

# UC Irvine

## Working Paper Series

### Title

A Real-Time Algorithm to Solve the Peer-to-Peer Ride-Matching Problem in a Flexible Ridesharing System

### Permalink

<https://escholarship.org/uc/item/8d2186s7>

### Authors

Masoud, Neda  
Jayakrishnan, R., PhD

### Publication Date

2015-07-01

**A Real-Time Algorithm to Solve the Peer-to-Peer Ride-Matching  
Problem in a Flexible Ridesharing System**

UCI-ITS-WP-15-1

**Neda Masoud, Ph.D. candidate  
R. Jayakrishnan, Ph.D.**

**Institute of Transportation Studies  
University of California, Irvine; Irvine, CA 92697-3600, U.S.A.  
nmasoud@uci.edu, rjayakri@uci.edu**

**July, 2015**

**Institute of Transportation Studies  
University of California, Irvine  
Irvine, CA 92697-3600, U.S.A.  
<http://www.its.uci.edu>**

**ABSTRACT**

Real-time peer-to-peer ridesharing is a promising mode of transportation that has gained popularity during the recent years, thanks to the wide-spread use of smart phones, mobile application development platforms, and online payment systems. An assignment of drivers to riders, known as the ride-matching problem, is the central component of a peer-to-peer ridesharing system. In this paper, we discuss the features of a flexible ridesharing system, and propose an algorithm to optimally solve the ride-matching problem in a flexible ridesharing system in real-time. We generate random instances of the problem, and perform sensitivity analysis over some of the important parameters in a ridesharing system. Finally, we introduce the concept of peer-to-peer ride exchange, and show how it affects the performance of a ridesharing system.

## 1 INTRODUCTION

Congestion in urban transportation networks is one of the common problems faced by many countries around the world. In US alone, 3.7 billion hours and 2.3 billion gallons of gas are wasted in traffic jams every year (U.S. Department of Transportation, 2006). In addition to having a direct impact on travel time and fuel consumption, congestion imposes indirect costs by increasing travel time uncertainty, as well as emission levels which adversely affect population health and ecosystems.

Managing congestion by expanding the infrastructure is costly and damaging to the environment. An alternative way would be to make more efficient use of existing capacities on the roads. Transit systems are a conventional example for this principle.

Transit systems in urban networks mainly include buses and rail services. They typically carry multiple passengers, and therefore can help reduce vehicle miles traveled (VMT) and ease congestion in urban networks. One drawback of transit services is that they operate on fixed routes and schedules. This limits transit coverage, both geographically and temporally. Moreover, due to government regulations on fares, transit services usually fail to act as financially independent entities, and are in need of subsidies.

Para-transit services were originally introduced to run as supplementary services alongside transit, and as a mean to increase the flexibility of public transport. These services are demand-responsive, and usually serve multiple passengers at a time, based on spatiotemporal proximity of the requests they receive. Routes and schedules are fairly flexible according to demand. Para-transit services include all shuttle-like services that serve customers such as airport travelers, employee or student commuters, or the elderly and disabled. Since the passage of Americans with Disabilities Act of 1990, however, the term para-transit has been used more commonly to refer to the services provided to persons with disabilities, or to the elderly. In this paper, we reserve the term para-transit to refer to such services.

Although para-transit services can be beneficiary to demographics they target, these demographics are fairly limited. In addition, most services that fall under this category are offered by non-profit organizations that are supported by federal funding. Due to the limit on the amount of subsidies, it is not possible to extend these services to the general public.

These limitations along with the desire for a more comfortable (and possibly quicker) ride have resulted in a much higher demand for private sector alternatives, such as taxis.

Taxis are a private form of demand-responsive transportation alternative. They provide door-to-door transportation, but at a higher cost that not everyone can afford. Shared taxi/limousine/shuttle services provide an opportunity for customers to share their trips, cutting the cost of the journey, and potentially reducing the total miles traveled in the system (which translates into lower environmental impact). Over the years, different shared-use services have been designed. Flexible route transit systems (Quadrioglio et al. 2008; Li and Quadrioglio 2010; Qiu et al. 2014), and the High-Coverage Point-to-Point Transit system (Cortés and Jayakrishnan, 2002) are a few examples.

In all alternative modes of transportation discussed above, drivers work as system employees. Some of the discussed transportation alternatives are more flexible than others in terms of routing and scheduling, and some have the potential to benefit the environment and customers by allowing them to share (parts of) their trips. In the next family of transportation alternatives we discuss, drivers are not employed by the agencies.

Masoud, Jayakrishnan

A different family of transportation alternatives which has attracted considerable attention during the recent years is founded based on the principle of shared-use mobility (Shaheen and Chan, 2015). These alternatives try to reduce the cost of flexible transportation by reducing (or completely eliminating) the capital and operating costs. Informal carpooling is one of the first and common forms of trip sharing, in which already-familiar individuals use the same vehicle for their travel, which typically involves the same origin and destination points. Examples include parents who take turns in taking their children to school, colleagues who travel to work together, etc. In this form of carpooling, vehicles belong to individuals, and the driver of the vehicle has personal interest in the trip, regardless of whether additional passengers are present in the car. The incentives for informal carpooling could be saving time through carpool lanes, or not having to drive one's own vehicle every single day, if participants take turns in driving. This form of carpooling is usually pre-arranged, and happens among individuals who share commonalities beyond the time and location of their trips.

Transportation Network Companies (TNC), such as Uber and Lyft, are among the more recent faces of shared-use mobility alternatives. TNCs use private vehicle owners and their personal vehicles to provide flexible and on-demand transportation services. These individuals work for the company as private contractors instead of employees. This substantially reduces the cost of capital and human resources, while generating revenue for both the company and the drivers. Essentially, the basic services provided by TNCs act as a lower-cost alternative to taxi services, but not only do they not address the problem of increasing travel demand and congestion, but they add to it. In terms of sustainability, TNCs bear the same cost to the environment as taxi companies. Recently, two of the more prominent TNCs, Uber and Lyft, have introduced sharing services Uberpool and Lyft Line. Such services can certainly reduce the cost and environmental impacts of the base services.

Peer-to-peer (P2P) ridesharing systems aim at capturing the benefits of TNCs while alleviating their adverse impact on the environment. These systems are founded on the principle of sharing economy. Sharing economy, also known as collaborative consumption, is a fairly old concept that focuses on the benefits obtained from sharing resources (products or services) that would otherwise go unused. This economic model has gained more popularity in the recent years, giving birth to many P2P services in different fields (for examples refer to Böckmann, 2013). The advent of the internet has extended the domains of sharing economy to global populations, and has highlighted its benefits. Moreover, new computer platforms allow easy and quick development of companion mobile applications that facilitate sharing economy.

Similar to TNCs, drivers in a P2P ridesharing system use their own vehicles to carry passengers, and do not work as agency employees. Contrary to the TNC operations, drivers in a P2P ridesharing system are making trips to perform activities of self-interest (as in the case of informal carpooling), i.e., they do not roam around the city only to pick up and deliver passengers. This setting can lead to services that are more environmentally-conscience and cost-efficient compared to the sharing services offered by TNCs.

The overwhelming success and well-reception of TNCs by the public suggest a bright future for more environmentally-friendly ridesharing systems. Since founded in 2009, Uber has managed to expand its operations in 58 countries and more than 300 cities worldwide. According to business insider, in 2014 Uber had more than 160,000 registered driver-partners in the US, and the number of new drivers subscribing to Uber each month increased exponentially, approaching 40,000 (Business Insider, 2015). The company is expected to hit the \$10 billion revenue by the end of 2015 (Business Insider, 2014).

Table 1 summarizes features of the transportation alternatives discussed above. This table suggests that ridesharing systems have the potential to outperform other alternatives in terms of cost, flexibility, and impact on the congestion and the environment, if they succeed to operate as financially independent systems.

Although P2P ridesharing systems emerged in the US in 1990's, they failed to survive as stand-alone entities. The higher cost of car ownership, the possibility of making safe on-the-spot electronic payments, and the prevalence of GPS-enabled cellphones which allow easy tracking of participants in a rideshare transaction are among the many factors that have influenced a promising return of ridesharing systems in recent years.

A look at the literature on P2P ridesharing suggests that low participation rate has been one of the main obstacles of ridesharing systems since the early days (Haselkorn et al., 1995; Levofsky and Greenberg, 2001; O'Sullivan, 2011; and RTrip, 2009). The operational advantages of ridesharing systems over TNCs could also serve as a potential obstacle in their successful implementation.

Drivers in a ridesharing system are available only during certain time windows, and have limited budget and certain origin and destination locations that is of personal interest to them. Therefore, the spatiotemporal coverage of drivers in a ridesharing system can be limited. Unsuccessful attempts to participate in a ridesharing system, either as a rider or a driver, may discourage users from returning to the system. Therefore, having an advanced ride-matching technique that maximizes the number of successful driver-rider matches is an indispensable part of a ridesharing system. Employing a good ride-matching algorithm not only improves short-term revenues by increasing the number of served riders, but as well significantly enhances customer experience, builds trust and reputation, encourages existing customers to re-use the system, and brings prospective customers to consider

We define a fully-flexible P2P ridesharing system as one that has the following four features: (1) Capable of providing multi-hop routes to riders, (2) Capable of optimally routing drivers, (3) Capable of making driver-rider matches in real-time, and (4) Equipped with a ride-matching algorithm produces the optimal (or near-optimal) match for multiple drivers and riders simultaneously. We call a ridesharing system that satisfies only the first three features, a semi-flexible ridesharing system. We discuss these features in the next section in more detail.

In this paper, we develop a real-time ride-matching algorithm that not only creates a maximal set of driver-rides in real-time, but also provides the optimal routing of all participants in a semi-flexible P2P ridesharing system. A route plan for a driver involves the exact route the driver should take, the time and location of the stops he/she needs to make to pick up/drop off riders, and the list of riders that ought to be picked up or dropped off at each stop. A rider's route plan involves the time and location the rider is picked up/dropped off by each driver.

Our formulation of the ride-matching problem perfectly and explicitly incorporates the first three components of a flexible system. Note that the semi-flexible ride-matching problem is known as a complex problem and computationally challenging to solve in real time. Adding the fourth component (multi-driver multi-rider feature) adds additional complexity to the model, and renders the optimal solution unattainable in real-time (Masoud and Jayakrishnan, 2015). Still, we incorporate the fourth component by introducing the concept of "P2P ride exchange". We argue that as more riders take advantage of such exchange opportunities, the final matching approaches the optimal matching obtained for a fully-flexible ridesharing system.

The rest of this paper is organized as follows: In the following section, we provide a review of the relevant literature that attempt modeling and/or solving ride-matching problems.

Masoud, Jayakrishnan

We follow this review by a discussion on how our proposed algorithm contributes to the literature. We then talk about the concept of P2P ride exchange, and discuss how this feature, when incorporated into the system, can improve the system performance. Finally, we solve a series of randomly generated problem instances, and study the effect of relaxing some of the assumptions made in the definition of the ridesharing system in section 3.

Table 1. Transportation alternatives differing in flexibility, cost, and environmental impacts

Transportation alternatives	Transit	Para-Transit (disabled)	Taxi	Shared Taxi/Van	Informal Carpooling	TNC	P2P Ridesharing
Drivers as Employees	✓	✓	✓	✓	✗	✗	✗
Arrangement: pre-arranged	✓	✓	✓	✓	✓	✗	✓
Arrangement: on-spot	✗	✓	✓	✓	✗	✓	✓
Multiple passengers	✓	✓	✗	✓	✓	✓	✓
Drivers peer to riders	✗	✗	✗	✗	✓	✗	✓
Financially stand-alone	✗	✓	✓	✓	✓	✓	?
Cost to customers	Low	Low	High	Medium	Low	Low	Low
Flexibility	Low	Medium	High	Medium	Medium	High	High

## 2 Literature Review

A ride-matching algorithm can be viewed as the engine of a ridesharing system. This problem can be considered as a generalization of the classic Dial-a-Ride (DAR) problem in the transportation literature. Dial-a-ride problem optimizes the pick-up and delivery of passengers in special settings that typically involve door-to-door transportation, and is commonly used in para-transit systems. In such systems, passengers in need of rides contact the agencies regarding their transportation needs, and the agencies assign vehicles to their requests. In the basic DAR problem vehicles and riders are assumed to be homogeneous. All vehicles start from the same depot in the morning, to which they return at the end of the day (Savelsbergh and Sol, 1995; Cordeau and Laporte, 2007a; Cordeau and Laporte, 2007b). Over the years, researchers have studied more complicated and practical forms of the DAR problem, by assuming time windows for riders' requests (Jaw et al., 1986; Psaraftis, 1983), multiple depots (Cordeau and Laporte, 2007a), heterogeneity among vehicles and passengers (Carnes et al., 2013; Braekers et al., 2014), and the possibility of transfers for passengers (Masson et al., 2014; Stein, 1978; Liaw et al., 1996).

The ride-matching problem is very similar to the more advanced DAR problems, in definition and therefore mathematical modeling. The main characteristic of a P2P ride-matching problem that differentiates it from DAR is the fact that drivers in a ridesharing system are also considered customers. The set of vehicles (and their drivers) is not deterministic (i.e. their availability is not generally known in advance). Once they register, drivers are available only in narrow sprites in space and time. These characteristics play an important role when it comes to developing algorithms to find optimal matchings. Since riders and drivers have very limited spatiotemporal proximity compared to the DAR problem, the standard formulation of DAR is not the most efficient way to model a ridesharing system (Masoud and Jayakrishnan, 2015).

If we were to manipulate the mathematical formulation of DAR to model the ride-matching problem, we would have to add multiple sets of constraints (and decision variables) in order to model travel time windows and pre-determined origin and destination locations specified by each driver. These additional constraint sets increase the degree of complexity of the optimization problem. On the other hand, these additional constraints shrink the feasible solution

Masoud, Jayakrishnan

space of the problem. In this paper, we exploit the latter effect to develop an efficient solution algorithm for the ride-matching problem.

In practice, ridesharing systems usually match a rider with a driver if they share the same origin and destination. From a computational point of view, this task can be easily done by making a series of comparisons in polynomial time. More sophisticated ridesharing systems ask each driver to specify their route, and assign the driver to riders with origin and destinations along that route.

Theoretically, this is a simple form of matching in a ridesharing system, if the goal is to match each driver with exactly one rider. This can be easily accomplished by modeling the system as a bipartite graph, with riders and drivers modeled as two mutually exclusive sets. Although this type of matching can be performed fast, it does not satisfy the requirements of a fully flexible system. Schaub et al. (2014) show that even a slight deviation from this simple model can make the problem intractable. For example, they show that relaxing drivers' path in the problem can make it NP-complete, under specific objective functions.

The literature of ride-matching in P2P ridesharing systems has been receiving more attention during the last couple of years. The first feature of a fully flexible ridesharing system is its ability to propose multi-hop route plans to riders. This feature allows riders to transfer between drivers, if necessary.

The general belief among transportation planners that transfers in transport systems are intolerable is not entirely accurate. The undesirability of transfers is usually attributed to the perception that transfers are time-consuming, or can make the trip unreliable. In the former case, if transfers are optimized, they can actually reduce the total travel time; and the latter case will not happen if proper measures are taken to ensure reliability of transfers. Furthermore, transfers are in fact an inseparable part of a high-quality multi-modal transport system (Vukan, 1981). Designing multi-hop ridesharing systems can facilitate integrating these systems with available public transportation, providing a seamless transport platform where ridesharing systems can serve as feeders to public transit. Masoud and Jayakrishnan (2015) show that considering transfers can considerably increase the number of matched riders. In their work, they allow riders to specify the maximum number of transfers they are willing to make, and incorporate this measure in their formulation of the problem. Agatz et al. (2009), Herbawi and Weber (2011), Ghoseiri (2013), and Masoud and Jayakrishnan (2015) consider this feature in their models.

Another desirable feature in a ridesharing system is its ability to propose optimal routes to drivers. Alongside multi-hop routes, this feature plays an important role in increasing the number of served riders (Masoud and Jayakrishnan, 2015). Most of the work in the literature consider this feature in modeling ridesharing systems (Masoud and Jayakrishnan 2015; Agatz et al. 2009, 2011; Teodorovic and DellOrco 2005; Baldacci et al. 2004; Di Febbraro et al. 2013; Herbawi and Weber 2012). Ghoseiri (2013) allows drivers to take detours from a pre-determined path. Although this approach is superior to considering fixed routes, it still imposes limits on routing of drivers.

A flexible ridesharing system should be equipped with a ride-matching problem that is capable to providing optimal (or high-quality near optimal) solutions in a short period of time. The majority of works in the literature present a mathematical model a ridesharing system as an optimization problem, but don't discuss the solution algorithms (Agatz et al. 2009; Di Febbraro et al. 2013). Others propose heuristic solutions (Ghoseiri, 2013; Herbawi and Weber, 2012, 2011; Calvo et al., 2004; Teodorovic and DellOrco, 2005). Masoud and Jayakrishnan (2015) present a formal mathematical formulation of the problem, and propose an algorithm to cut the



solution time. The final solution time, however, is appropriate for a system that matches participants on a rolling-horizon basis, and not instantaneously.

Finally, a flexible ridesharing system should be equipped with a ride-matching algorithm that can find an optimal match for a set of riders and drivers. A ridesharing system that ignores this aspect of matching typically serves riders based on a first in, first out (FIFO) order. That is, once a rider is matched, the routes of the drivers involved in the rider's route plan are fixed. These drivers can still be considered in the ride-matching problems that need to be solved later, but with fixed paths. Fixing drivers' paths comes at an opportunity cost for the riders that enter the system later (Note that even with fixing drivers' paths, it is still possible for a driver to carry multiple riders, however with lower probability). This is a missing aspect in most existing studies in the literature (Herbawi and Weber. (2011); Di Febbraro et al. (2013)).

Table 2 lists the features of a fully-flexible ridesharing system, and indicates the extent to which each study covers these features. This table shows that the work by Masoud and Jayakrishnan (2015) is the only study that includes all the features of a fully- (and semi-) flexible ridesharing system, as defined in this paper. However, as mentioned before, the solution method proposed in Masoud and Jayakrishnan (2015) is suitable for a ridesharing system implemented on a rolling-horizon basis, since the solution time of their proposed algorithm does not allow for instantaneous matchings. In this paper, we propose an algorithm that can find the optimal solution to a semi-flexible ride-matching problem in a matter of seconds, and introduce a mechanism that can help transform the optimal solution found by our algorithm to approach the optimal solution of a fully-flexible ride-matching problem. Finally, we set aside our initial assumption that riders and drivers are two mutually exclusive sets. We assume that a subset of riders have access to vehicles, and their registering in the system as riders is simply a preference. In case the system is unable to find a match for these riders, they may use their personal vehicles and contribute to the ridesharing system as drivers. Under this assumption, we study the best and worst case scenarios of the number of riders the system can serve.

Table 2. Literature on P2P ridesharing

Author	Flexible Paths	Multi-Hop	Multiple Riders	Formulation/ Solution Algorithm	Optimal Solution
Calvo et al. (2004)	✓	✗	✓	Greedy heuristic	✗
Baldacci et al. (2004)	✓	✗	✓	Optimization	✓
Teodorovic and DellOrco (2005)	✓	✗	✓	Bee colony optimization (meta-heuristic)	✗
Agatz et al. (2009)	✓	✓	✗	Optimization	✓
Agatz et al. (2011)	✓	✗	✗	Optimization	✓
Herbawi and Weber (2011)	✗	✓	✗	Genetic/Evolutionary Algorithms	✗
Herbawi and Weber (2012)	✓	✗	✓	Exact formulation + Heuristic solution	✗
Di Febbraro et al. (2013)	✓	✗	✓	Optimization	✓
Ghoseiri (2013)	✗	✓	✓	Exact Formulation + Heuristic solution	✓
Schaub et al. (2014)	✗	✗	✓	Bipartite Matching	✓
Masoud and Jayakrishnan (2015)	✓	✓	✓	Optimization + Exact algorithm	✓

### 3 RIDESHARING SYSTEM

In this study, we consider a ridesharing system in which users can postpone registering their trips to the moment when they are ready to start their trips. This type of ridesharing system that is capable of integrating requests in real-time is called a dynamic system.

Masoud, Jayakrishnan

We divide the set of participants,  $P$ , in the system into two sets of riders,  $R$ , and drivers  $D$ . A rider  $r \in R$  joins the system hoping to find a set of drivers who can collectively serve his/her ride request. A driver, on the other hand, is planning to travel to perform an activity of self-interest, and is willing to share the unused capacity of his/her vehicle with those who are in need of rides. We start the modeling process assuming that sets  $R$  and  $D$  are mutually exclusive, although we study the impact of dropping this assumption in later sections.

We identify a set of stations in the network, where participants can switch between drivers/transportation modes, and begin and end this trips (Note that the assumption that participants start and end their trips at stations is simply for the sake of simplicity, and does not affect the running time of the proposed algorithm). Stations should be located at areas with high demand for the ridesharing system, such as high density residential/recreational areas, business districts, and central transit stations.

Furthermore, we discretize the study time horizon into a set of short time intervals,  $dt$ , and index the  $i^{th}$  time interval by  $t_i$ . In this study, the length of each time interval is set to one minute. Discretizing the study time horizon into short time intervals allows for using time-dependent travel times matrices, and alongside discretizing the space dimension through introducing stations, provides the basis for the proposed ride-matching algorithm in this paper.

In order to participate in the ridesharing system, riders and drivers need to register their trips. During the registration process, a participant  $p \in P$  is asked to provide information on their origin and destination locations, denoted as  $O_p$  and  $D_p$  respectively, their travel time budget for the trip, denoted as  $T_p^{TB}$ , and a travel time window for their trip  $[T_p^{ED}, T_p^{LA}]$ , where  $T_p^{ED}$  is the earliest time interval participant  $p$  is willing to start the trip, and  $T_p^{LA}$  is the latest time interval they need to have arrived at their destination.

In addition to travel time budget and travel time window, in attempt to provide a pleasant ridesharing experience for the system users, a rider  $r \in R$  is asked to provide the maximum number of transfers they are willing to make (denoted by  $V_r$ ), and a driver  $d \in D$  is asked to provide a limit on the number of people they are willing to have as passengers at any moment in time (denoted by  $C_d$ ). In addition, all participants can indicate their preferences such as age, gender, etc on the people they travel with.

Although drivers use their personal vehicles, they are encouraged to leave the routing of their vehicles to the system. The system attempts to find a route for riders using the empty seats in the drivers' vehicles (and possibly using the available transit system), taking into consideration the travel time window and budget constraints of both riders and drivers, maximum number of transfers specified by riders, maximum vehicle capacities by drivers, and participants' personal preferences.

In a network discretized in both time and space, we define a node  $n_i$  as a tuple  $(t_i, s_i)$ , and a link  $l$  as a tuple of two nodes, i.e.  $l = (n_i, n_j) = (t_i, s_i, t_j, s_j)$ . Based on this definition, a participant who travels on link  $l = (t_i, s_i, t_j, s_j)$  leaves station  $s_i$  during the time interval  $t_i$ , arrives at station  $s_j$  during the time interval  $s_j$ . Needless to say  $(t_j - t_i + 1)dt$  is an upper bound on the travel time between stations  $s_i$  and  $s_j$ .

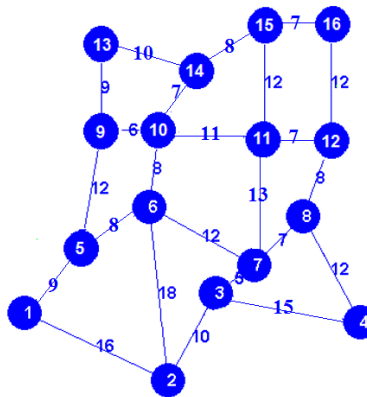
The ridesharing system serves riders on a FIFO basis. Once a rider is matched, the route plans of drivers involved in the rider's route plan are (partially) fixed. The objective of the ride-matching problem is to find the best path for a rider given the spatiotemporal and quality of service constraints on both riders and drivers.

Masoud, Jayakrishnan

All drivers in set  $D$  and all links on the time-expanded network can potentially contribute to form route plans for a rider. In the next section, we introduce the spatiotemporal ellipsoid method (STEM) to reduce the size of the input to the problem by eliminating subsets of links and drivers that cannot be used by the rider due to the spatiotemporal constraints on the participants. Next, we propose an algorithm to solve the ride-matching problem to optimality.

#### 4 ILLUSTRATIVE EXAMPLE

Throughout this paper, we use the following problem instance to illustrate the concepts and demonstrate different steps of the proposed algorithm. This example contains a network with 16 stations, one rider, and four drivers. The network is displayed in Figure 1, and the characteristics of the participants are presented in Table 1 below. It is assumed that participants have no restrictions on the people with whom they travel.



Participant	$O_p$	$D_p$	$T_p^{ED}$	$T_p^{LA}$	$T_p^{TB}$
$r_1$	5	12	2	39	37
$d_1$	7	12	3	20	17
$d_2$	14	5	3	28	25
$d_3$	3	14	4	38	34
$d_4$	1	2	10	29	19

Figure 1. Problem Instance Used For Illustrative Examples throughout the Paper

#### 5 SPATIOTEMPORAL ELLIPSOID METHOD (STEM)

The spatiotemporal ellipsoid method (STEM) generates for each rider a time-expanded feasible network in which the rider's optimal route plan could be searched. This network is constructed from the links on the original network that are reachable by drivers. STEM first uses a simple geometric tool to identify the set of stations spatially reachable by each participant (driver or rider). Next, STEM finds the time intervals during which each station can be reached, forming the set of feasible links for each participant. Finally, the rider's time-expanded feasible network is generated using the set of feasible links reachable by drivers who are within spatiotemporal proximity of the rider. The three steps of STEM are discussed in more detail in the following.

### 5.1 Approximating the Set of Reachable Stations

Spatial proximity of each participant can be determined based on his/her origin and destination stations, and travel time budget. Assume, given a participant  $p \in P$ , we draw an ellipse on the network. The focal points of this ellipse are the participant's origin and destination stations, and its traverse diameter is an upper-bound on the distance that can be travelled by the participant during his/her travel time budget. Such an ellipse is displayed in Figure 2.

For an ellipse with focal points  $f_1$  and  $f_2$ , we know that the total straight traveling distance  $f_1M + Mf_2$  for any point  $M$  on the circumference of the ellipse is equal to the traverse diameter of the ellipse, and therefore corresponds to the travel time budget of the participant. This means that any station located outside of the circumference of the ellipse cannot be reached without violating the time budget constraint of the participant. Still, note that not all stations located inside a participant's ellipse are necessarily reachable by him/her, due to the fact that actual shortest path travel times on the network are expected to be longer than straight-line distance travel times over the map.

This method substantially reduces the number of potential stations a participant can visit, simply by evaluating the coordinates of the candidate (and not all) stations in the equation of an ellipse. Storing stations in an appropriate data structure based on their geographical location makes it easy to find the set of candidate stations<sup>1</sup>. We call the part of the network confined within and on the circumference of participant  $p$ 's ellipse the "reduced network" of participant  $p$ , and denoted by  $G_p$ .

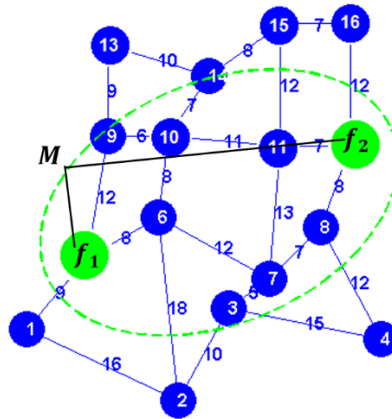


Figure 2. The reduced network  $G_r$  on the original network  $G$

Figure 3 displays the reduced networks for the participants in our example.  $r_1$  is planning to travel from station 5 to station 12.,  $G_r$  is pointed out in figure 3 as the section of the network confined inside the dashed green ellipse. The reduced networks for the four drivers in our example are also demonstrated in this figure, using solid red ellipses. The shaded area shows the intersection of the rider's reduced network with those of the drivers. We refer to the set of stations within the shaded area as the "feasible network" for rider  $r$ , denoted by  $G_r^F$ . Feasible network is defined for a rider, and contains only the section of the network that the rider can reach due to the space proximity provided by drivers in set  $D$ .

<sup>1</sup> For example, we can have one array in which stations are stored based on increasing value of their X-coordinates, and another array in which stations are stored based on the increasing value of their Y-coordinates

Defining reduced networks for participants serves three purposes. First, by studying the reduced networks of all participants, it is possible to make an early diagnosis whether a rider can be served in the system. For the rider to be served, he/she needs to have both time and space proximity by at least one driver along a path that connects the rider's origin and destination stations. Therefore, the minimum requirement for the rider to be served is to have space-proximity to at least one driver at his/her origin and destination stations. In Figure 3, for example, if driver 2 (travelling from station 14 to 5) was not in the set of drivers, the rider could not be served, due to lack of space proximity at its origin station (station 5).

In addition, studying the reduced networks makes it possible to shrink the set of potential drivers for each rider. Drivers who do not contribute to the rider's feasible network can be filtered out from the ride-matching problem. As an example, driver 4 in Figure 3 can be filtered out from the problem, reducing set  $D = \{1,2,3,4\}$  to  $D = \{1,2,3\}$ . Driver 3 does not have proximity to the rider's origin or destination, but can contribute to portions of the travel in case of a transfer. In fact, in this particular example, the only way the rider can travel from station 5 to station 12 is by being picked up by driver 2, transferring to driver 3, and finally transferring to driver 1. The exact transfer locations, and deciding whether such a route is feasible at all under the participants' time windows will be examined later. For instance, it might be the case that by the time the rider reaches station 6 or 10, driver 2 needs to have traveled beyond those stations to reach its destination on time (thereby, a spatiotemporal feasible path for the rider would not exist).

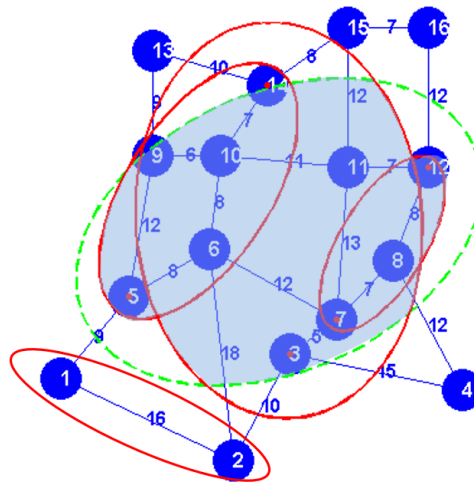


Figure 3. Reduced networks of all the participant in our example

The third benefit of forming reduced networks is that the network-dependent computations can be done much faster on  $G_p$  than  $G$ . One example of the computational savings attained by replacing  $G$  with  $G_p$  manifests itself in the second step of STEM (section 5.2), when time-proximity of stations is studied. To determine time-proximity, we need to calculate multiple shortest path travel times between the origin and destination stations of participants. Shortest path travel time matrices can be computed more efficiently on  $G_p$ . Figure 4 shows the result of calculating the shortest path travel times between the origin and destination stations of participants for problems of different sizes (i.e. various number of participants, and various network sizes). The surface on the top in this figure shows the running time of Dijkstra's algorithm on  $G$ , and the surface on the bottom shows the same running time on  $G_p$ .

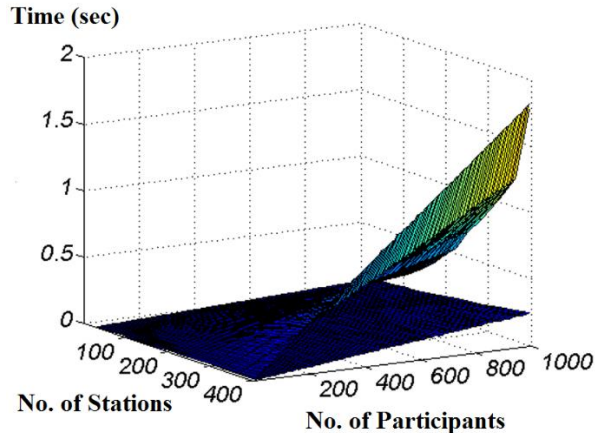


Figure 4. The surface on the top/bottom shows Dijkstra's running time on  $G/G_p$ . Results are averaged over 10 randomly generated networks for each problem

## 5.2 Constructing the Participants' Time-Expanded Reduced Networks

In the previous step, we formed the feasible network for the rider in the problem. However, not all stations in the feasible network are necessarily reachable by the rider, since we have yet to study time proximity of stations. To study time-proximity, we have to find the set of time intervals during which station  $i \in G_p$  can be reached by participant  $p$ . We denote this set as  $T_i^p$ , and use the following procedure to obtain its members.

To from the set of links for participant  $p$ , we first have to identify all the paths in  $G_p$  with travel times less than or equal to  $T_p^{TB}$ . To avoid enumerating the paths, we find the first  $k$ -shortest paths in  $G_r$  and  $G_d, \forall d \in D$ , such that the travel time on the  $k^{th}$  shortest path is less than or equal to  $T_p^{TB}$ , and the travel time on the  $(k+1)^{th}$  shortest path is higher than  $T_p^{TB}$ . After finding the  $i^{th}$  shortest path ( $i \leq k$ ) on  $G_p$ , the set  $T_i^p, \forall i \in G_p$  can be created based on the value of  $T_p^{ED}$ , and the time-dependent travel time matrices.

At this point, we know the time intervals during which each station can be reached. We can use this information to construct the "time-expanded reduced networks",  $G_p(t)$ , for participants. Figure 5 displays the time-expanded reduced network for driver 2 in our example.

Note that it is possible to have more than one origin and/or destination node in a time-expanded graph. Recall that each node is represented by as a 2-tuple of a time interval and a station. Nodes in the form of  $(t, O_p), \forall t \in T_{O_p}^p$  are all origin nodes, and nodes in form of  $(t, D_p), \forall t \in T_{D_p}^p$  are all destination nodes. We keep these nodes in origin and destination sets,  $OS(p)$  and  $DS(p)$  respectively. In Figure 5,  $OS(d_2) = \{(3,14), (4,14)\}$ , and  $DS(d_2) = \{(27,5), (28,5)\}$ .

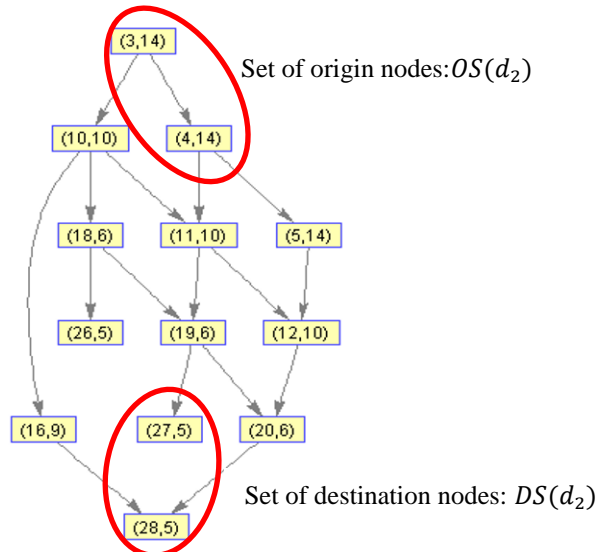


Figure 5. Time-expanded reduced network for driver 2 in Figure 3

### 5.3 Generating Rider's Time-Expanded Feasible Network

In section 7.1, we generated the **feasible network for rider  $r$ ,  $G_r^F$** , by scanning through the stations in the rider's reduced network,  $G_r$ , and checking whether these stations exist in any of the drivers' reduced networks. We can perform the same procedure on the links in the time-expanded reduced network of the rider,  $G_r(t)$ . The result is the rider's **time-expanded feasible network,  $G_r^F(t)$** .

Links that fall entirely within  $G_r^F(t)$  are the links that can be traversed by at least one driver. At this stage, we can filter out the set  $D$  even more, by eliminating drivers who do not contribute to the rider's time-expanded feasible network.

## 6 Solution Algorithm

### 6.1 Pre-Processing

The STEM algorithm constructs the time-expanded feasible network,  $G_r^F(t)$ , for rider  $r$ . The optimal route plan for the rider can be searched on  $G_r^F(t)$ . Even though this network is substantially smaller than the original network  $G$ , the number of feasible routes for the rider can still be very large. Different combinations of links on which to travel, and drivers with whom to ride, can make enumerating the paths computationally prohibitive (see Figure 6(b)). In this section, introduce an algorithm using which the optimal path for the rider can be found quite efficiently.

Figure 6(a) shows the time-expanded feasible network for the rider in our example. The goal is to find an optimal route plan for the rider to travel from his/her origin node to his/her destination node. We define the origin node as the node in the rider's origin set,  $OS_r$ , with the smallest time interval, and the destination node as the node in the rider's destination set,  $DS_r$ , with the largest time interval. In our example, the origin node is (3,5), and the destination node is (29,12).

The optimal route plan as defined in this study is a path that minimizes a linear combination of the in-vehicle travel time, wait time, and number of transfers. Before searching for the optimal path, two preliminary operations need to be performed on the graph. First, the nodes that are not descendants of the origin node, and predecessors of the destination node should be removed. An example of this operation is demonstrated in Figure 6(a). This figure displays the time-expanded feasible network of the rider in our example who is travelling from station 5 to station 12. The components of the graph containing nodes (22,7), (23,7), (24,7), and (14,9), (15,9) (marked by red rectangles in Figure 6(a)) have to be eliminated from the graph, since they are not descendants of the origin node (3,5). This task can be easily accomplished by scanning the adjacency matrix of the graph.

The second operation is to topologically sort the graph. A topological ordering for such a graph always exists, since the graph is a directed acyclic one (the graph is acyclic since nodes have a time component, and hence we can never go back in time on a path in the network). Figure 6(b) displays the resulting graph after performing these two preliminary operations. The numbers to the left of nodes in this figure show the topological orders. The numbers on the links show the IDs of the drivers who can carry the rider on the links.

Doing a depth first search (DFS) on the “revised” time-expanded feasible network (Figure 6(b)) determines whether a route plan for the rider exists at all. Starting from the first node in the topologically ordered graph, if DFS does not conquer a node in the destination set, then the rider cannot be matched in the system. Otherwise, if DFS finds a path (establishing that the ride-matching problem is spatially and temporally feasible), we will use the proposed algorithm to find the “best” path for the rider.

## 6.2 Main Algorithm

Let us start by defining sets  $D_j^{IN}$  and  $D_j^{OUT}$  as the set of drivers who enter and exit node  $j$  respectively, and set  $DN_j$  for each node  $j$  as a set of tuples  $(n_i, d)$  such that there is a link for driver  $d$  from  $n_i$  to  $n_j$  in the revised time-expanded feasible network. Furthermore, we denote by  $V(n_j, d)$  the minimum cost of the optimal path from the start node (node 1 in the topologically ordered graph) to node  $n_j$ , such that the last link on the path is traversed by driver  $d$ . The goal is to find  $V_r^* = \min_{n_j \in DS_r, d \in D_j^{IN}} \{V(n_j, d)\}$ .

We initialize the  $N_r \times D_r$  matrix  $V(\cdot)$  to infinity, where  $N_r$  is the number of nodes in rider  $r$ 's revised time-expanded graph,  $D_r$  is the set of drivers that contribute to rider  $r$ 's revised time-expanded feasible network. To keep track of the optimal solution, we introduce two additional matrices of the same size, called *Pred\_Node* and *Pred\_Driver*, and initialize them to zero. These matrices keep the predecessor node and the driver that takes the rider to the predecessor node on the optimal path, respectively.

We set the initial condition as  $V(n_1, d) = 0$ , for all  $d \in D_1^{OUT}$ . In the revised time-expanded feasible network in Figure 6(b), the initial condition is  $V(n_1, d_1) = 0$ .

After setting the initial condition, we traverse the nodes in the (topologically ordered) graph in ascending order. For each node  $n_j$ , and each driver  $d$  in the set  $D_j^{in}$ ,  $V(n_j, d)$  is computed as:

$$V(n_j, d) = \min_{n_i: (n_i, d) \in DN_j} \left\{ \min_{d' \in D_i^{in} \setminus ED(n_i, d')} \{V(n_i, d') + C_T \mathbf{1}_{\{d \neq d'\}}\} + C(n_i, n_j) \right\} \quad (1)$$



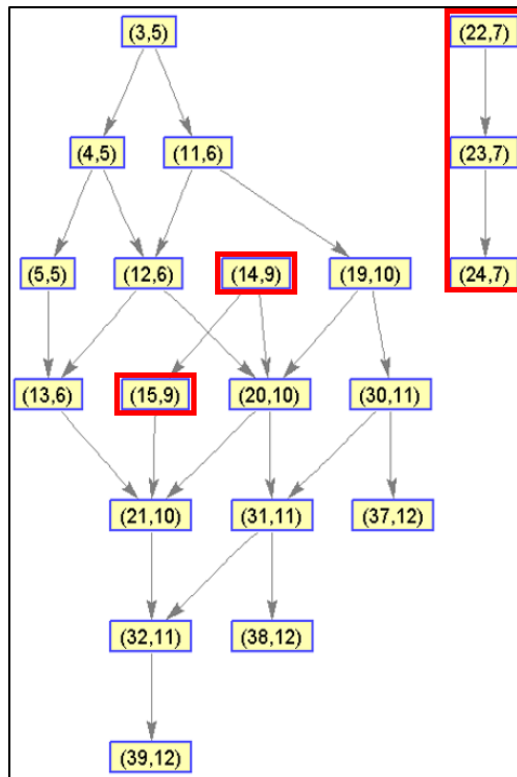
Masoud, Jayakrishnan

The indicator function  $\mathbf{1}_{\{d \neq d'\}}$  returns 1 if  $d \neq d'$  (i.e., if the rider makes a transfer) and returns zero otherwise.  $C_T$  is the penalty (cost) of a transfer. The function  $C(n_i, n_j)$  gives the cost of traveling on link  $(n_i, n_j) = (t_1, s_1, t_2, s_2)$ , and can be calculated using (2). The cost of travel as defined in (2) is the travel time on the link in case a travel takes place (i.e.,  $s_i \neq s_j$ ), or a penalty  $W$  if the rider waits at the station ( $s_i = s_j$ ) for one time interval. Note that if the rider stays in the same station for multiple time intervals, the algorithm will automatically accumulate  $W$  the appropriate number of times. This is due to the fact that the same station at different time intervals corresponds to separate nodes on the time-expanded feasible network. Further, note that  $C(n_i, n_j)$  in (2) can be easily extended to  $C(n_i, n_j, d)$ , in case a more general driver-dependent cost is desired.

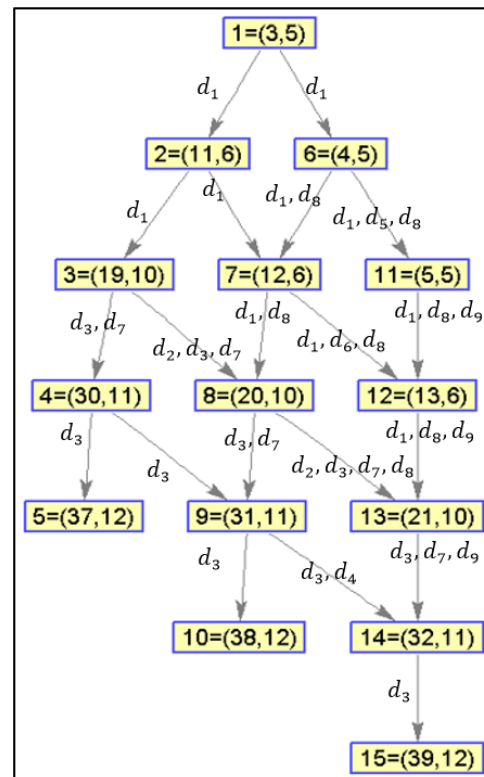
$$C(n_i, n_j) = \begin{cases} (t_j - t_i + 1)dt & s_i \neq s_j \\ W & s_i = s_j \end{cases} \quad (2)$$

We discuss the set  $ED(n_i, d')$  used in (1) further down.

After each computation of  $V(n_j, d)$  using the recursive function in (1), we save  $n_i^* = \operatorname{argmin}_{n_i: (n_i, d) \in DN_j} \left\{ \min_{d' \in D_i^{in} \setminus ED(n_i, d')} \{V(n_i, d') + C_T \mathbf{1}_{\{d \neq d'\}}\} + C(n_i, n_j) \right\}$  in  $\text{Pred\_Node}(n_j, d)$ , and  $d^* = \operatorname{argmin}_{d' \in D_i^{in} \setminus ED(n_i, d')} \{V(n_i, d') + C_T \mathbf{1}_{\{d \neq d'\}}\}$  in  $\text{Pred\_Driver}(n_j, d)$ . Once the optimal minimum cost for the rider is found, these matrices can be used to retrieve the corresponding optimal route plan(s).



6(a) Time-expanded feasible network



6(b) Revised time-expanded feasible network (after performing the preliminary operations)

Figure 6. Time-Expanded Feasible Network For The Rider In Our Example Before And After Revision

We define  $ED(n_i, d') = \{Pred\_Node(n_i, d') \setminus d'\}$ . This set includes a list of drivers on the optimal path to node  $n_i$ , excluding the driver on the last link. Drivers in this set are excluded from set  $D_i^{in}$  in (1) to ensure the feasibility of the solution. Although very unlikely, it could happen that a route includes links that cannot co-exist. If a driver covers multiple links on a rider's route plan, and those links belong to different feasible paths of the driver, then the resulting route plan is not feasible. To clarify, an example is shown in Figure 7. In this example, driver  $d$  has two potential paths to make his/her trip, while driver  $d'$  has only one. There is only one route plan for the rider, marked by dashed red arrows. This route plan consists of three links. The first link is covered by driver  $d$  via driver  $d$ 's first potential path, the second link is covered by  $d'$ , and the third link is covered by  $d$  via driver  $d$ 's second potential path. Clearly the resulting route plan is not feasible, since  $d$  cannot carry the rider on both links. Excluding drivers in set  $ED$  from the list of potential drivers in (1) ensures that such a route plan is never generated. In the example in Figure 7, the rider cannot be served.

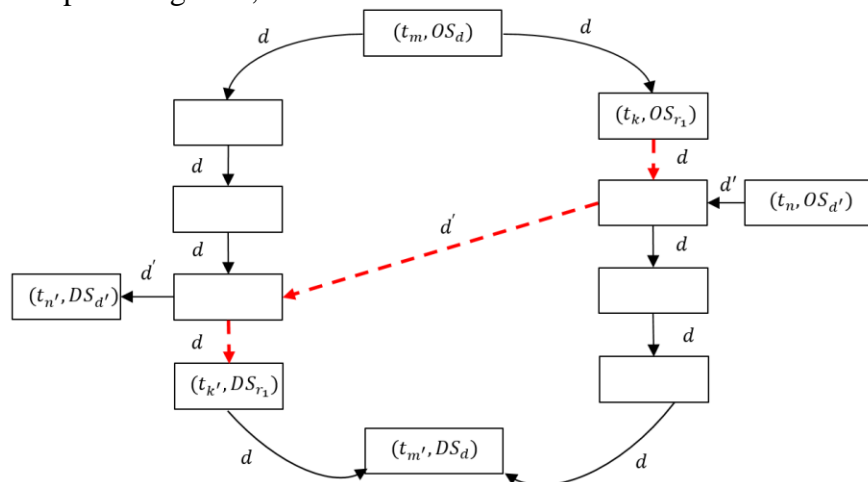


Figure 7. Example of an infeasible solution

Using the proposed dynamic algorithm, not only the optimal route plan, but all the feasible route plans for a rider are readily available. Therefore, if the solution that is deemed optimal based on the objectives of the system (i.e. number of transfers, and traveling and waiting times) did not satisfy other personal requirements of participants, we can easily access and examine the next best solutions.

Once a rider's route plans is fixed, it is easy to find the optimal route plans for drivers. If a driver is part of the rider's route plan, then a portion of his/her path is fixed. Given the fixed portions, it is easy to find the shortest travel time path for the driver from the driver's origin station to the fixed portion, between the fixed portions, and from the last fixed portion to the driver's destination station. Notice that because the fixed portions of a drivers' path have time components, it is easy to order them. After these ordering is done, Dijkstra's algorithm can be used to find the shortest path between the fixed stations in the driver's reduced graph.

## 7 NUMERICAL EXPERIMENTS

In this section we generate multiple random instances of the ridesharing problem in a network with 49 stations. All problem instances contain 1000 participants. We vary the ratio of

Masoud, Jayakrishnan

drivers and riders in each problem instance, and assess the performance of the proposed ridesharing algorithm as the ratio of rider in the problem changes.

The participants' earliest arrival time is generated uniformly within a one hour time period. We discuss the selection of participants' origin and destination stations later on, in each section.

All drivers are assumed to have four empty seats, and riders are assumed to accept up to 3 transfers. We use 1 min time intervals, and consider a randomly generated travel time budget within the interval  $[T_p^{ST} \ 1.1T_p^{ST}]$  for participant  $p \in P$ , where  $T_p^{ST}$  is the shortest path travel time between the participant's origin and destination stations. In the interest of simplicity, we assume that participants have no personal requirements on the people with whom they travel.

In the next two sections, we use the proposed algorithm to find the optimal solutions for the randomly generated problems. In section 7.1, the origin and destination locations of participants will be selected based on a uniform random distribution. In section 7.2, we study a more practical case, where trip ends are determined randomly, but based on a clustering of the network.

### 7.1 Uniform Random Selection of Trip End Locations

For the problem instances generated in this section, the origin and destination stations of participants are selected from the 49 stations, based on a uniform random distribution.

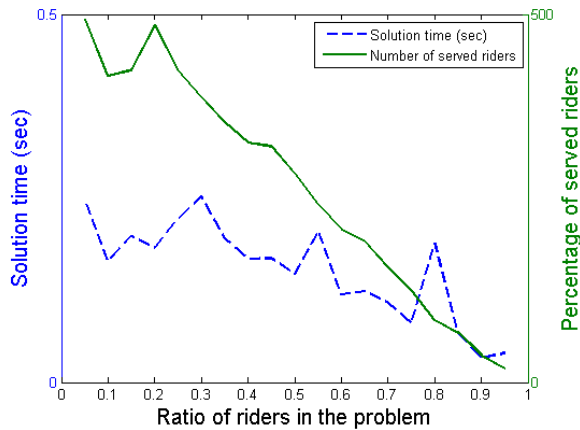
The results we report in this section are averaged over 10 randomly generated instances, for each problem. Figure 8(a) and (b) display the solution time, and the percentage and number of served riders respectively. As mentioned before, riders are considered in a FIFO basis, and once a rider registers in the system, a ride-matching problem for that rider is solved. The solution time reported for each problem instance in figures 8(a) and 8(b) is the highest solution time among all riders in the problem (averaged over 10 randomly generated instances).

Figure 8(a) shows that the percentage of served riders decreases with the ratio of riders in the problem. Note that since the number of participants is held constant, number of drivers decreases, as the number of riders increases. Figure 8(b) suggests that number of served riders is maximized when number of riders in the system is slightly lower than number of drivers, i.e. the ratio of drivers to riders is slightly higher than 1. At his best case scenario (highest number of riders served), about 30% of the demand for rides is satisfied.

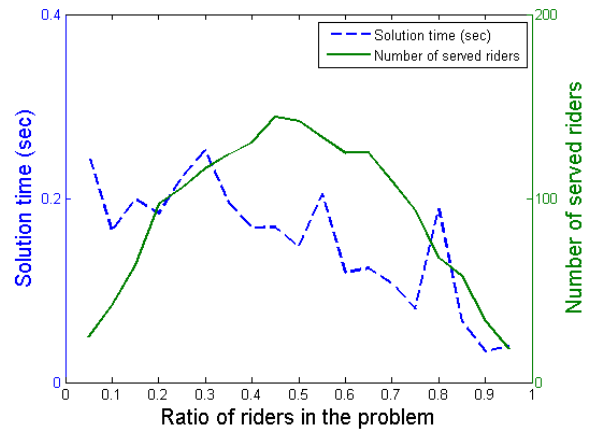
If the interest of the system is to maximize the percentage of served riders, about 50% of riders can be served in the best case scenario. Notice that this higher performance is obtained when the number of riders in the system is only 50, and there are 950 drivers in the system.

Figure 8(c) shows the number of matched participants in the system. This figure suggests that number of matched participants is maximized in the same range where the number of served riders is maximized. In fact, when riders constitute about 45% of the total number of participants, the number of matched riders and drivers are both maximized. Notice that this is not an obvious conclusion, since each driver can carry multiple riders, and each rider can transfer between multiple drivers. Furthermore, notice that these results are only valid for the case where origin and destination locations are randomly selected. We show in the next section that in more practical settings where business districts and residential areas are separated (and therefore trip ends have higher spatial proximity), number of matched participants can increase drastically.

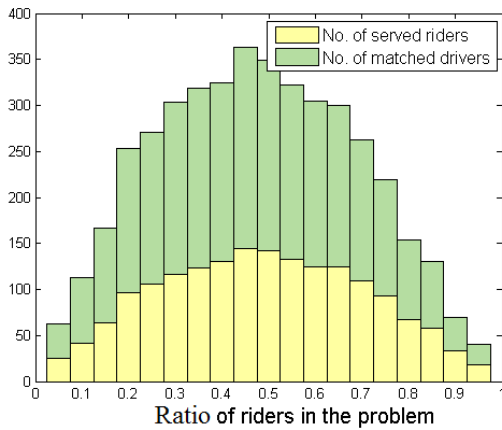
Finally, figure 8(d) shows the distribution of number of transfers for each problem instance. This figure suggests that the majority of satisfied riders do not have to make any transfers. This figure also suggests that the number of additional riders that can be served due to the integration of transfers into the system is not trivial.



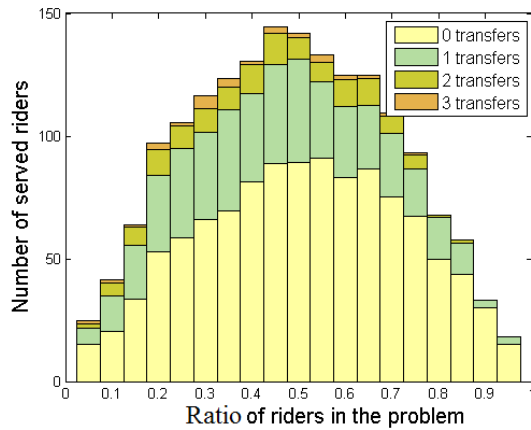
8(a) Change in solution time and percentage of served riders with ratio of riders



8(b) Change in solution time and the number of served riders with ratio of riders



8(c) Distribution of number of participants matched in each problem



8(d) Distribution of number of transfers in the optimal solutions

Figure 8. Algorithm performance for each randomly generated problem instance

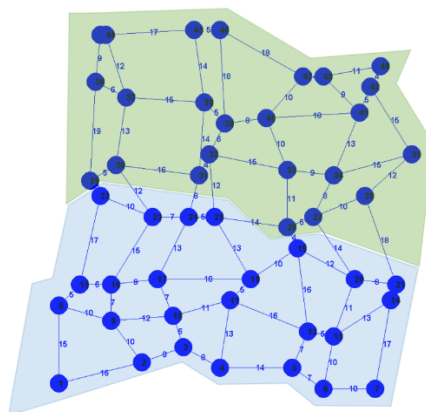
## 7.2 Uniform Random Selection of Trip Ends in Clusters

In the previous section, we made an impractical assumption that the origin and destination locations of trips are completely random. In reality, residential areas are usually close together, and separate from commercial and business districts. In this section, we identify two geographically distinct regions on the network to represent residential and business districts. Similar to the previous section, we generate different problem instances, varying the percentage of riders in the problems. For each ratio of riders, we generate and run 10 different problem instances, and report the average results.

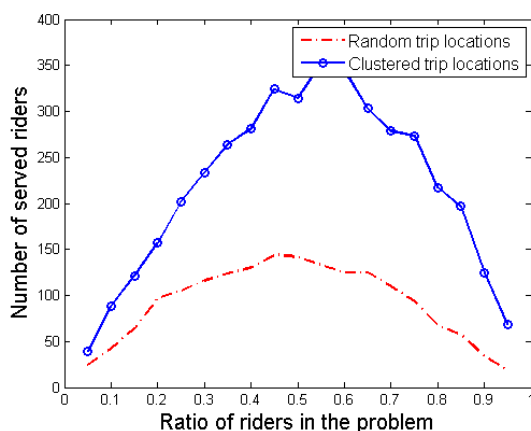
Figure 9(a) shows the clustering of a sample network. The area on the top is considered the business district, and the area on the bottom the residential district. Each problem instance represents the simulation of a morning peak period, in which participants travel from a randomly selected station in the residential area, to one in the business district.

Masoud, Jayakrishnan

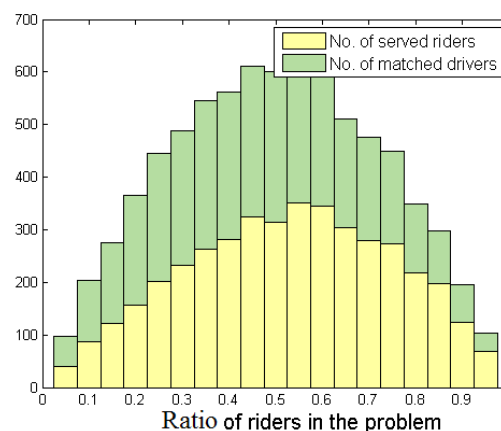
Figure 9(b) shows that number of served riders has the same trend compared to the scenario where origin and destination locations were completely random, but in this case the number of served riders has more than doubled. Figure 8(c) suggests that in this more practical setting, when the system has the optimum ratio of drivers and riders, more than 60% of the participants will actually participate in the system.



9(a) A sample clustering of the network



9(b) Comparison of number of served riders between the completely random scenario and the clustering scenario



9(b) Distribution of matched participants in the clustering scenario

Figure 9. System performance for the scenario with distinct business and residential areas

## 8 The critical mass

For a ridesharing system to be able to work independently, a critical mass of participants is required. In the previous sections, we demonstrated the performance of a ridesharing system with 1000 participants. We showed that in the more practical scenario of having distinct business and residential areas, more than 60%, and in the completely random case, about 35% of participants can be successfully matched. It is intuitive to expect the average number of successful matches to increase with the number of participants. However, for a system to survive, a minimum number of participants should be able to use the system successfully. This

Masoud, Jayakrishnan

success will encourage users to consider participating in the system again, and promote the system to their friends and families.

In this section, we do a sensitivity analysis on the number of participants in the system. This analysis can shed light on the relationship between system performance, and the number and ratio of participants.

Figure 10 shows the percentage of matched participants as the total number of participants changes from 200 to 1000 individuals. Similar to the previous sections, for a given number of participants, we generate multiple random problem instances by changing the ratio of riders. The travel ends of participants were selected at random with uniform probabilities, similar to section 7.1. Figure 10 suggests that the increase in the number of participants leads to higher system performance, in terms of the percentage of matched participants. The peak performance of the system in all cases, however, occurs in the same range.

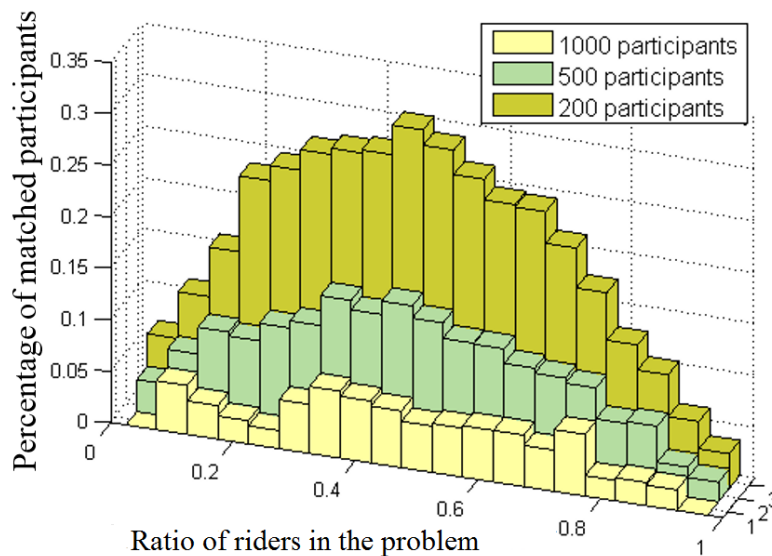


Figure 10. Sensitivity analysis over the number of participants in the system

## 9 P2P Exchange

Ridesharing systems are in general spatiotemporally sparse. The temporal sparsity stems from the rather tight time windows and limited travel time budgets of participants, and the spatial sparsity is due to the participants' fixed origin and destination locations. This spatiotemporal sparsity limits the number of satisfied requests. In fact, one of the contributors to the initial failure of P2P ridesharing systems in the US in the 1990s was the very small number of served rider requests. Today, advancements in the communication technology and prevalence of smartphones, along with the young generation's fascination with use of technology has led to a larger pool of riders and drivers for ridesharing systems. However, as discussed in the previous sections, a critical mass of participants is still an important requirement for ridesharing systems, to ensure that they can operate independently, without having to outsource drivers. Therefore, a ridesharing system has to try and make the best use of its limited available resources, the drivers.

A ridesharing system that takes riders into consideration only one at a time, is in fact wasting its very limited and valuable resources by fixing route plans of matched drivers. Figure 11 shows an example of how fixing drivers' route plans can deteriorate the performance of the

Masoud, Jayakrishnan

system. This example includes two riders and two drivers. Assume rider 1 has registered in the system before rider 2, and hence the system starts by finding a match for rider 1. Vehicles 1 and 2 are both candidates to be matched with this rider. Eventually the system matches rider 1 with driver 2, since driver 1 has to make a larger detour to carry the rider. Driver 2's route plan, marked in solid blue line, is then fixed. The next rider in queue is rider 2. This rider could have been matched with driver 2 through the route marked with the blue dashed line, if driver 2's route plan had not been fixed. However, under the current state of the system, rider 2 cannot be matched. The ridesharing system, therefore, can only match one rider with one driver.

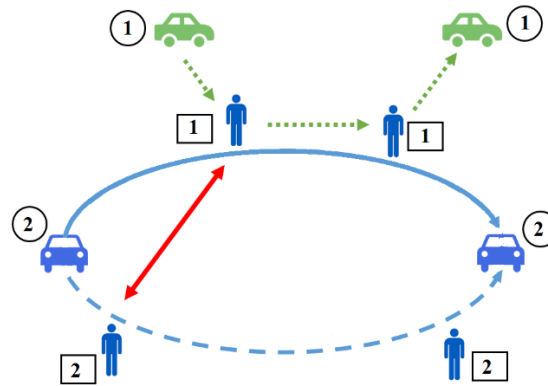
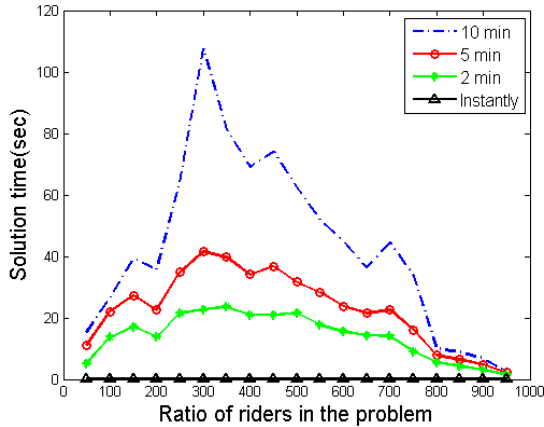


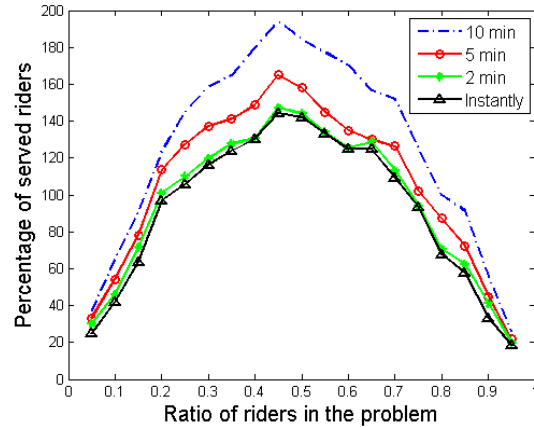
Figure 11. P2P ride exchange

In the example in figure 11, we could have increased the system performance by solving a ride-matching problem that included the two available riders, as opposed to considering them one at a time. The solution to such a problem would match rider 1 with driver 1, and rider 2 with driver 2, serving both riders, and involving all 4 participants.

The problem with this approach is that solving such a problem is too time-consuming. Even efficient algorithms (such as the one proposed by Masoud and Jayakrishnan, 2015) can take minutes to solve, for moderate size instances of the problem. To demonstrate the difference in solution time of a many-to-many ride-matching problem (which includes multiple riders and multiple drivers) with the one-to-many problem (which includes one rider and multiple drivers) proposed in this paper, we solved the problem instances in section 7.1 using both approaches. For the many-to-many problem, we use the decomposition algorithm proposed by Masoud and Jayakrishnan (2015) on a time-horizon basis, by reoptimizing the system at 2, 5, and 10 min time intervals. In this approach, instead of matching one rider at a time, we solve a problem that includes all the riders that can start their trips within the mentioned time interval (such riders can be determined based on the travel time windows of their trips). Figure 12 depicts the results of this comparison.



12(a) Change in the solution time



12(b) Change in the number of served riders

Figure 12. Solution time and number of served riders for each randomly generated problem instance

Figure 12 suggests that the solution time for a many-to-many ride-matching problem is too high for real-time applications. On the other hand, the number of matched riders improves when a many-to-many problem is solved. In this section, we propose “P2P ride exchange” as a mechanism to improve the solution of a many-to-one ride-matching problem.

Although we can use the proposed algorithm in this paper to solve a many-to-one ride-matching problem to optimality, the formulation of the problem itself is not optimal. P2P ride exchange can help us move from the sub-optimal solution obtained by matching one rider at a time, toward the optimal solution that can be obtained by including all riders in the matching problem, without the unattractive side-effect of the increased solution time.

In this approach, we still solve the matching problem for one rider at a time, and on a FIFO basis. However, we do not fix the route plans of matched drivers. If a rider can be served only using a driver that has been previously matched (but through a conflicting path, or schedule), then the two riders can start negotiating, and possibly making a ride exchange. Note that this is easy for the system to propose alternative route plans to riders involved in a negotiation, since all the feasible route plans for each rider are generated using the algorithm in section 6.2.

In the example in figure 11, if we solve the ride-matching problem for rider 2 without fixing the route plan of driver 2, the solution will route driver 2 on the dashed path, and match it with rider 2. However, driver 2 was previously routed differently, and assigned to rider 1. These two riders can now engage in a negotiation. Since rider 1 can also be matched with driver 1, there is a good chance that rider 1 will accept the proposed ride exchange by rider 2, in exchange for money, or credit toward using the ridesharing system. In case the negotiation is successful, the final matching would be equal to the matching obtained by solving a many-to-many ride-matching problem.

In general, if all riders reach to agreements to make the ride exchanges suggested by the system, the number of served riders would ultimately equal the number of matches obtained from a many-to-many problem.

In addition to the benefits that P2P exchange can offer to the system, it can also make participating in the system more fun. Riders can earn money or credit by selling their route plans, and either settling for less efficient paths suggested by the system, or using other modes of transport. However, for such a system to work properly, a good mechanism should be designed to ensure that individuals cannot take advantage of the system by booking false rides.



## 10 Overlapping sets of drivers and riders

Initially, we made the assumption that riders and drivers form two mutually exclusive sets. In this section, we study the scenario in which participants who register in the system as riders might be doing so because they prefer travelling as riders, and not because they don't own or have access to a vehicle. If the system is not able to serve these participants as riders, then they will drive their own vehicles, and join the system as drivers. An extreme case would be to study the case where all riders have access to vehicles, and are willing to join the system as drivers. This case shows the maximum benefits a ridesharing system can offer.

In this section, we study the impact of this assumption on the problem instances in section 7.1. Figure 13 shows the maximum number of served riders under different percentage of riders who later join the system as drivers. In the case of 0%, the problem becomes equivalent to the problem of a system with mutually exclusive set of riders and drivers. As this percentage increases, the shape of the curve starts to change. The best performance of the system can be obtained when 100% of riders have access to vehicles, and are willing to join the system as drivers, if they cannot be served as riders. It is interesting to see that in this best case scenario, the number of served riders has an increasing trend.

In the basic case of the problem, number of served riders reaches its peak when riders constitute around 45% of the participants. As the ratio of riders increases, the number of served riders decreases, because there are fewer number of drivers available. If riders who cannot be matched join the system as drivers, this practically increases the number of drivers in the ranges where previously there was a shortage of drivers, and this leads to higher number of riders being served. Figure 13 suggests that encouraging such a strategy in ridesharing systems can drastically improve its performance.

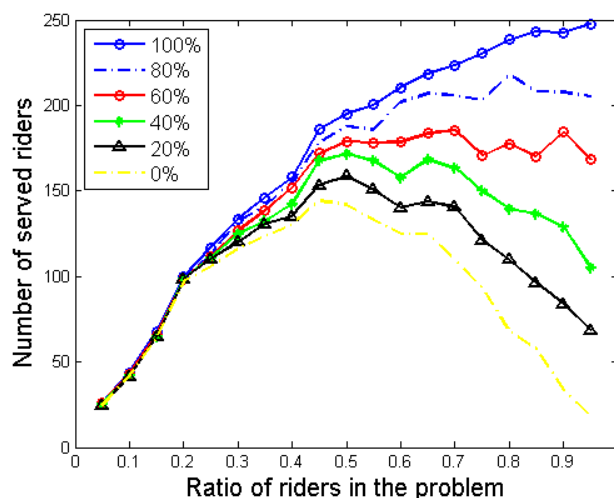


Figure 13. Number of served riders under different percentage of riders who become drivers

## 11 SUMMARY AND CONCLUSIONS

In this paper, we define a flexible ridesharing system as a multi-hop system with the ability to find route plans for riders by means of optimally routing drivers, and provide rides in real-time. Real-time ride-matching is a central part of a flexible P2P ridesharing system. It ensures that users who look for rides not long before their departure times have a chance to be

Masoud, Jayakrishnan

served in the system. We propose a real-time ride-matching algorithm that maximizes the number of served riders in the system, while making the trips as comfortable as possible by taking into consideration users' preferences on whom to ride with, and minimizing the number of transfers and waiting times for riders.

We introduce the concept of P2P ride exchange to increase the number of successful matches in the system. In this framework, previously matched riders receive offers from other riders to give up their route plans in exchange for money, credit, or simply receiving another route plan. The numerical studies suggest that P2P ride exchange can potentially increase the number of served riders.

Finally, we study the scenario where users who initially join the system as riders, but are unsuccessful in finding a ride, drive their private vehicles, and join the system as drivers. Results suggest that encouraging such a behavior could substantially improve the performance of the ridesharing system.

## References

- Agatz, N., A. Erera, M. Savelsbergh, and X. Wang (2009). Sustainable passenger transportation: Dynamic ride-sharing. ERIM Report Series Reference No. ERS-2010-010-LIS.
- Agatz, N. A., A. L. Erera, M. W. Savelsbergh, and X. Wang (2011). Dynamic ridesharing: A simulation study in metro atlanta. *Transportation Research Part B: Methodological* 45(9), 1450-1464. Select Papers from the 19th {ISTTT}.
- Baldacci, R., V. Maniezzo, and A. Mingozzi (2004). An exact method for the car pooling problem based on lagrangean column generation. *Operations Research* 52(3), pp. 422-439.
- Böckmann, M. (2013). The shared economy: It is time to start caring about sharing; value creating factors in the shared economy. University of Twente, Faculty of Management and Governance.
- Braekers, K., A. Caris, and G. K. Janssens (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological* 67(0), 166 -186.
- Business Insider (2014). Uber Is Generating A Staggering Amount Of Revenue. Internet: <http://www.businessinsider.com/uber-revenue-projection-in-2015-2014-11>. Accessed: 2015-06-20.
- Business Insider (2015). Uber Just Released Its First Report On Its Drivers Here Are The Numbers. Internet: <http://www.businessinsider.com/uber-driver-data-report-2015-1>. Accessed: 2015-06-20.
- Calvo, R. W., F. de Luigi, P. Haastrup, and V. Maniezzo (2004). A distributed geographic information system for the daily carpooling problem. *Computers & Operations Research* 31(13), 2263-2278.
- Carnes, T. A., S. G. Henderson, D. B. Shmoys, M. Ahghari, and R. D. MacDonald (2013). Mathematical programming guides air-ambulance routing at ornge. *Interfaces* 43(3), 232-239.
- Cordeau, J.-F., and G. Laporte (2007a). The dial-a-ride problem: models and algorithms. *Annals of Operations Research* 153(1), 29-46.
- Cordeau, J.-F., and G. Laporte (2007b). The dial-a-ride problem: models and algorithms. *Annals of Operations Research* 153(1), 29 46.

Masoud, Jayakrishnan

CortØs, C. and R. Jayakrishnan (2002). Design and operational concepts of high-coverage point-to-point transit system. *Transportation Research Record: Journal of the Transportation Research Board* (1783), 178-187.

Di Febbraro, A., E. Gattorna, and N. Sacco (2013). Optimizing dynamic ride-sharing systems. Presented in the TRB 2013 annual meeting.

Ghoseiri, K. (2013). Dynamic rideshare optimized matching problem. PhD Dissertation at University of Maryland.

Haselkorn, M., J. Spryidakis, C. Blumenthal, S. Michalak, B. Goble, and M. Garner (1995). Bellevue smart traveler: design, demonstration and assessment.

Herbawi, W. and M. Weber (2011). Ant colony vs. genetic multiobjective route planning in dynamic multi-hop ridesharing. In *Tools with Artificial Intelligence (ICTAI), 2011 IEEE Conference*.

Herbawi, W. and M. Weber. (2011). Comparison of multiobjective evolutionary algorithm for solving the multiobjective route planning in dynamic multi-hop ridesharing. *Proceedings of the 11th European conference on Evolutionary computation in combinatorial optimization, Torino, Italy*, 84-95.

Herbawi, W. and M. Weber (2012). A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*.

Jaw, J.-J., A. R. Odoni, H. N. Psaraftis, and N. H. Wilson (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological* 20(3), 243 - 257.

Levofsky, A. and A. Greenberg (2001). Organized dynamic ride sharing: The potential environmental benefits and the opportunity for advancing the concept. In *Transportation Research Board 2001 Annual Meeting*.

Li, X. and L. Quadrifoglio (2010). Feeder transit services: choosing between fixed and demand responsive policy. *Transportation Research Part C: Emerging Technologies* 18(5), 770-780.

Liaw, C.-F., C. C. White, and J. Bander (1996, Sep). A decision support system for the bimodal dial-a-ride problem. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 26(5), 552-565.

Masoud, N. and R. Jayakrishnan (2015). A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem. In *Transportation Research Board 94th Annual Meeting, Number 15-2755*.

Masson, R. et al. (2014). The dial-a-ride problem with transfers. *Computers & Operations Research* 41(0), 12 - 23.

O'Sullivan, S. (2011). Case study in real-time ridesharing: Sr 520 carpooling pilot project, seattle, WA. In *18th ITS World Congress*.

Psaraftis, H. N. (1983). An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17(3), 351-357.

Qiu, F., W. Li, and J. Zhang (2014). A dynamic station strategy to improve the performance of ex-route transit services. *Transportation Research Part C: Emerging Technologies* 48, 229-240.

Quadrifoglio, L., M. M. Dessouky, and F. Ordóæez (2008). Mobility allowance shuttle transit (mast) services: Mip formulation and strengthening with logic constraints. *European Journal of Operational Research* 185(2), 481-494.

Masoud, Jayakrishnan

RTrip (2009). RTrip. Internet: <http://www.gortripblog.com/2011/07/avego-go520-is-new-and-improved>. Accessed: 2014-04-30.

Savelsbergh, M. W. P. and M. Sol (1995). The general pickup and delivery problem. *Transportation Science* 29(1), 17-29.

Schaub, T., G. Friedrich, and B. O'Sullivan (2014). *ECAI 2014: 21st European Conference on Artificial Intelligence, Volume 263*. IOS Press.

Shaheen, S. and N. Chan (2015). *Mobility and the sharing economy: Impacts Synopsis*. Technical report, TRANSPORTATION SUSTAINABILITY RESEARCH CENTER, University of California Berkeley.

Stein, D. M. (1978). Scheduling dial-a-ride transportation systems. *Transportation Science* 12(3), 232-249.

Teodorovic, D. and M. DellOrco (2005). Bee colony optimization-a cooperative learning approach to complex transportation problems. In *Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini-EURO Conference and 10th Meeting of EWGT (13-16 September 2005)*. Poznan: Publishing House of the Polish Operational and System Research, pp. 51-60.

U.S. Department of Transportation (2006). *National strategy to reduce congestion on America's transportation network*. pdf: <http://isddc.dot.gov/OLPFiles/OST/012988.pdf>.

Vukan, V. (1981). *Urban public transportation: Systems and technology*.