A Multi-Modal and Multi-Objective Journey Planner for Integrating Carpooling and Public Transport

Jafar Jamal, Roberto Montemanni, David Huber, Marco Derboni, and Andrea E. Rizzoli
Dalle Molle Institute for Artificial Intelligence (IDSIA – USI/SUPSI), Manno, Switzerland
Email: {jafar, roberto, david, marco.derboni, andrea}@idsia.ch

***Abstract*—SocialCar is a research project that aims at integrating carpooling with traditional transportation systems in urban areas, while benefiting from social media to enhance the user's experience. The system is based on route planning and ride matching algorithms to provide the users with alternatives for their trips. In this work, we overview the multiple approaches in the literature to model transportation networks and carpooling services, and a route planning algorithm which integrates multiple transportation types together. Finally, the performance measures of the route planner are reported.**

***Index Terms*—multi-modal routing, temporal networks, time-dependent graphs, carpooling**

I. Introduction

Travelling for short or long distances has become a daily activity for most people worldwide. For some commuters the preferred mean of transportation is a car. In such case, many drivers use navigation systems to easily find the preferred path from origin to destination (using a service such as Google Maps). On the other hand, if a commuter prefers to use public transportation, there are many web services that offer to plan a trip using public transportation[1,2]. In some cases, using multiple means of transportation in a single trip would be a real advantage. For example, the commuter can drive to the nearest station and use public transportation to reach the desired destination. Another effective mean of transportation is carpooling, which can be defined as the cooperation of two or more commuters regarding the use of a single vehicle to meet their mutual commuting needs.

The use of carpooling has multiple benefits, such as reducing the number of vehicles participating in the transportation system, and it also benefits the individuals participating in the carpool. Some of the benefits include reducing fuel costs on the participants of the carpool, reduced toll costs where applicable, and potentially it may reduce driving stress for the passengers in the vehicle, there may also be some benefits in the social aspect. Governments in some countries such as the United States and parts of Europe encourage the use of carpooling.

SocialCar[3] is a research project that aims to integrate carpooling into existing mobility systems by means of powerful planning algorithms and integration of big data from public transportation, carpooling, crowd sourcing, and social networks. The project's mission is to design, develop, test, and roll-out a service that simplifies the travel experience of citizens in urban and peri-urban areas. SocialCar differs from existing carpooling services by not being a commercial service that offers a point to point unimodal service, but it fully integrates the private and the public transport networks, taking advantage of the best of both offerings. However, SocialCar users – that is both passengers and drivers can also benefit from existing carpooling services. These services are interfaced with SocialCar so that the data on carpool offers becomes available to a wider community.

The contributions of this work to the SocialCar project are restricted to the development of the trip planner, which includes developing a route planning algorithm, modeling the multi-modal transportation network, and the interaction of the algorithm with the provided data.

II. Related Work

The basis of many state-of-the-art algorithms for route planning forms Dijkstra's Algorithm [Dijkstra, 1959] in the area of shortest path computation. For road networks, many algorithms exist to improve the performance of calculating a shortest path from one location to another. Among these are A* [Zeng *et al.* 2009], Arc-Flags [Möhring *et al.* 2007], and Contraction Hierarchies [Geisberger *et al.* 2007, 2012].

Transit networks are a spatial representation of bus, trains, and other transit routes available in a specified area. In such a network each transit route is modeled with links representing the path it follows, and nodes representing the stops along the path. Transit networks are considered to be more complicated than road networks; therefore several models exist for transit networks. The two most common models are time-expanded and time-dependent networks [Pajor, 2009]. For the time expanded model the same algorithms used for the road network can be applied. An approach used

Manuscript received July 1, 2017; revised November 11, 2017.
[1] http://www.maps.google.com
[2] http://citymapper.com
[3] http://socialcar-project.eu

doi: 10.18178/jtle.5.2.68-72

for transit networks is to pre-compute transfer patterns [Bast *et al*. 2007]. This method leads to query-times of a few milliseconds.

Some approaches for transit network consider walking to transit stations, and others consider different modes of transportation restricted to a certain hierarchy. Braun (2012) and Delling *et al*. (2012) considered a less restricted multi-modal networks. They show in their work that using Pareto Sets with multiple criteria enlarges the set of optimal paths to become impractical. Moreover, many paths are very similar, and the query times increase to the order of minutes. Braun represented a model restricting approach which significantly reduces the size of the Pareto sets. However, Braun considered his approach to be too restrictive relative to preserving quality [Brodesser, 2013].

Multi-criteria optimization is also receiving growing attention. Among the considered criterion are total duration of the trip, walking time, number of transfers, and cost. In section 4 we will however follow such an approach.

III. Problem Description

As transportation networks are becoming more interconnected and available means of transportation are increasing, enhancement and variations of traditional route planning algorithms are becoming of more and more importance. The problem of route planning involving different means of transportation is called multi-modal route planning [Brodesser, 2013]. The goal of the problem is very simple. Given a source and target locations in the transportation network, and a departure time, with the desired means of transportation, the route planner should return an optimal route with respect to travel time (and possibly other factors such as, reduced waiting time, reduced number of modal changes and bus changes) that shows us which roads and means of transportation to use. In order to solve such a problem, the different transportation options needs to be structured as a multi-layer temporal network. The multi-modal network is strictly related to the algorithms that can be used to solve the planning problem, so its design and implementation depends on the specific algorithms to be applied to it.

A. The Multi-layer Temporal Network

In order to create alternative route solutions for users willing to move from origin to destination selecting alternative transportations, the algorithms used need an underlying network. The network must be made available as a *graph*, composed by *nodes*, which are connected by *arcs* (connections/links). The nodes represent the junctions in the network, from where links depart. A link connects two nodes.

The network is composed of *layers*, where each layer represents a mode. Nodes that are present in multiple layers simultaneously represent intermodal connection points. Interlayer links that connect two layers together represent the travelling time (by foot) required to transit from a transport mode to another. The nodes where a modal change can take place are defined as *switch points* in the network. Fig. 1 shows an example of a multi-layer temporal network. The layers associated with scheduled transport are defined as *temporal networks* [Gallotti et al. 2015][Holme and Saramäki, 2012]. Edges in the transportation network represent a segment in a route, while the nodes represent stops/stations.

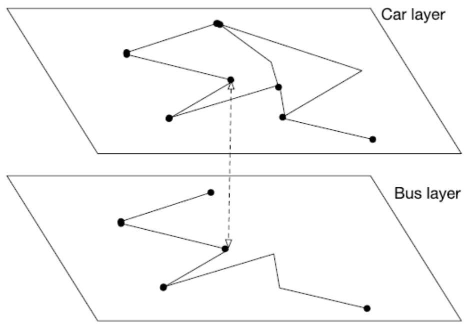

Figure 1. Two different layers contain the graph for routing. The dashed line represents interlayer links which allow a modal change across a selected route.

Modeling a public transportation network requires an additional level of complexity with respect to a road network, as routes are possible only when services are scheduled. In order to represent transport services offered on top of the transfer network, each node in the public transportation network is annotated with a list of departure times containing all the routes departing from that node. The timetable information (bus number, arrival and departure time, and destination) is used to generate the temporal network representation. Each ride from the station to the next one is represented with a directed edge with duration equal to the difference between the arrival time at the destination node and the departure time at source node. The resulting network is called a time-expanded graph [Köhler et al. 2002]. The main advantage of using this representation is that standard algorithm for shortest path(s) work with slight modifications, as the resulting network is static. On the other hand, a drawback is that the size of the resulting network could be large, which can lead to substantially longer computation times, with respect to other methods. However, preliminary experiments suggested that this approach is suitable for the needs of our application. Fig. 2 shows an example of a time-expanded graph.

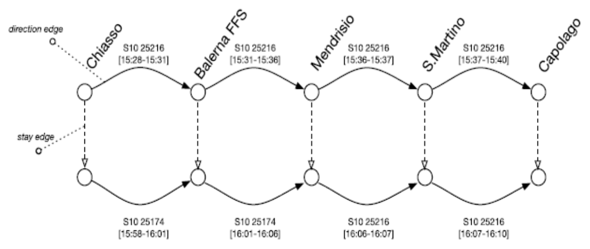

Figure 2. A timetable converted into a time-expanded digraph. Each edge has the departure time and arrival time between the consecutive nodes.

In multilayer networks, each transport mode is represented in a separate layer (Fig. 1), and each transport line can also be represented as a different layer. Solutions where the traveler can switch modes and lines have been defined as switch points. Such a solution is described in Fig. 3. The road network is therefore considered as another layer on top of the previous ones, but no time constraints are fixed on the graph edges.

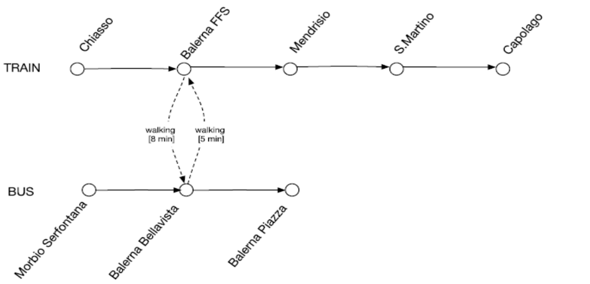

Figure 3. The passenger can switch to the bus line through the dashed edges.

Carpooling services are a hybrid transport mode halfway between public transportation (in a sense that carpoolers tend to stick to fixed schedules) and private transport (carpoolers are not bound to a specific route, as they can change the route dynamically depending on traffic conditions). For the before mentioned reason, it is possible to represent carpooling services as a public transportation service, while deviation will be represented using real-time feed. The real-time feed contains the essential information that is needed to describe alterations to previously planned routes, such as cancellations, delays, and updated routes. Another crucial element is the availability of places on the car (residual capacity). Unlike public transportation, cars have very limited capacity, and as soon as all available seats are filled, the car is no longer available as a service; therefore real-time feed must also be enriched with such information.

Fig. 4 represents a typical situation. The car starts the trip from a source point (possibly place of residence) with the driver and one passenger. At the next stop two passengers are picked up and then the car is fully occupied, which means that there is no residual capacity to serve other passengers, such information is used by the route planner so that this particular service is not considered as a potential solution.

B. Route Planning and Carpool Matching

One of the main goals of our team is to develop a fast and efficient method to be able to integrate route planning and carpool matching into a single algorithm. This can be achieved by specializing algorithms of the Dijkstra family. Dijkstra's algorithm computes a shortest path in a graph from a given source to all other nodes. Since carpooling routes are treated as public transportation the resulting graph is multilayer temporal network.

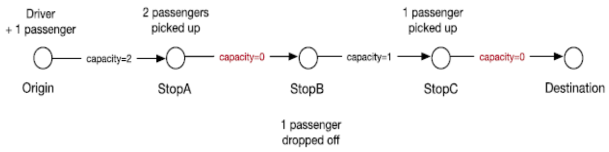

Figure 4. A carpooling service with pick-ups and drop-offs.

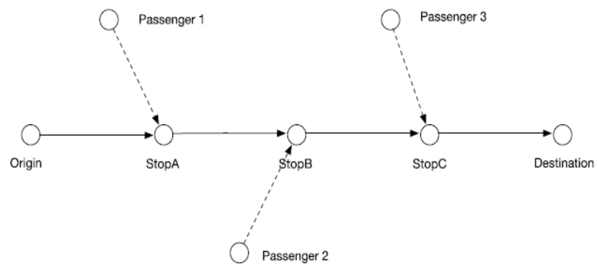

Figure 5. The carpoolers are picked up at pre-determined points along the optimal desired path from origin to destination.

The carpool matching algorithm we considered, where a car cannot change its original route, and passengers need to travel to the closest possible stop. As shown in Fig. 5, the car travels from origin to destination along a pre-determined route. The passenger can be picked up at intermediate stops that in our implementation, the stops coincide with public transport stops in order to favor modal changes.

IV. Shortest Path Algorithm and the Objective Function

Dijkstra's algorithm [Dijkstra, 1959] for finding single source shortest path(s) works as follows: let the node at which we are starting be called the initial node. Let the distance of node y be the distance from the initial node to y. Dijkstra's algorithm will assign initial distance values and will try to improve them step by step. In our case, the distance is the travelling time between two nodes, as a simple objective function, or a weighted sum of multiple factors.

In common presentations of Dijkstra's algorithm, initially all nodes are entered into the priority queue. This is however not necessary, the algorithm can start with a priority queue that contains only one node (the initial node), and insert new nodes as they are discovered (by checking if the node is already in the queue, if it is we decrease its key, otherwise we insert it). Not inserting all nodes in a graph makes it possible to find the shortest path even for graphs which are too large to represent. The original algorithm does not use a *min-priority queue* and runs in time $O(V^2)$ where $|V|$ is the number of nodes. However, another implementation based on a min-priority queue runs in time $O(|E| + |V|log|V|)$ where $|E|$ is the number of edge.

Dijkstra's algorithm solves the *earliest arrival time problem* by setting the weights of the arcs to the time needed to travel between two nodes instead of the distance between them. However, in a multi-layer temporal network, the travel time of an arc depends on both the arrival time at the source node, and the travel mode used to traverse the arc. In most route planning algorithms, other factors are taken into consideration on top of the travelling time. Those factors are mainly, travelling time on foot, time spent in transportation, waiting time for public transportation, and the number of bus changes. All those factors must be considered in the objective function to be minimized as a weighted sum of all the factors.

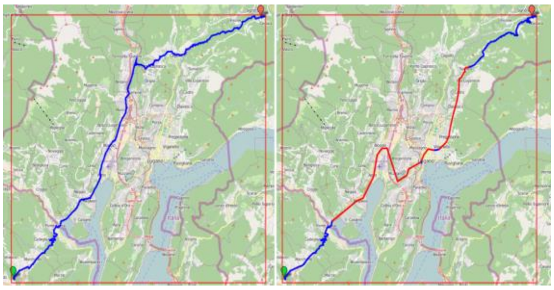

Figure 6. Worst case scenarios of the two examples. The picture on the left considers the road network alone. The picture on the left considers both the road and transit networks.

Our goal is to minimize the function *T*, where α, β, γ are constants, *t* is the transportation time, *w* is the walking time, *wt* is the waiting time, and *b* is the number of bus changes.

$$T = t + \alpha w + \beta wt + \gamma b$$

The weights of the function are determined experimentally based on the kind of area the network is covering, and which modes of travelling to favor more than others. It is also possible to generate multiple solutions by changing the values of the weights and calling the algorithm again, to generate a solution with more bus changes but less total travel time, or increased walking time and less transportation use.

V. Experimental Results

All the experiments in this section are performed on a map portion that covers a part of canton Ticino in Switzerland. The road network contains approximately 98k (98’782) nodes and 205k (205’282) arcs, while the transit network contains approximately 500 (571) nodes and 66k (66’397) connections.

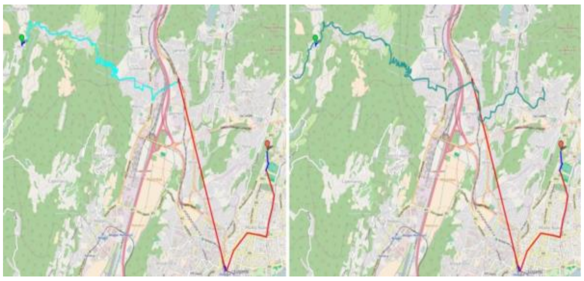

Figure 7. The picture on the left show a trip plan solution with carpooling, while the picture on the right shows the same solution with the carpooler’s original route drawn in green.

Figure 8. The solution suggested when considering traditional transportation methods alone.

Computing a shortest path (earliest arrival time) for car routing using the road network alone takes 0.3 seconds on average to compute. The worst case running time of computation is obtained by computing a path from the bottom left corner to the upper right corner. Using the same two initial points with the transit network included, the worst case scenario takes 0.7 seconds on average to compute. Fig. 6 shows the computed route for both examples.

As mentioned in section 4, the running time of Dijkstra’s algorithm depends on the size of the network, in terms of both the number of nodes and arcs. While considering the transit network does not increase the number of nodes considerably, however the number of arcs is increased by approximately 30%. The size of the solution space which needs to be explored is increased which affects the time spent querying the network.

Another point worth mentioning is that we are able to stop the execution of Dijkstra’s algorithm once the target node is fixed (de-queued), to avoid scanning the entire network. This does not affect the correctness of the solution, because once a node is fixed, then the shortest path form the source to the target has been found, this is a property of the algorithm. This means that the closer the two points are to each other, the smaller is the size of the portion of the graph which needs to be processed. That is because Dijkstra’s algorithm works by scanning the network in a breadth-first approach by increasing the radius around the source node, which is the size of the so-far processed portion of the graph. Therefore, the worst case running time is obtained by calculating the shortest path between the two farthest nodes in the graph.

In all forthcoming figures, the green marker marks the departure point, and the red marker marks the destination point. Every route consists of multiple legs, where a blue leg stands for the foot/car portion of the route, a red leg stands for the public transportation portion of the route, and a cyan leg stands for the carpool portion of the route.

Fig. 7 shows an example of a solution that considers carpooling as a transportation method. In this solution, the trip starts at 17:00, where the commuter starts the trip by walking for 2 minutes to the nearest carpool pick up point for a 17 minutes’ drive to reach the train station at 17:22. The commuter then gets on the train at 17:27 to reach the next stop at 17:38, to proceed walking for two minutes to the next bus stop to get on the bus at 17:42 to reach the final destination at 17:51 after a 3 minute walk. The suggested solution results in a total travel time of 51 minutes and 3 different transportation changes. The next example shows the alternative solution in the case when carpooling is not considered as a transportation method or as an alternative solution to what happens if the user misses the carpool. Fig. 8 shows the suggested solution. The alternative solution results in a total of 5 minutes walking, 48 minutes of transportation, with 3 different transportation changes, to reach the final destination at 18:21 with total travel time of 1 hour and 21 minutes. Comparing the two previous solutions, considering carpooling as an alternative transportation method reduces the total travel time by 30 minutes.

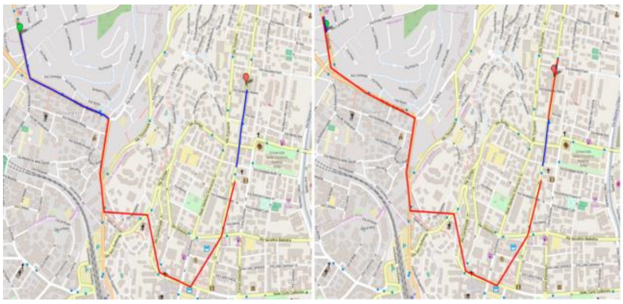

Figure 9. The two solutions produced when changing the weights of the objective function.

Finally we demonstrate how changing the objective functions' weights affects the solution produced. In the first solution, we penalize the *WalkingTime* by 1.5, the *TransportationTime* by 1, and the *BusChanges* by 20. The used weights should result in a solution which does not restrict walking for relatively short distances in order to reduce waiting time for transportation relative to the departure time, and not favoring the use of public transportation when the distance from the current point to the final destination is within walking distance. Fig. 11 (left picture) shows such a solution. The solution suggests walking to one of the bus stops along the route to reduce the waiting time at the closest stop from the original departure point, and getting off the bus at the closest station within walking distance to the final destination.

In the second solution, we penalize the *WalkingTime* by 30, the *TransportationTime* by 1 and the *BusChanges* by 20. The used weights favor using public transportation over walking, even for what could be considered in the range of a walking distance. Fig. 9 (right picture) shows such a solution, which is close to the first solution; however, the trip starts by walking to the closest bus stop to the departure point, in the cost of increasing the waiting time. We can also notice that after leaving the bus at the mutual point between the two solutions, a bus change occurs to reach the final destination, thus reducing the total walking time between the two solutions, on the cost of doing one final bus change to reach the final destination.

VI. Conclusion

In this work we introduced multiple approaches taken from the literature to model transportation networks and introduced an approach to model carpooling services and integrate it with traditional transportation methods. Dijkstra's algorithm is used as a route planning algorithm with a linear weighted objective function. The implemented algorithm results in reasonable query times, and the algorithm suggests reasonable solutions in terms of user convenience. Using carpooling as an alternative transportation method could reduce travel time considerably, where the travel time on foot is reduced when travelling from remote areas.

In terms of solution quality, the algorithm used needs to expand the objective function to improve the matched carpooling route based on factors such as the driver's reputation, and possibly other factors. Moreover, the algorithm shows sensitivity to the provided departure time, where a different solution (possibly better) could be suggested when postponing or delaying the departure time of the trip.

Acknowledgment

This work is co-funded by the European Union's Horizon 2020 Research and Innovation Program under grant agreement no. 636427 "SocialCar: open social transport network for urban approach to carpooling".

References

[1] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes, "In transit to constant time shortest-path queries in road networks," in *Proc. Meeting on Algorithm Engineering & Experiments*. *Society for Industrial and Applied Mathematics*, 2007, pp. 46-59.

[2] M. Braun, "Multi-Model route planning with transfer patterns," Master's thesis, Albert-Ludwigs-Universität Freiburg, 2012.

[3] M. Brodesser, "Multi-Modal route planning," Master's thesis, Albert-Ludwigs-Universität Freiburg, 2013.

[4] D. Delling, J. Dibbelt, T. Pajor, D. Wagner, and R. F. Werneck, "Computing and evaluating multimodal journeys," Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany. 2012

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.

[6] R. Gallotti and M. Barthelemy, "The multilayer temporal network of public transport in Great Britain," *Scientific Data*, 2015.

[7] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and simpler hierarchical routing in road networks," in *Proc. International Workshop on Experimental and Efficient Algorithms*, Berlin, Germany, 2008, pp. 319-333.

[8] R. Geisberger, P. Sanders, P. Schultes, and D. Vetter, "Exact routing in large road networks using contraction hierarchies," *Transportation Science*, vol. 46, no. 3, pp. 388-404. 2012.

[9] P. Holme and J. Saramäki, "Temporal networks," Physics Reports, vol. 519, no. 3, pp. 97-125. 2012.

[10] E. Köhler, K. Langkau, and M. Skutella, "Time-expanded graphs for flow-dependent transit times," in *Proc. European Symposium on Algorithms*, Berlin, Germany, 2002, pp. 599-611.

[11] R. H. Möhring, H. Schilling, B. Schütz, D. Wagner, and T. Willhalm, "Partitioning graphs to speedup dijkstra's algorithm," *Journal of Experimental Algorithmics*, vol. 11, pp. 2-8, 2007.

[12] T. Pajor, "Multi-modal route planning," Diploma thesis, Institut für Theoretische Informatik ,Universität Karlsruhe, Karlsruhe, Germany, 2009.

[13] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: The case for A," *International Journal of Geographical Information Science*, vol. 23, no. 4, pp. 531-543, 2009.