

# Routing with public transport and ride-sharing

Sacha Varone<sup>1</sup>, Kamel Aissat<sup>2</sup>

<sup>1</sup> University of Applied Sciences and Arts Western Switzerland (HES-SO), HEG Genève, Switzerland  
`sacha.varone@hesge.ch`

<sup>2</sup> LORIA, University of Lorraine, France  
`kamel.aissat@loria.fr`

**Keywords:** *Multi modal, ride-sharing, public transportation, real time.*

## 1 Background

Mobility within a region can be done using different transportation modes, either public or private, which might be a function of the preferences or of the possibilities of the commuter. We propose an algorithmic approach to the real time multi-modal earliest-arrival problem (EAP) in urban transit network, using public transportation and ride-sharing, so that the trip duration is minimized. We start from using public transportation, included walking sub-paths and then, considering (quasi) real time ride-sharing opportunities, try to transfer sub-paths from public modality to ride-sharing modality. For that purpose, we consider riders, who provide requests to go from one location to another one, and drivers who offer ride-sharing. Riders' goal is to find a quickest trip in a dynamic context that effectively combines the use of several modes of transportation.

The search for a quickest or shortest path has been well studied in the literature, and has now very efficient algorithms to solve it, in the order of a millisecond for a continental trip: Bast *et al.* [1] survey recent advances in algorithms for route planning. Computation of shortest paths are key components in route planning, but not sufficient to deal with multi-modal problems, since they involve multi-criteria paths, transition or waiting times, etc. which is usually not taken into account in shortest paths on pure road networks. For example, Ambrosino and Sciomachen [2] include several features in their objective function, so that they focus on the modal change node, and propose a two step algorithm for computing multi-modal routes. An exact algorithm considering arriving and leaving time-windows has been proposed in Liu *et al.* [3]. Multi-criteria search by computing the Pareto set is done in [4].

In public transportation problems, solving earliest-arrival problems (EAP) knowing the departure and arrival station, departure time and timetable information is reviewed in [5, 6]. Timetabling information and EAP solving is nowadays often available on-line for users. Real time journey using ride-sharing opportunities faces the commuting point problem: it has to be decided where the pick-up and drop-off locations have to be located. Bit-Monnot *et al.*[7] define this as the 2 synchronization points shortest path problem (2SPSPP). The time complexity of their approach prevents its use in real-time ride-sharing. In our approach, we first find a quickest path using public transportation and consider its different transit stops. We then only allow pick-up and drop-off around those stops to reduce the search space. We also consider the *best offer selection problem*, i.e. for a given rider, we select the best driver that improves the rider's itinerary, under driver's detour time and driver's waiting time constraints.

## 2 Approach

The following situation is considered: a rider  $u$  wishes to go from an origin point  $O_u$  to a destination point  $D_u$ . He might use either public transportation, ride-sharing or a combination of both. We assume that public transportation timetabling is known<sup>12</sup>, as well as the couple origin-destination of car-sharing drivers. Let's  $P = O_u, x_1, \dots, x_{nbs}, D_u$  be the itinerary using public transport, where  $x_i$  are transshipment stops.

### Initialization

An initialization step gets the public transportation path  $P$ , with its arrival time and departure time on each of its commute node between position  $O_u$  and destination  $D_u$ . Next, we compute possible driving substitution paths along  $P$ . A list of potential drivers is then initialized.

### Closeness estimation

We define an estimate on how close is a substitution driving path  $OD$  to a public transportation sub-path of  $P$ . This estimated distance is defined as the minimal sum of the estimated distance from a vertex in  $OD$  to a vertex in  $P$ , and backward from  $P$  to  $OD$ . Four different points are used, so that only non-trivial substitution are allowed (i.e. no substitution of a single vertex). For that purpose, we estimate the distance between two points given by their latitude/longitude with the Haversine formula. Using this closeness estimation, we restrict the list of potential drivers to those whose  $OD$  is close enough to  $P$ . This is based on the maximum detour allowed by the driver, and the maximal waiting time constraint for the driver at a pick up location. This can be seen as a projection view of a path to another one, while including the waiting time.

### Shortest paths computation

Once a driver  $k$  is classified as a potential driver, we compute a set of shortest paths (i.e. not estimated but exact) in order to check the constraints of *time detour* and *waiting time*. This serves as a basis to substitute some part of the public transportation path  $P$  with a ride-sharing modality. We construct shortest driving paths from the position of the driver to each of the transshipment stops, with the exception of the last one  $D_u$ . The goal is to find a driving way towards possible pick up stops. We then compute driving paths from next transshipment stops, with the exception of the first one  $O_u$ , to the driver's destination  $D_k$ . Such a path will be used once a driver has taken off a rider, and the route towards its destination  $D_k$  has been found.

### Substitution process

We now have the three main components in order to eventually find the best substitution sub-paths of  $P$  for the rider.

- The set of shortest paths from drivers to stops in  $P$
- The set of possible ride-sharing along  $P$  (i.e substitution sub-paths from one stop to another one)
- The set of shortest paths from stops in  $P$  to drivers' final destination.

Only feasible substitution sub-paths that do not increase the arrival time of the rider to his destination are considered. For that purpose we introduce the notion of *reasonable substitution sub-path* with the following definition:

We say that an arc  $(x_i, x_j) \in P$  form a reasonable substitution sub-path for the driver  $k$  and the rider  $u$  if and only if the maximal waiting time  $w_{\max}^k$  constraint for the driver  $k$  at a pick up location, the maximum detour constraint for the driver  $k$  and the latest arrival time constraints for the rider  $u$  are satisfied.

---

<sup>1</sup>Switzerland: <http://transport.opendata.ch>

<sup>2</sup>France: <http://www.navitia.io>

The objective function that defines a best substitution sub-path is based on time-savings: among all *reasonable substitution sub-path*, we select the *best substitution sub-path*  $(x_i, x_j)$  that generates for the rider the most positive time-savings, if he uses ride-sharing (with driver  $k$ ) rather than public transport from the pick up stop  $x_i$  to the drop off stop  $x_j$ .

#### All together

Finally, in order to find a best reasonable substitution sub-path among all drivers, it remains to scan each driver  $k$  in the potential driver's list and select the driver  $k^*$  that generates the best reasonable substitution sub-path  $(x^*, y^*)$ . It is to be noted that from  $y^*$ , the remaining of rider's route using public transportation has to be recomputed to take into account his new arrival time at  $y^*$ . Therefore, the path  $P$  is updated, using  $y^*$  as the new origin. The new starting time is the sum of the eventual waiting time at the pick up location for the rider, its arrival time at  $y^*$  and the modality change time. The driver's route is then  $O_k \rightsquigarrow x^* \rightsquigarrow y^* \rightsquigarrow D_k$ .

**Complexity** A complexity of the whole process in a worst case analysis shows that it has a running time proportional to the number of stops, the cardinality of the set of potential drivers, and the time required to recover the public transportation path using an API (Application Programming Interface). Of course, it also depends on the running time of shortest paths algorithms, which could be Dijkstra-like or more efficient ones like hub labelling or contraction hierarchies.

### 3 Numerical results and conclusion

The methodology to test our approach is based on simulations, since we are not aware of benchmarks that fit the problem we solve. We first chose  $k$  clusters corresponding to  $k = 4$  cities in Switzerland (Fribourg, Lausanne, Bern, Neuchâtel). We then create  $L_r = 10$  requests between for each pair of clusters, starting randomly between 7:00 and 8:00 AM on a weekday. For each request, we retrieve public transportation from available API. We then create  $L_d = 5$  driving trips close enough from the stops given by public transportations segments, using geographical maps from OpenStreetMap<sup>3</sup>, as well as routing process available through the open source tool OsmSharp<sup>4</sup>. We then apply our algorithms to measure the gain in terms of arrival time, as well as the running time of our algorithms.

The results of our simulation give some insight about the efficiency of our approach. First, this shows that a true application running in real-time is possible since the whole process running on a personal computer requires only a few seconds. Second, there is a significant gain for the rider in terms of arrival time. Nevertheless the objective function, although appropriate for certain types of users, might be redefined in several terms: the total monetary cost for the rider, the deviation from the origin-destination trip for the driver, the number of transshipment stops, etc. Our approach can integrate those features as a weighted sum in the objective function.

The problem described in this paper is a first step in designing new multi modal transportation process. Multi modalities usually includes pedestrian, cycling, private car or public bus or train transportations. Our approach allows to also include ride-sharing.

---

<sup>3</sup><http://www.openstreetmap.org>

<sup>4</sup><http://www.osmsharp.com>

## References

- [1] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. Werneck, “Route planning in transportation networks,” MSR-TR-2014-4 8, Microsoft Research, 1 2014.
- [2] D. Ambrosino and A. Sciomachen, “An algorithmic framework for computing shortest routes in urban multimodal networks with different criteria,” *Procedia - Social and Behavioral Sciences*, vol. 108, no. 0, pp. 139 – 152, 2014. Operational Research for Development, Sustainability and Local Economies.
- [3] L. Liu, J. Yang, H. Mu, X. Li, and F. Wu, “Exact algorithms for multi-criteria multi-modal shortest path with transfer delaying and arriving time-window in urban transit network,” *Applied Mathematical Modelling*, vol. 38, no. 9-10, pp. 2613–2629, 2014.
- [4] D. Delling, J. Dibbelt, T. Pajor, D. Wagner, and R. Werneck, “Computing multimodal journeys in practice,” in *Experimental Algorithms* (V. Bonifaci, C. Demetrescu, and A. Marchetti-Spaccamela, eds.), vol. 7933 of *Lecture Notes in Computer Science*, pp. 260–271, Springer Berlin Heidelberg, 2013.
- [5] M. Müller-Hannemann, F. Schulz, D. Wagner, and C. Zaroliagis, “Timetable information: Models and algorithms,” in *Algorithmic Methods for Railway Optimization* (F. Geraets, L. Kroon, A. Schoebel, D. Wagner, and C. Zaroliagis, eds.), vol. 4359 of *Lecture Notes in Computer Science*, pp. 67–90, Springer Berlin Heidelberg, 2007.
- [6] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis, “Efficient models for timetable information in public transportation systems,” *J. Exp. Algorithmics*, vol. 12, pp. 2.4:1–2.4:39, June 2008.
- [7] A. Bit-Monnot, C. Artigues, M.-J. Huguet, and M.-O. Killijian, “Carpooling : the 2 synchronization points shortest paths problem,” in *13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*, vol. 13328, (Sophia Antipolis, France), p. 12, Sept. 2013.