Solving transit first mile/last mile problem with ridesharing

Pramesh Kumar
University of Minnesota - Twin Cities, kumar372@umn.edu

Alireza Khani
University of Minnesota, akhani@umn.edu, http://www.cege.umn.edu/directory/faculty-directory/khani.html

---

1. Introduction

Due to limited transit network coverage, suburban commuters often face the transit first mile/last mile (FMLM) problem. To deal with this, they either drive to a park-and-ride location to take transit, use carpooling or drive directly to the destination to avoid inconvenience. Ridesharing, an emerging mode of transportation, can solve the transit first mile/last mile problem. In this setup, a driver can drive a ride-seeker to a transit station, from where the rider can take transit to his/her respective destination. We develop a transit-based ridesharing matching algorithm to solve this problem. The method uses the schedule-based transit shortest path to generate feasible matches and then uses a matching optimization program to find the optimal match between riders and drivers. The proposed method not only assigns an optimal driver to the rider but also assigns an optimal transit stop and a vehicle trip departing from that stop for rest of the riders itinerary. We also introduce the application of space-time prism (STP) in the context of ridesharing and utilize constrained rider and driver movement in the network depending on the available time budget and current location. We use simulated data obtained from activity-based travel demand model of Twin Cities, MN to show that the transit-based ridesharing can solve the FMLM problem and save a significant number of vehicle-hours spent in the system. An algorithm to solve this problem dynamically using a rolling horizon approach is also presented.

2. Problem Statement

In this study, we consider a transit-based ridesharing program which receives a set of requests P. These requests ($P = R \cup D$) can be partitioned into a set of riders R, who are looking for a ride and have first mile/last mile problem, and a set of drivers D, who are willing to give a ride to the ride

seekers to a transit station. Each request $p \in P$ is defined by a tuple $\{OR(p), DS(p), \tau^{ed}(p), \tau^{la}(p)\}$, indicating its origin $OR(p)$, destination $DS(p)$, earliest departure time from origin $\tau^{ed}(p)$, and latest arrival time at destination $\tau^{la}(p)$. Let t_{ij} and $\hat{t}_{ij}$ be driving and transit time from location i to location j respectively. Let $\mathcal{G}_{\mathcal{R}}(\mathcal{N}_{\mathcal{R}}, \mathcal{A}_{\mathcal{R}})$ be a directed graph representing road network, where $\mathcal{N}_{\mathcal{R}}$ is the set of nodes and $\mathcal{A}_{\mathcal{R}}$ is the set of links connecting these nodes. Similarly, $\mathcal{G}_{\mathcal{T}}(\mathcal{N}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}})$ represents a directed graph representing a transit network, where $\mathcal{N}_{\mathcal{T}}$ represents a set of nodes in transit network and $\mathcal{A}_{\mathcal{T}}$ represents a set of arcs in transit network. A schedule-based transit network capture complexities of transit service such as precise waiting time, in-vehicle time, and transfer time. The transit-based ridesharing matching problem described in this paper can be defined as follows: Given a set of requests P, a road network $\mathcal{G}_{\mathcal{R}}$, and a transit network $\mathcal{G}_{\mathcal{T}}$, find an optimal match between a driver and rider to share a ride which optimizes a given objective, an optimal drop off transit stop/station location $s \in S$ for the rider, and an optimal transit itinerary $\mathcal{I}$ for the rider. A match $(r, d, s, \mathcal{I})$ between a rider r and driver d, drop off stop location s, transit trip itinerary $\mathcal{I}$ is feasible if d should depart from $OR(d)$ after $\tau^{ed}(d)$, d should reach at $OR(r)$ after $\tau^{ed}(r)$, r should reach at $DS(r)$ before $\tau^{la}(r)$, and d should reach at $DS(d)$ before $\tau^{la}(d)$.

3. Solving transit-based ridesharing matching problem

The number of possible solution grows exponentially with the number of possible trips, transit stations, and transfer options. To reduce the size of problem, we consider *Space-time prism (STP)* for a rider or driver which is defined as the envelope of all possible space-time paths between known locations and times. A prism without any stationary active anchor consists of a time-forward cone $FC(\tau)$ pointed on the first anchor $OR(p)$ and a time-backward cone, $BC(\tau)$ pointed on the second anchor $DS(p)$.

$$FC_p(\tau) = \{n \mid \tau \geq \tau^{ed}(p) + t_{OR(p)n}, \tau \leq \tau^{la}(p)\} \quad \forall p \in P \tag{1}$$

$$BC_p(\tau) = \{n \mid \tau \leq \tau^{la}(p) - t_{nDS(p)}, \tau \geq \tau^{ed}(p)\} \quad \forall p \in P \tag{2}$$

The solution to transit-based ridesharing matching problem follows a two-step procedure. The first step finds a set of feasible matches between drivers and riders and in the second step, we use matching optimization to assign riders to drivers:

1. An algorithm to find feasible matches: The proposed method leverages *schedule-based transit shortest path* (SBTSP) algorithm for finding feasible matches by incorporating space-time constraints. The algorithm starts with the initialization of a set of potential matches as $\mathcal{M} = \phi$. For every rider $r \in R$, we run a backward shortest path from $DS(r)$. We initialize a scan eligible list (SEL) and maintain two different types of labels–time labels γ and generalized cost labels γ^{gc}. The time labels are used to check if the time constraints are satisfied for both riders and drivers

while the generalized cost labels γ^{gc} are used to maintain a minimum generalized cost for a rider which is calculated as the weighted sum of the cost of traversing different types of link in the transit network. We check the Bellman's principle of optimality and restrict the network search by using driving time as a lower bound on transit time from $OR(r)$ to node j as a space-time constraints. While updating labels of each node $j \in \mathcal{N}_{\mathcal{T}}$, each driver $d \in D$ is checked for his/her compatibility with given rider r. Then we check if d can reach $OR(r)$ by $\tau^{ed}(r)$, and if d can reach $DS(d)$ before $\tau^{la}(d)$. If all these conditions are satisfied, then we add (r, d, j, ξ) to $\mathcal{M}$.

2. Optimal assignment of drivers to riders The second and last step of the procedure is to find an optimal match between riders and drivers. This is formalized as an Integer Linear Program (ILP). The goal of this optimization is to find an optimal match out of feasible matches which optimize a given objective. Let $\mathcal{M}_r = \{m \in \mathcal{M} | r \in m\}$ and $\mathcal{M}_d = \{m \in \mathcal{M} | d \in m\}$ be the set of all feasible matches of r and d present in $\mathcal{M}$ respectively. The set of feasible matches can be seen as edges between riders and drivers in a *bipartite graph*, where a rider can appear more than once with different drop off location and itinerary. Let us assume a binary variable ϵ_k, which takes the value 1, if edge $k \in \mathcal{M}$ is selected, and 0 otherwise. Let t_k^{vhrs} be the vehicle-hrs savings associated with edge $k \in \mathcal{M}$. t_k^{vhrs} savings can be computed as:

$$t_k^{vhrs} = t_{OR(r)DS(r)} + t_{OR(d)DS(d)} - t_k^{drive} \text{ for } (r, d) \in k \quad \forall k \in \mathcal{M} \tag{3}$$

We consider two different objectives for this problem: maximize total number of matches $Z_1 = \sum_{k \in \mathcal{M}} \epsilon_k$ and maximize veh-hrs savings, $Z_2 = \sum_{k \in \mathcal{M}} t_k^{vhrs} \epsilon_k$. The constraints make sure that each rider is assigned to at most one driver and each driver is assigned to at most one rider.

$$\begin{aligned} \underset{\epsilon}{\text{minimize}} \quad & Z \\ \text{subject to} \quad & \sum_{k \in \mathcal{M}_r} \epsilon_k \leq 1 \quad \forall r \in R \\ & \sum_{k \in \mathcal{M}_d} \epsilon_k \leq 1 \quad \forall d \in D \\ & \epsilon_k = \{0, 1\} \quad \forall k \in \mathcal{M} \end{aligned} \tag{4}$$

4. Results and findings

We created a simulation environment using Twin Cities activity-based travel demand data to assess the benefits of such program. We found that such a ridesharing program can save a significant amount of veh-hrs spent in the system. The number of matches found using both objectives were found to be close to each other. However, maximizing veh-hrs savings can save a lot more veh-hrs than maximizing the total number of matches. We also closely observed different trip components of driver and rider and found that driving from rider's origin to drop-off station is a significant

Table 1 Static simulation results

Sample number	Number of participants	Number of feasible matches found	Number of riders and drivers in feasible matches	Objective (min)	Optimal number of matches found	Total vehicle-hrs savings	Average driver detour time (min)	Shared time between driver and rider (min)	Average transit time (min)	Average walking time (min)	Average waiting time (min)
1	1,120 D, 1,059 R	22,397	364 R, 604 D	Z_1	319	107.4	10.12	29.54	6.06	10.75	0.5
				Z_2	316	122.47	7.65	29.9	6.41	10.85	0.63
2	1,035 D, 1,144 R	10,415	218 R, 344 D	Z_1	192	55.97	11.58	27.86	6.44	10.1	0.59
				Z_2	192	67.03	8.82	28.1	7.49	10.09	0.66
3	1,096 D, 1,083 R	9,936	178 R, 299 D	Z_1	155	48.81	11.39	29.54	6.59	11	0.61
				Z_2	154	57.15	8.48	29.44	7.64	10.85	0.64
4	1,092 D, 1,087 R	10,021	182 R, 334 D	Z_1	166	48.14	11.78	28.36	5.43	10.24	0.58
				Z_2	166	55.97	9.08	28.24	5.76	10.01	0.54
5	1,099 D, 1,080 R	8,924	187 R, 346 D	Z_1	168	47.17	11.95	28.12	5.29	10.57	0.463
				Z_2	166	58.7	8.29	28.52	6.32	10.46	0.4
Average	1,088 D, 1,097 R	12,338.6	263 R, 386 D	Z_1	200	61.5	11.36	28.68	5.29	10.532	0.5486
				Z_2	198	71.6	9.2	34.4	6.32	10.452	0.574

component of both rider and driver's itinerary. This is due to less time flexibility in riders time schedule as transit travel time is generally higher than the corresponding driving time. We also performed a sensitivity analysis on the participation rate, time flexibility, and driver-rider ratio. With an increase in time flexibility and participation rate, the runtime also increases producing more number of feasible matches. Less driver-rider ratio produces more number of matches. Table 1 shows results from a static simulation performed.

References

[1] Khani, A., Hickman, M. and Noh, H. 2014. Trip-Based Path Algorithms Using the Transit Network Hierarchy, Networks and Spatial Economics 15(3), 635.