (arcs). Function $\mathrm{w} : A \to \mathbb{R}_0^+$ assigns a non-negative weight to each arc in the digraph $D$. Using such a setup, *Dijkstra's algorithm* [1] for finding shortest routes in a network can be applied. However, this approach is not directly capable of considering real-world constraints like *turn restrictions*, *limited access* or multi-modality. Even more, one can easily think of weight functions $\omega : A \times A \to \mathbb{R}^+$ assigning a weight $\omega(a_1, a_2)$ to arc $a_2$ depending on the previously traversed arc $a_1$. Obviously, this weight function $\omega$ cannot be handled by Dijkstra's algorithm. Therefore some (partly straightforward) extensions are necessary when trying to integrate (all of) these functionalities. In the following, we will outline the basic ideas applied and highlight some (typical) pitfalls which occur when performing these tasks.

The first design decision to be made in such a situation is to decide on the structure of the underlying network: Either all computations are performed on one *annotated* network where each arc represents a physical infrastructure (e.g. a road) and access restrictions (e.g. which type of vehicle is allowed to travel along this road) are represented by *annotations*. Otherwise, one can introduce a *layered* network where each *layer* of the network corresponds to one possible MOT (e.g. walking or public transport). In this case, an arc exists in one layer only if the corresponding MOT is allowed to traverse the corresponding road.

**Ensure:** an intermodal, layered network

```
generate walking network;
for all MOTs m to be added do
    for all nodes v of MOT m do
        find closest node v' in walking network;
        if distance between v and v' ≤ threshold then
            introduce intermodal arcs (v, v') and (v', v);
        end if
    end for
end for
```

Fig. 1. Algorithm for building the intermodal layered network.

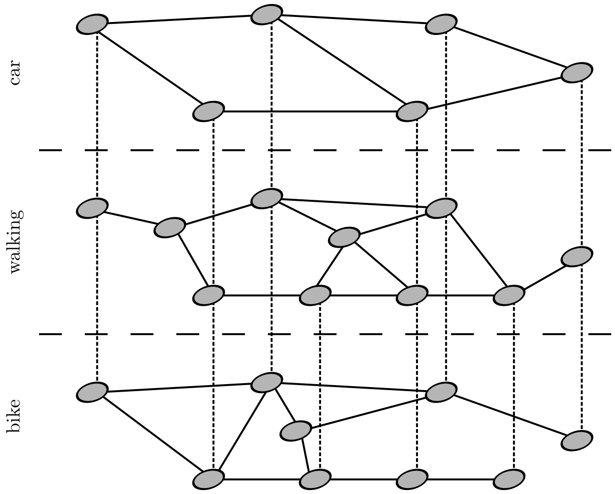
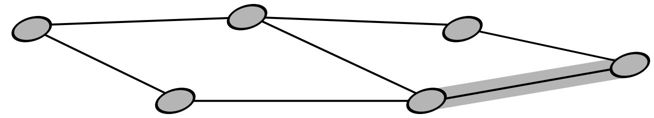


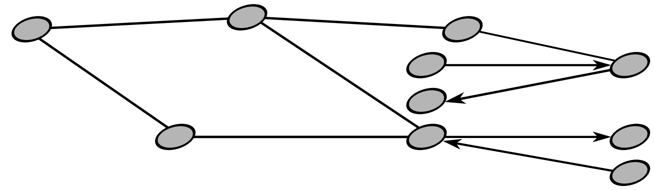

Fig. 2. Layered network graph. Note that connections between the layers are available at certain nodes (dotted lines).

To be more flexible, we decided to employ the layered network approach. To facilitate computing multi-modal routes, it is necessary to link the different layers with each other. For this purpose, we decided to use the walking graph as basis (since each route can only start and end at positions where walking is allowed). The layers for the other MOTs are then connected to the basis layer by applying the following procedure, see also Alg. 1: For each vertex of the newly added layer, we decide which walking vertex, i.e. a vertex in the basis layer, is closest. Unless this (closest) walking vertex is farther away than a pre-defined threshold, we introduce two *intermodal* arcs (one in each direction) connecting the walking vertex with the node of the newly added layer implying that a change from walking to the current MOT (and vice versa) is allowed at this position. The weight of these newly introduced intermodal arcs can be chosen according to their *interpretation*. E.g., when an arc represents a change from car to walking, then the time needed for traversing this arc can incorporate average time for finding a parking lot. In addition, parking restrictions or the nonexistent parking space can easily be considered by adapting the weights on those arcs. A sketch of the resulting layered network can be seen in Figure 2.

Choosing this layered approach contains several advantages: First, Dijkstra's algorithm can be directly applied. Second, the framework is flexible with respect to adding (and removing) additional MOTs on the fly (as long as walking is available). Third, it is possible to incorporate special situations via the weighting function: E.g., when your car is parked at a specific location, then a change from walking to your car can be easily realized by setting the weight of the corresponding intermodal edge (where your car is parked) to 0, while all other intermodal edges (from walking to your car) are set to infinity. Even more, (dynamic) mobility concepts like car- or bike-sharing can be realized by changing the weight function according to the availability of vehicles at rental stations.

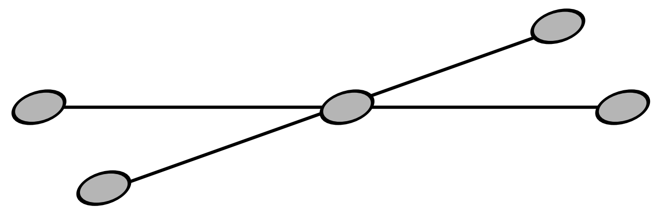


(a)

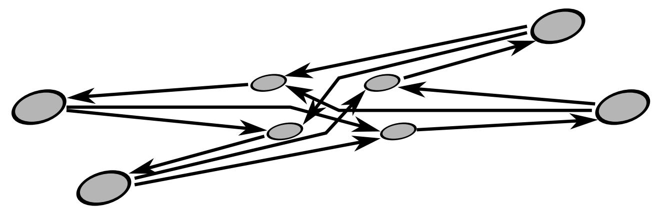


(b)

Fig. 3. Network adaptions for handling access restrictions in living streets. The original network (a) with a restricted access road (shaded) and the extended network (b) with additional nodes and arcs.



(a)



(b)

Fig. 4. Necessary adaptions for incorporating turning restrictions. In this example, we assume that for all directions left turns are prohibited.

For incorporating special access restrictions as given in residential areas (e.g. living streets where only source and targeted traffic is allowed), we add for each living street two arcs, where for one only the incoming node and for the other only the outgoing node is connected with the remaining network, cf. Fig. 3.

Finally, it is necessary to incorporate turning restrictions. This can be easily achieved by duplicating nodes for junctions where turning restrictions are existing: A new vertex is introduced for each *outgoing* arc (at this junction). Incoming arcs are now duplicated and connected to all outgoing arcs where access is permitted, cf. Fig. 4.

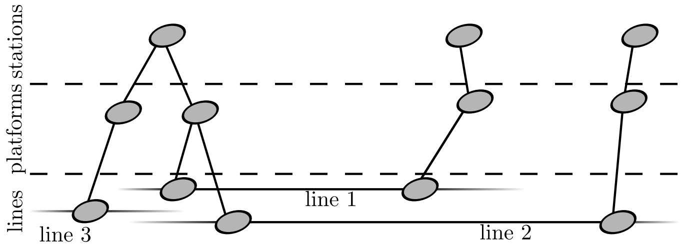



Fig. 5. Scheme of a public transportation network.

### A. Public Transportation

Beside this general techniques, the public transportation (PT) network needs a little more attention. In contrast to (motorized) individual traffic, PT is characterized by the fact that different lines are interconnected with each other via common stops. While sometimes changing from one line to another is possible without changing platforms, in other situations (longer) walks are necessary for transfer. Therefore, the PT network consists of several types of vertices and arcs which can be interpreted as follows, see also Fig. 5: Vertices can either represent stations, platforms or stops, where *stops* are the actual stopping positions of the vehicles and one stop is serviced by exactly one line. Several stops can be located at the same *platform*. One or more platforms result in a *station*. For arcs, an analogous concept is applied: Stops of the same line are directly connected via arcs. Obviously, there are also arcs between stops and the platform the stops are assigned to. Finally, arcs are introduced between platforms and the station they are associated to.

Using this concept, it is possible to model walking times/distances for transfers between different stops. Please note however, that based on this principle not all edges within the PT network correspond to journeys in mass transportation mediums but some represent walks, escalators, elevators, or stairs.

### B. Weight Functions

As already outlined above, routing is done based on the multi-modal network(s) as well as the weight function applied to the arcs. Using dynamic weight functions (dependent on departure time, current traffic situation, current weather condition (and forecast), current availability of vehicles in sharing systems, etc.) it is easy to provide (highly) dynamic trip planning services. However, some weight functions might even be more complex: For example, the estimation the energy consumption along an arc in the network is not only dependent on the speed traveled along this arc but also on the speed traveled on the previous arc (since de/acceleration has to be regarded). For this purpose, it is necessary to realize the previously mentioned weight function $\omega(a_1, a_2)$ such that Dijkstra's algorithm can still be applied (which can only correctly handle simple weight functions). Fortunately, $\omega(a_1, a_2)$ can be easily realized by transforming the underlying graph such that each node is replaced by an edge and each edge is replaced by a node. Obviously, an efficient implementation of this procedure does not actually need to perform this transformation in advance but can do this on the fly.

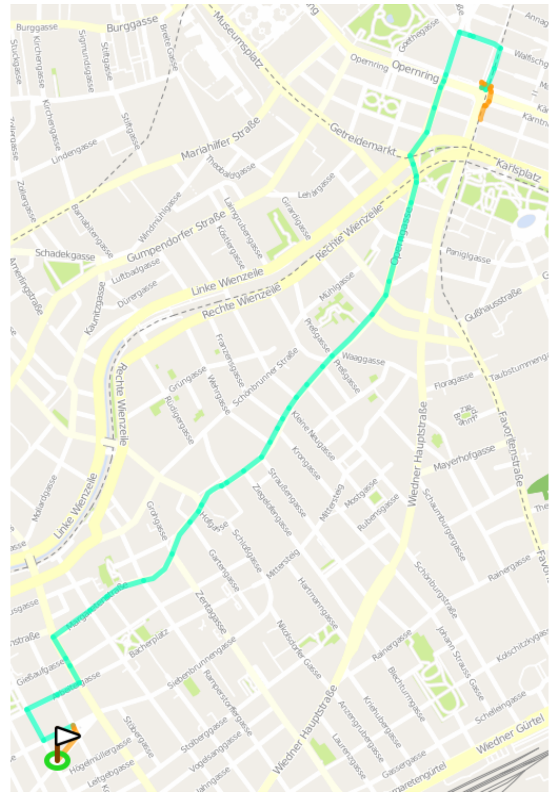



Fig. 6. An intermodal route in Vienna from the Opera into the $5^{th}$ district. The involved MOTs are walking (orange) and bike-sharing (cyan).

Furthermore, minimal adaptions are necessary when routing in a PT network: While for individual transportation each link can be accessed at *any* time, PT arcs can only be traversed whenever a PT vehicle (e.g. bus) is currently operating the line. Therefore, the travel time for the corresponding edges is extended such that the (initial) waiting time is contained in the weight of the first edge.

### C. General Route Planning

Equipped with the above presented methods, it is now possible to compute multi-modal routes incorporating different modes of transport (MOTs) including electric vehicles. For this purpose, a user has to define which MOTs are relevant for her. In addition, the parking spot of the car (or bike) has to be announced since otherwise a transition to this MOT is not possible. The finally computed route suggests which MOTs should actually be used such that the optimal route can be obtained.

Although theoretically possible, there is no energy consumption computation for multi-modal trips. The objective value to be optimized (during the routing) is in fact a weighted sum of individual parameters (e.g. travel time, energy consumption when using electric vehicles, distance travelled, etc.).

## V. Experiments

To highlight the performance of the proposed framework, we performed some tests on an laptop with an Intel Core2 Duo T9600 (2.8 GHz) CPU and 8 GB RAM. Since the performance of the applied algorithms is dependent on the size of the underlying graphs, we decided to test the methods on a small graph covering Vienna with about 64 000 arcs and a larger network covering the three Austrian states Vienna, Lower Austria and Burgenland with about 530 000 arcs.

For the smaller graph, it took on average (over 2 000 randomly chosen routes) $43\,\mathrm{ms}$ to compute one shortest distance route, while the computation of one minimum energy route took about $185\,\mathrm{ms}$. For the larger setup, computation times raised to $438\,\mathrm{ms}$ (shortest distance) and $2\,203\,\mathrm{ms}$ (minimum energy route), respectively. Since, however, energy routing uses the (currently) most complex weight function (incorporating several dynamic data sources like weather forecast and current traffic information), it can be concluded that response times of the system are satisfactory.

Please note, that in this tests Dijkstra's algorithm was always executed on the complete graph, i.e., the algorithm was not terminated as soon as the target node was reached but a minimum weight route to *all* nodes in the graph was computed. This was mainly done, such that the worst case performance of the framework could be tested.

If applied within a trip planning service, it is most likely that seldomly requests share the same source and/or destination point such that a more efficient, bidirectional variant of Dijkstra's algorithm can be applied which significantly improves the runtime performance. The bidirectional variant searches the minimum weight route simultaneously starting from the source (looking forward) and the target (looking backward). As soon as the forward and backward looking search meets, the minimum weight route is found and the algorithm can be early terminated. However, computation times for single routes strongly depend on the distance between source and target when applying this optimization technique.

To summarize, energy routing is about five times slower than shortest distance routing. Yet, the performance is convincing having the highly dynamic weight function in mind. Since the worst case complexity of Dijkstra's algorithm is $O(n \cdot \log(n))$, with $n$ being the number of arcs in the network, the 8-fold increase in edges lead to 10- to 11-fold increase in calculation time.

In addition, we present an intermodal route through Vienna in Figure 6, where walking and bike-sharing are incorporated. Please note, that computation times do not increase when intermodality is considered, since the runtime performance solely depends on the number of edges contained in the network.

Finally, it has to be highlighted that during the $19^{th}$ ITS World Congress [8] a prototype incorporating minimum energy routing developed in the EMPORA project was presented during a demonstration session. For this purpose, the minimum energy routing was extensively tested in real-world conditions (including an integration of the minimum energy routing procedure in a mobile navigation device) [9].

## VI. Conclusions and Future Work

Within this paper, we presented a systematic approach providing an efficient but yet flexible framework for computing intermodal routes. Among others, one of the main functionalities is the provision of energy efficient routes, i.e. routes minimizing the energy consumption when traveling from A to B. Using the proposed approach, several rather complex real-world restrictions can be incorporated into trip planning such that still the well-known (and efficient) Dijkstra's algorithm [1] for computing shortest routes in a given network can be applied. Most importantly, the system can be used for estimating the energy consumption along a planned route such that range anxiety arising in connection with e-mobility can be minimized and range-safety can be provided to the user.

Although the framework is rather comprehensive, several future topics are still left open to be solved (or included): For example, the energy computations for routes are based on a rather basic model. At this time, we do not have had the opportunity to validate the model via test drives using electric vehicles since the (detailed) energy values (battery capacities, etc.) are not open accesible. It is, however, planned to focus on that topic in the future.

Furthermore, it is of great importance to incorporate behavioral differences between users when planning an intermodal trip. For some users, it is no option to take a bus (e.g. due to accessibility) while for others this might be the best option. Therefore, route choice and mode choice modeling will be incorporated into the trip planning service. However, such a functionality can only be provided based on a sufficient data basis which has to be collected in real-world scenarios (e.g. by tracking via mobile devices). For this purpose, user provided data is currently gathered via other research projects like MyITS [10].

Furthermore, route alternatives need to be offered to the users such that the acceptance of such a system can be further improved. E.g., for many users the display of two or three alternative routes improves the driving experience since additional constraints (like shopping destinations or scenic routes) can be much more easily incorporated during pre-trip planning.

## References


[1] E. W. Dijkstra, "A note on two problems in connexion with graphs." *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[2] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, "Exact routing in large road networks using contraction hierarchies," *Transportation Science*, 2012, in press.

[3] D. Delling, J. Dibbelt, T. Pajor, D. Wagner, and R. F. Werneck, "Computing multimodal journeys in practice," in *Experimental Algorithms*, ser. LNCS, V. Bonifaci, C. Demetrescu, and A. Marchetti-Spaccamela, Eds. Springer Berlin Heidelberg, 2013, vol. 7933, pp. 260–271.

[4] J. Eisner, S. Funke, and S. Storandt, "Optimal route planning for electric vehicles in large networks," in *AAAI*, W. Burgard and D. Roth, Eds. AAAI Press, 2011.

[5] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Springer Berlin Heidelberg, 2013.

[6] K. Kraschl-Hirschmann and M. Fellendorf, "Estimating energy consumption for routing algorithms," in *Intelligent Vehicles Symposium*. IEEE, 2012, pp. 258–263.

[7] K. Kraschl-Hirschmann, "Energieorientierte straennetzbewertung fr routensuchverfahren," Ph.D. dissertation, Technische Universität Graz, 2013.

[8] "19th ITS World Congress," October 22–26 2012, lasted accessed May 2013. [Online]. Available: http://2012.itsworldcongress.com/

[9] M. Fasching, "Linking electro mobility, eco-routing and charging infrastructure," in *Proceedings of the 19th World Congress on Intelligent Transport Systems*, 2012, p. Paper Nr. EU00320.

[10] T. Eiter, T. Krennwallner, M. Prandtstetter, C. Rudloff, P. Schneider, and M. Straub, "Semantically enriched multi-modal routing," in *Proceedings of the 19th World Congress on Intelligent Transport Systems*, 2012, p. Paper Nr. EU00427.