
Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

# A Ride-Sharing Problem with Meeting Points and Return Restrictions

Wenyi Chen, Martijn Mes, Marco Schutten, Job Quint

informs
http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Vol. 53, No. 2, March–April 2019, pp. 401–426
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

# A Ride-Sharing Problem with Meeting Points and Return Restrictions

**Wenyi Chen,**[a,b] **Martijn Mes,**[b,*] **Marco Schutten,**[b] **Job Quint**[b]

[a] Department of Supply Chain Management and Information Systems, ESC Rennes School of Business, 35065 Rennes, France;
[b] Department of Industrial Engineering and Business Information System, University of Twente, 7522 NB Enschede, Netherlands

[*] Corresponding author

**Contact:** wenyi.chen@esc-rennes.com, http://orcid.org/0000-0003-2058-7745 (WC); m.r.k.mes@utwente.nl, http://orcid.org/0000-0001-9676-5259 (MM); m.schutten@utwente.nl, http://orcid.org/0000-0001-5924-223X (MC); j.quint@utwente.nl (JQ)

**Received:** September 29, 2016
**Revised:** November 29, 2017
**Accepted:** February 8, 2018
**Published Online in Articles in Advance:** January 30, 2019

**https://doi.org/10.1287/trsc.2018.0832**



**Abstract.** Ride sharing has been widely acknowledged as an effective solution for reducing travel costs, congestion, and pollution. This paper considers the ride-sharing problem of the scheduled commuter and business traffic within a closed community of companies that agree to share the calendars of their employees. We propose a formulation in the form of a general integer linear program (ILP) for the aforementioned ride-sharing problem, which incorporates return restrictions to satisfy the business needs, as well as meeting points and the option for riders to transfer between drivers. All the instances with 40 and 60 participants and most of the instances with 80 participants can be solved to optimality within a time limit of two hours. Using instances of up to 100 participants, the ILP can be solved with a gap of no more than 1.8% within the time limit. Because of the high computational complexity, we develop a constructive heuristic that is based on the savings concept. This heuristic is also able to combine ride sharing with the use of an external mobility service provider. Our numerical study shows that ride sharing can be an effective way of reducing the number of trips and vehicle miles. Particularly, ride sharing creates more benefits when the participation is high and when the origins and the destinations of the trips are more spatially concentrated. The results show that ride sharing can create up to 31.3% mileage savings and up to 28.7% reduction in the number of cars needed to fulfill employees' travel schedules. We also illustrate our model using a real-life ride-sharing problem of a Dutch consultancy and research firm.

**Funding:** This research was partially funded by Dinalog, the Dutch Institute for Advanced Logistics, as part of the research and development project CargoHitching.
**Supplemental Material:** The online appendix is available at https://doi.org/10.1287/trsc.2018.0832.

**Keywords:** ride sharing • carpooling • sustainable transport • transfers • return restrictions

## 1. Introduction

Rising motorization and increasing traffic density have intensified the problem of greenhouse gas (GHG) emissions worldwide (Kuntzky, Wittke, and Herrmann 2013). The transport sector accounted for 24.4% of the European Union's total GHG emissions in 2013; passenger cars contributed almost 45% of the transport sector's emissions. Although the fuel efficiency and emission characteristics of passenger cars have improved steadily, rapid growth in car ownership and in distances traveled offset any potential improvements in terms of environmental impacts (European Environment Agency 2013). Even with increasing environmental awareness and concern, many road users are still car dependent, either by choice or because they are constrained by circumstances (Stradling 2007). Given the constantly low car occupancy rates—for example, 1.2 persons per car on average in Lisbon, 1.3 in Sydney, 1.5 in London, and 1.6 in Singapore (International Transport Forum 2015)—ride sharing plays an increasingly important role in providing mobility and reducing $CO_2$ emissions, traffic congestion, and parking problems. Assuming that one person was added to each commute, the International Energy Agency (2005) estimated that carpooling could reduce the number of kilometers traveled by 12.5%, which would lead to a 7.7% reduction in fuel use. Delhomme and Gheorghiu (2016) provide an example: the people who carpool for a distance of 48 kilometers could save up to 33% of the monthly costs of commuting compared to those who choose to drive alone. Additionally, ride sharing may save time because commuters are able to use high-occupancy vehicle (HOV) lanes reserved for the vehicles with two or more occupants, reduce driver fatigue (Stiglic et al. 2015), and lead to social benefits by enlarging carpoolers' social networks (Agatz et al. 2012).

Ride-sharing services currently on the market range from simple online bulletin boards to sophisticated systems that offer real-time, on-demand matching, routing, and payment service (Stiglic et al. 2015; Furuhata et al. 2013). Ride-sharing providers like Uber, Lyft, and Flinc show that innovative use of technology can revolutionize personal mobility (Savelsbergh and Van Woensel 2016). Because of the challenge of coordinating itineraries and schedules between participants, ride-sharing coordination is mostly an informal and disorganized activity, and only in certain cases can travelers make use of ride sharing as a regular transportation alternative (Furuhata et al. 2013). In this paper, we focus on systems that offer automated matching of scheduled commuter/business traffic within a closed community of companies that agree to share their calendars. An example of a provider offering such a service is Zimride (http://zimride.com/). The service provider connects intercity drivers and passengers through social networking. One of its signature services is to provide a private ride-sharing network within a corporation or a university (e.g., Harvard University). People within the same network register their trips, typically on a daily basis. Each trip contains detailed information, such as the origin and destination, the earliest departure and latest arrival times, the maximum acceptable detour, and the capacity of the car. In addition, it also contains registration information of the person, such as age, gender, educational level, special interests, etc., which may also be obtained from his or her social network. The system provides suggestions to individuals to share a car based on the details of their trips as well as their registration information.

Companies also have a financial motivation to support ride sharing among employees, especially companies that offer individually owned company cars to their employees. Since employees are not directly confronted with the marginal costs of using the cars, access to company cars leads to higher car use (van Dender and Clever 2013) and higher costs for companies. It is also socially undesirable to have cars with low occupancy on the road. Therefore, it is to a company's benefit to reduce the number of single-person trips by car. A recent development has been to replace individually owned company cars with accessibility to mobility in general, through the use of public transport and shared cars. Ride sharing among colleagues can also be a promising solution. By promoting ride sharing, a company can reduce their total mobility cost by reducing the total vehicle miles driven by all employees as well as the total number of vehicles needed to fulfill their mobility demands. Furthermore, the outcomes of ride sharing are also aligned with societal objectives for reducing emissions and traffic congestion, which should also be of companies' interest concerning their corporate social responsibility.

The goal of this paper is to provide the means for a closed corporate community to facilitate ride sharing by matching the employees. Such a community can be a company or a consortium of companies that are willing to share their calendars in some way. It can also be initiated by a car leasing company that aims to provide mobility services instead of only leased cars. In general, employees' agendas are well planned in advance. Thus, we limit our attention to the offline ride-share matching problem: given the detailed information about people's trips within a given time period, such as the maximum acceptable detour and the capacity of the car, as well as the information about the users, find the optimal matching to minimize the overall cost of the commuter traffic. To increase the chance of matching, we incorporate the features of meeting points and multiple hops in the model. Furthermore, we provide a heuristic for solving nontrivial problem instances of the considered NP-hard optimization problem.

From a company's perspective, the ride-sharing service should not affect the employees' work-related mobility. Thus, one of the main contributions of our work is the incorporation of return restrictions; that is, a match is possible only if (i) the rider can reach all his destinations during the planning horizon and (ii) he can return to the meeting point where his car is parked to drive to his final destination. As a result, the role of a person being a driver or a passenger is not fixed. Even more challengingly, it can change during the planning horizon.

The remainder of this paper is organized as follows. In the next section, we position our research in the context of the relevant literature. After introducing the ride-sharing model with meeting points and return restriction (RS-M&R) in Section 3, we present the mixed integer programming formulation in Section 4. We propose a heuristic for solving the RS-M&R in Section 5. In Section 6, the proposed methodology is illustrated using a real-life ride-sharing problem of Significant BV, a Dutch consultancy and research firm. Section 7 provides the experimental settings. The numerical results are presented in Section 8. Section 9 concludes this paper with key findings and directions for future research.

## 2. Literature Review

In recent years, ride sharing has received growing interest from both academia and industry. In this section, we relate our work to the existing literature by discussing the features comprised by the new model we propose in the next section. In particular, we analyze the extent to which these features have been addressed in the literature. For a more comprehensive overview of the ride-sharing literature, we refer the reader to the recent review paper by Furuhata et al. (2013), where the authors provide an extensive overview of the literature by presenting the state of the art of existing ride-sharing systems and discussing the key challenges in the widespread use of ride sharing.

Traditionally, optimization of ride-sharing services involves solving a class of complex vehicle-routing problems with time constrained pickups and deliveries (Mahmoudi and Zhou 2016), or solving dynamic pickup and delivery problems (Berbeglia, Cordeau, and Laporte 2010). For example, Hosni, Naoum-Sawaya, and Artail (2014) formulate the shared-taxi problem, which is to assign passengers to taxis and compute the optimal routes of taxis as a mixed integer program. A Lagrangian decomposition heuristic is proposed to solve the problem. Wang, Dessouky, and Ordonez (2016) modify existing pickup and delivery problems with time windows to study how the optimal routes change as a function of incentives for ride sharing. In particular, they consider changes in passenger travel time and toll cost due to the availability of HOV lanes. Their results show that it can be beneficial to take detours to pick up additional passengers when the time saving on HOV lanes is significant.

The popularity of ride-sharing providers has stimulated a growing body of research on optimization technology in the dynamic setting of this area, for example, Winter and Nittel (2006), Agatz et al. (2011), Stiglic et al. (2015), Lee and Savelsbergh (2015), and Stiglic et al. (2016). Agatz et al. (2012) provide an overview of dynamic ride sharing and the relevant algorithmic approaches for matching drivers and riders in real time. They point out that transfers are not considered in the literature yet because of the increasing computational burden resulting from the increased number of drivers and riders that are involved in a matching in a multihop setting. Along the same vein, Naoum-Sawaya et al. (2015) study the ride-sharing systems within large organizations and propose a stochastic optimization model to determine the optimal distribution of limited number of company vehicles while insuring robustness against vehicle unavailability.

One way to increase a rider's chance of finding a match is to allow him to transfer between different drivers to reach his destination. Gruebele (2008) describes such a multihop and multirider routing system in detail, without providing a solution methodology. Herbawi and Weber (2011) consider a single-rider version of the multihop ride-sharing problem, where drivers do not deviate from their routes and schedules. An evolutionary multiobjective route-planning algorithm is used to obtain good quality solutions in reasonable runtime. Herbawi and Weber (2012) extend the previous work to match multiple riders with multiple drivers having time windows and allowing possible detours from their routes. They propose a genetic algorithm and show that it can be used to solve the model in reasonable time. Drews and Luxen (2013) show that the problem studied by Herbawi and Weber (2012) can also be solved by exploiting time-expanded graphs representing the drivers' offers.

In the traditional recurring ride-sharing problem, a match is possible only if a driver is able to pick up the rider at his starting location and drop him off at his ending location. A more recent development to facilitate ride sharing is to consider a setting in which commuters travel to and from meeting points. We are aware of three papers that consider meeting points in the context of ride sharing: Aissat and Oulamara (2014), Bruck et al. (2017), and Stiglic et al. (2015).

Assuming the matching between a driver and a rider is done, Aissat and Oulamara (2014) focus on finding the start and end meeting points to minimize the total travel distance of all the drivers. No restriction is placed on the locations of the starting and end meeting points relative to the rider's origin and destination, respectively. If a rider's travel effort from his origin to the starting meeting point and from the end meeting point to his destination is considered, the optimal solution may be very different.

Bruck et al. (2017) study a daily carpooling problem of an Italian company, whose aim is to encourage its employees to carpool to reduce transportation costs and $CO_2$ emissions. They focus on a case in which all employees are headed to the same workplace but may have different working shifts. Assuming employees with the same shift can carpool together to reach a common destination at the same time, time windows are not considered. Employees are allowed to drive to intermediary points and then use a single car from there on. To limit the complexity of the problem, however, only home addresses of the employees are allowed to be potential meeting points.

Closely related research that also considers ride sharing with meeting points can be found in Stiglic et al. (2015). This work represents the single-driver, multiple-rider setting. The drivers are allowed to make only one pickup and one drop-off. The potential locations of the pickup and drop-off points are confined in a certain walking distance from the riders' origins and destinations. Our work is different in several ways. First, the participants who have a car are also allowed to ride with others. Thus, potential meeting points are defined differently, and the participants with a car can have flexible roles. Second, transfers are allowed. Third, return restrictions are incorporated. Fourth, a participant can have more than two trips during the planning horizon. As a result of the listed differences, the bipartite matching structure and the solution method proposed in Stiglic et al. (2015) are not applicable to our problem.

Agatz et al. (2011) also consider return trip matches and flexible roles, but focus on the single-driver, single-rider ride-sharing problem. When instances contain large numbers of participants with flexible roles, they find it difficult to solve the offline problems to optimality.

In this paper, we consider a multihop ride-sharing problem with the incorporation of meeting points and return restrictions. The contribution of this paper is threefold. First, we design an integer linear program (ILP) specifically tailored to the essentials of a closed corporate ride-sharing environment, which is characterized by the foreseeability of the participants' schedules yet limited flexibility in their schedules and itineraries. Accordingly, our model incorporates the following design features: (i) meeting points; (ii) return restrictions; (iii) transfers; (iv) flexible roles of being a driver or a rider, if possible; and (v) the possibility of switching roles during the planning horizon. Second, we show that the ILP provides high-quality solutions to ride-sharing problem instances with up to 100 participants, and also provides useful insights into the potential benefits of implementing ride sharing. Third, considering the difficulty in finding a good feasible solution for such a complex problem, we propose a constructive heuristic, which can serve as a good starting point for improvement heuristics for solving large-scale problem instances. The proposed heuristic is also able to consider an extended problem where an external mobility service (EMS) provider can be used as a backup option.

## 3. Problem Description

The problem under consideration is defined on a directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes (denoted by $i$, $j$, $k$) representing the possible locations for departure, arrival, or transfer, and $\mathcal{A}$ is the set of arcs that directly connect two aforementioned locations, that is, it represents the road network. With each arc $(i, j) \in \mathcal{A}$, a travel time $t_{ij}$ is associated based on the travel distance.

We are given a set of persons $\mathcal{P}$, with or without cars, who need to travel from one location to another. During the planning horizon, say a day, a person $p \in \mathcal{P}$ might have multiple trips, each of which is considered as a ride $r \in \mathcal{R}$. Generally speaking, a person will have two trips of commuter traffic plus some business trips during the day, if any. These trips are ordered chronologically. For example, if a person has three trips for a day, then the first (i.e., $r = 1$) and the third (i.e., $r = 3$) trips belong to the commuter traffic, while the second trip (i.e., $r = 2$) belongs to the business traffic. On ride $r$, $p$ will travel from his origin $o_{pr}$ to his destination $d_{pr}$, and $SP_{pr}$ represents the set of arcs belonging to his shortest path from $o_{pr}$ to $d_{pr}$. An earliest time $e_{pr}$ at which he can depart from his origin $o_{pr}$ and a latest time $l_{pr}$ at which he has to arrive at his destination $d_{pr}$ are also associated with person $p$. For those participants who have a car, the choice of being a driver or a passenger is flexible. A driver may take a single passenger or multiple passengers (sequentially or simultaneously) along the journey, as long as the capacity of his vehicle $v_p$ is not exceeded. Similarly, a passenger may ride with a single driver from his origin to his destination or may transfer from one driver to another en route to his destination. It is possible that a participant $p$ needs to wait during the transfers, the total waiting time is capped by the participant's maximum waiting time $m_{pr}$.

Our objective is to develop a mechanism for ride sharing within a closed corporate community to reduce the overall cost of operating the commuter and business traffic, which consists of the costs associated with (i) vehicle miles such as fuel and tolls and (ii) the inconvenience and efficiency loss due to transfers and other time losses. One of the key characteristics of the commuter/business traffic is the limited flexibility of employees to change their itineraries and schedules. To overcome this challenge, the ride-sharing system has to be well designed to minimize the effort and inconvenience for the participants.

That being said, three important features are considered in our ride-sharing problem. First, the participants will be matched only when limited detour is required, and a preprocessing procedure will be introduced to determine the limited allowable detour in Section 6. Second, meeting points are introduced to take advantage of any flexibility of every participant in terms of time and mobility, and only routes with small detours are constructed. Most probably, people have multiple trips during a day. The simplest example is a return trip from work to home. Thus, the third feature is to impose return restrictions, such that (i) the participants who leave their car and start sharing rides are able to arrive at their destinations throughout the day, and (ii) the participants and their cars are able to return to their home locations at the end of the day.

Let $Z_{prij}$ be a binary variable that denotes whether person $p$ travels as a driver on arc $(i, j)$ on his ride $r$. Let $Y_{pqrwij}$ be a binary variable that represents whether passenger $p$ rides with driver $q$ on $p$'s ride $r$ and $q$'s ride $w$ on arc $(i, j)$. Let $D_{pri}$ and $A_{pri}$ denote person $p$'s departure and arrival times at node $i$ on his ride $r$. A well-designed ride-sharing plan requires a seamless coordination among drivers and passengers, including the determination of (i) the ride-sharing plan $Z_{prij}$ and $Y_{pqrwij}$, and (ii) the associated departure times $D_{pri}$ and the arrival times $A_{pri}$ at each node.

A summary of the notations used in formulating the problem can be found in Table A1 in Online Appendix A.

## 4. Mathematical Formulation

In this section, we present a mixed integer program for the ride-sharing model with meeting points and return restriction. Given the complexity of the problem, we start illustrating the ride-sharing mechanism with only meeting points (denoted by RS-M) in Section 4.1. Assuming each participant has only one ride during the planning horizon, the RS-M offers a ride-sharing mechanism for such a single-ride setting. This assumption is relaxed in Sections 4.2 and 4.3. In Section 4.2, we describe the modeling construct for the return restrictions, which lays the groundwork for developing our analytical framework for the RS-M&R. Section 4.3 incorporates the return restrictions into the RS-M and presents the proposed mixed integer program for the RS-M&R.

### 4.1. Ride-Sharing with Meeting Points

In this section, we propose the ride-sharing model with meeting points for the single-trip problem. By using the RS-M, the company can determine (i) the optimal single-trip ride-sharing arrangement among the employees and (ii) the corresponding time schedule. The objective is to minimize the cost of the commuter and business traffic of a company, which consists of the cost incurred from vehicle miles and the costs of penalizing the efficiency losses. The efficiency losses are related to (i) arriving too early for appointments, (ii) the waiting time for transfers, and (iii) the inconvenience and potential risks associated with the number of transfers. Accordingly, the objective function in our formulation of the RS-M is given by (1). Each of the four terms has a weight attached:

$$\min \Big\{ \alpha_1 \sum_p \sum_{(i,j)\in SP_p} t_{ij} Z_{pij} + \alpha_2 \sum_p (l_p - A_{p,d_p}) + \alpha_3 \sum_p \Big( (A_{p,d_p} - D_{p,o_p}) - \sum_{i,j} t_{ij} x_{pij} \Big) + \alpha_4 \sum_p \sum_q \sum_i S_{pqi} \Big\}. \tag{1}$$

The RS-M is confined by two sets of constraints: (i) spatial constraints and (ii) capacity and time constraints.

*Spatial constraints.* The spatial constraints are imposed to find feasible matches among participants based on the spatial information (i.e., origins and destinations):

$$Y_{ppij} = 0 \quad \forall\, p \in P,\ i,j \in \mathcal{N}; \tag{2}$$

$$\sum_q Y_{pqij} + Z_{pij} = x_{pij} \quad \forall\, p \in P,\ i,j \in \mathcal{N}; \tag{3}$$

$$\sum_q \sum_i Y_{pqij} + \sum_k Z_{pjk} \le 1 \quad \forall\, p \in \mathcal{P},\ j \in \mathcal{N}; \tag{4}$$

$$Y_{pqij} \le Z_{qij} \quad \forall\, p,q \in \mathcal{P},\ i,j \in \mathcal{N}; \tag{5}$$

$$S_{pqj} \ge \sum_k Y_{pqjk} - \sum_i Y_{pqij} \quad \forall\, p,q \in \mathcal{P},\ j \in \mathcal{N}; \tag{6}$$

$$Z_{pij}, Y_{pqij}, S_{pqi} \in \{0,1\} \quad \forall\, p,q \in \mathcal{P},\ i,j \in \mathcal{N}. \tag{7}$$

Constraints (2) exclude the possibility of persons carpooling with themselves. Constraints (3) ensure that a person can either carpool with a driver or drive on his own on each arc $(i,j)$ he has to travel. Constraints (4) ensure that person $p$ cannot drive anymore for the rest of the trip after he leaves his car and rides with someone else. Constraints (5) ensure that person $p$ can ride with person $q$ from node $i$ to node $j$ only if $q$ drives from $i$ to $j$. Constraints (6) keep track of the nodes where the carpools start. Constraints (7) are domain constraints.

*Capacity and time constraints.* The capacity and time constraints are imposed to find feasible matches based on (i) the available passenger seats of the cars, and (ii) the time compatibilities of the participants:

$$\sum_p Y_{pqij} \le v_q \quad \forall\, q \in \mathcal{P},\ i,j \in \mathcal{N}; \tag{8}$$

$$A_{pj} = D_{pi} + t_{ij} \quad \forall\, p \in \mathcal{P},\ (i,j) \in SP_p; \tag{9}$$

$$A_{p,d_p} \le l_p \quad \forall\, p \in \mathcal{P}; \tag{10}$$

$$D_{p,o_p} \ge e_p \quad \forall\, p \in \mathcal{P}; \tag{11}$$

$$D_{pi} \ge A_{pi} \quad \forall\, p \in \mathcal{P},\ i \in \mathcal{N}; \tag{12}$$

$$D_{pi} - D_{qi} \le M(1 - Y_{pqij}) \quad \forall\, p,q \in \mathcal{P},\ i,j \in \mathcal{N}; \tag{13}$$

$$D_{pi} - D_{qi} \ge -M(1 - Y_{pqij}) \quad \forall\, p,q \in \mathcal{P},\ i,j \in \mathcal{N}; \tag{14}$$

$$(A_{p,d_p} - D_{p,o_p}) - \sum_{(i,j)\in SP_p} t_{ij} x_{pij} \le m_p \quad \forall\, p \in \mathcal{P}; \tag{15}$$

$$D_{pi}, A_{pi} \ge 0 \quad \forall\, p \in \mathcal{P},\ i \in \mathcal{N}. \tag{16}$$

Constraints (8) are capacity constraints for all the drivers. Constraints (9) calculate the arrival times of persons based on the associated departure times. Constraints (10) and (11) ensure that each person departs after the corresponding earliest departure time and arrives before the corresponding latest arrival time. Clearly, the departure time cannot be earlier than the arrival time at the same station, which is considered by Constraints (12). Constraints (13) and (14) ensure that the departure time of a driver equals the departure time of the passenger with whom the driver shares a ride. Constraints (15) prevent a person's waiting time during the trip being greater than his maximum waiting time. Constraints (16) are nonnegativity constraints.

### 4.2. Incorporating Return Restrictions

In this section, we extend the basic model to consider return restrictions. This extension does not lead to an additional objective. However, the return restrictions impose two extra conditions on the model; that is, a person can leave his car at a certain node and ride with someone else if and only if (i) he will pass through this node on a later ride and (ii) he is able to return via ride sharing. To this end, we introduce a new set $\mathcal{R}_p = \{1, 2, \ldots, n_p\}$, the elements of which are used as an indicator of the current ride of a person $p$. This enables us to cope with multiple rides per person in a certain period of time. We also introduce a new (dependent) binary variable $C_{pri}$ to keep track of the location where a person's car is parked at the end of each ride. The constraints concerning the return restriction are as follows:

$$C_{pri} \leq \sum_{w=r+1}^{n_p} \sum_j Z_{pwij} \quad \forall p \in \mathcal{P}, r \in \mathcal{R}_p, i \in \mathcal{N}; \tag{17}$$

$$C_{pri} \leq 1 - \sum_j Z_{prij} \quad \forall p \in \mathcal{P}, r \in \mathcal{R}_p, i \in \mathcal{N}; \tag{18}$$

$$C_{pri} \geq \sum_j Z_{prji} - \sum_k Z_{prik} \quad \forall p \in \mathcal{P}, r \in \mathcal{R}_p, i \in \mathcal{N}; \tag{19}$$

$$C_{pri} \geq C_{p,r-1,i} - \sum_j Z_{prij} \quad \forall p \in \mathcal{P}, 2 \leq r \leq n_p, i \in \mathcal{N}; \tag{20}$$

$$C_{pri} \leq \sum_j Z_{prji} + C_{p,r-1,i} \quad \forall p \in \mathcal{P}, 2 \leq r \leq n_p, i \in \mathcal{N}; \tag{21}$$

$$C_{pri} \in \{0,1\} \quad \forall p \in \mathcal{P}, r \in \mathcal{R}_p, i,j \in \mathcal{N}. \tag{22}$$

Constraints (17) construct the first condition of the return restriction mentioned at the beginning of Section 4.2, that is, if person $p$ does not drive from node $i$ to any other node on a later ride ($w > r$), then he cannot park his car at $i$ on ride $r$. Constraints (18) state that if person $p$ drives away from node $i$, then his car cannot be parked there. In contrast, if $p$ drives to node $i$ on ride $r$ but does not drive out of $i$ on the same ride, then the car is parked at $i$ at the end of ride $r$, which is ensured by Constraints (19). Constraints (20) synchronize the location of the car at the end of different rides; that is, the car is still parked at node $i$ if person $p$ parks his car at $i$ on ride $r$ and carpools on the entire ride $r+1$. Once the car is picked up from this node, Constraints (18) set the corresponding variable back to 0. Constraints (21) prevent a car being parked at node $i$ at the end of ride $r$ if the car was not parked there at the end of ride $r-1$ or driven to $i$ on ride $r$. Constraints (19)–(21) determine where a car is parked at the end of a ride. With this information, we can model the return restriction condition (ii) that a person is able to pick up the parked car; otherwise, the car cannot be parked. Constraints (22) are domain constraints for the newly defined variables.

### 4.3. Ride-Sharing Model with Meeting Points and Return Restrictions

Based on the RS-M, and the constraints we added to address return restrictions, we provide the complete mathematical formulation for the RS-M&R in this section. As we address later on, the complete formulation requires more changes besides adding the return constraints to the RS-M formulation:

$$\min \left\{ \alpha_1 \sum_p \sum_r \sum_{(i,j) \in SP_{pr}} t_{ij} Z_{prij} + \alpha_2 \sum_p \sum_r (l_p - D_{p,r,d_{pr}}) + \alpha_3 \sum_p \sum_r \left( (A_{p,r,d_{pr}} - D_{p,r,o_{pr}}) - \sum_{i,j} t_{ij} x_{prij} \right) + \alpha_4 \sum_p \sum_q \sum_r \sum_i S_{pqri} \right\} \tag{1'}$$

s.t.

$$Y_{pprwij} = 0 \quad \forall p \in \mathcal{P}; r, w \in \mathcal{R}_p; i, j \in \mathcal{N}; \tag{2'}$$

$$\sum_q \sum_w Y_{pqrwij} + Z_{prij} = x_{prij} \quad \forall p, r, i, j; \tag{3'}$$

$$\sum_q \sum_w \sum_i Y_{pqrwij} + \sum_k Z_{prjk} \leq 1 + C_{p,r-1,j} \quad \forall p \in \mathcal{P}; 2 \leq r \leq n_p; i, j; \tag{4'}$$

$$\sum_q \sum_w \sum_i Y_{pqrwij} + \sum_k Z_{p,r+1,j,k} \leq 1 + C_{prj} \quad j = d_{pr}, \forall p; 1 \leq r \leq n_p - 1; i, j; \tag{23}$$

$$Y_{pqrwij} \leq Z_{qwij} \quad \forall p, q \in \mathcal{P}; r \in \mathcal{R}_p; w \in \mathcal{R}_q; i, j; \tag{5'}$$

$$S_{pqrj} \geq \sum_w \sum_i Y_{pqrwij} - \sum_w \sum_k Y_{pqrwjk} \quad \forall p, q \in \mathcal{P}; r, j; \tag{6'}$$

$$\sum_p \sum_r Y_{pqrwij} \leq v_q \quad \forall q \in \mathcal{P}, w \in \mathcal{R}_q, i, j; \tag{8'}$$

$$A_{prj} = D_{pri} + t_{ij} \quad \forall p, r, i, j; \tag{9'}$$

$$A_{pri} \leq l_{pr} \quad i = d_{pr}, \forall p, r; \tag{10'}$$

$$D_{pri} \geq e_{pr} \quad i = o_{pr}, \forall p, r; \tag{11'}$$

$$D_{pri} \geq A_{pri} \quad \forall p, r, i; \tag{12'}$$

$$D_{pri} - D_{qwi} \leq M(1 - Y_{pqrwij}) \quad \forall p, q \in \mathcal{P}; r \in \mathcal{R}_p; w \in \mathcal{R}_q; i, j; \tag{13'}$$

$$D_{pri} - D_{qwi} \geq -M(1 - Y_{pqrwij}) \quad \forall p, q \in \mathcal{P}; r \in \mathcal{R}_p; w \in \mathcal{R}_q; i, j; \tag{14'}$$

$$(A_{p,r,d_{pr}} - D_{p,r,o_{pr}}) - \sum_{(i,j)} t_{ij} x_{prij} \leq m_{pr} \quad \forall p, r, (i,j) \in SP_{pr}; \tag{15'}$$

$$C_{pri} \leq \sum_{w=r+1}^{n_p} \sum_j Z_{pwij} \quad \forall p \in \mathcal{P}, 1 \leq r \leq n_p - 1, i \in \mathcal{N}; \tag{17}$$

$$C_{pri} \leq 1 - \sum_j Z_{prij} \quad \forall p, r, i; \tag{18}$$

$$C_{pri} \geq \sum_j Z_{prji} - \sum_k Z_{prik} \quad \forall p, r, i; \tag{19}$$

$$C_{pri} \geq C_{p,r-1,i} - \sum_j Z_{prij} \quad \forall p, r, i; \tag{20}$$

$$C_{pri} \leq \sum_j Z_{prji} + C_{p,r-1,i} \quad \forall p, 2 \leq r \leq n_p, i; \tag{21}$$

$$Z_{prij}, Y_{pqrwij}, S_{pqri}, C_{pri} \in \{0,1\} \quad \forall p, q \in \mathcal{P}; r \in \mathcal{R}_p; w \in \mathcal{R}_q; i, j \in \mathcal{N}; \tag{24}$$

$$D_{pri}, A_{pri} \geq 0 \quad \forall p \in \mathcal{P}, r \in \mathcal{R}_p, i \in \mathcal{N}. \tag{25}$$

The labels of the constraints link the constraints from RS-M&R with the constraints from RS-M, which we have already explained in Sections 4.1–4.2. Most of these constraints are simply modified by adding an additional dimension of $r$ in the notation. Here, we draw the reader's attention to Constraints (4′) and (23). Constraints (4′) are modified based on Constraints (4) to cope with return restrictions. As described in Section 4.1, Constraints (4) are to prevent a person from driving once he leaves his car to join a carpool. With the return restrictions, however, it is possible for him to return to the location where he leaves his car and drive again. This situation is considered in Constraints (4′). Constraints (23) consider the boundary situation at the destinations. These constraints ensure that person $p$ cannot carpool to his destination on ride $r$ and leave this node with his car on ride $r+1$, unless his car has been parked at this location.

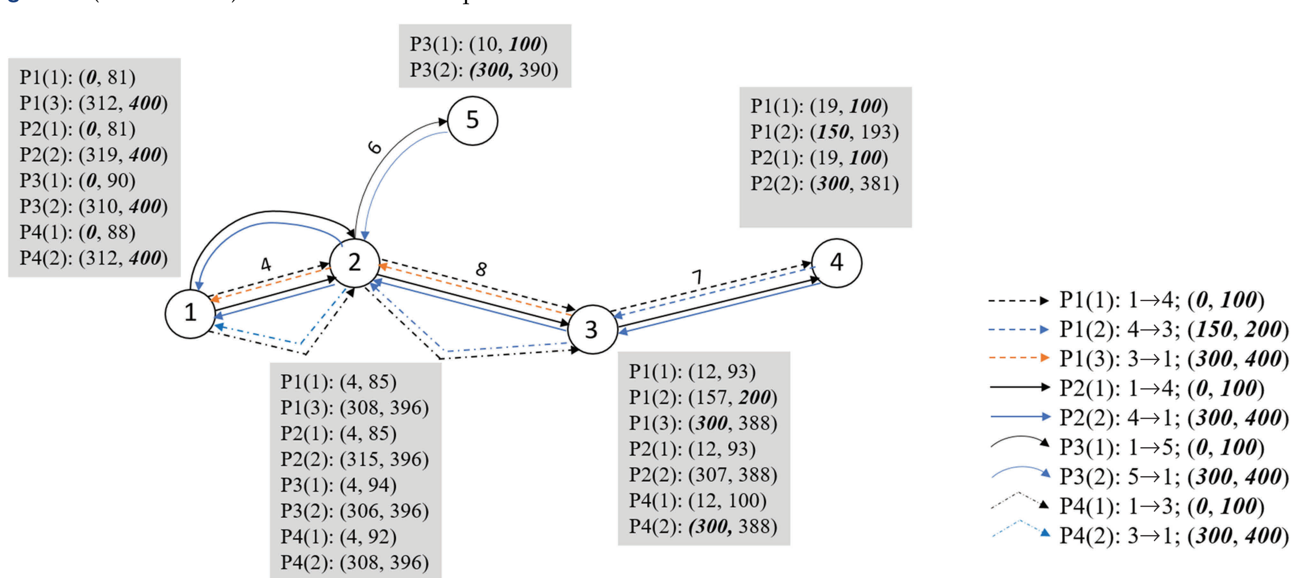
## 5. Heuristic Approach

When the numbers of persons and transfer points are large, the RS-M&R can become computationally prohibitive to solve. Moreover, when considering the return restrictions, it is not even an easy task to find a good solution to start with. In this section, we propose a constructive heuristic, which is based on the savings concept, to serve as a good starting point for improvement heuristics that can be used to solve large-scale problem instances of the RS-M&R that we may face in practice. For expository purposes, we assume that each of the participants has a car. However, this assumption can be easily relaxed to cover a more general setting where some of the participants do not have a car; see Agatz et al. (2011). Considering the flexibility that each participant has as being a driver or a rider, we make the following design choices to limit the number of role changes:

- Once a person is assigned to provide a ride to others, he will always be a driver.
- A person can leave his car at most once. In other words, once a person leaves his car (either at home or at a car parking point), he will be a rider until he picks up his car at the car parking point where he left his car. Then, he will become a driver for the remainder of his trip(s).

The basic idea of the heuristic is to use shared rides to satisfy the mobility demands of the participants who have the most ride-sharing potential as being riders. To this end, the heuristic greedily assigns other participants who can drive the aforementioned participants with high ride-sharing potential. We define participant $p$ as being covered by $q$ on arc $(i,j)$ when $q$ can drive $p$ on $(i,j)$. Correspondingly, the total distance that $q$ cannot drive $p$ is defined as the *uncovered distance of $p$ by $q$*, which is denoted by $u_{pq}$. A smaller $u_{pq}$ indicates a higher ride-sharing potential of $p$ by having only $q$ as the driver. We also define the *minimum uncovered distance of $p$* as the minimal value of the uncovered distances of $p$ by $q \neq p$ (i.e., $\min_{q\neq p} u_{pq}$). We further define the *uncovered distance of $p$* as the distance that remains uncovered after considering a subset of participants $q \neq p$ as potential drivers. Note that these variables are dynamic, and their values may thus change during the course of program execution. Instead of using "covered distance," we believe that the uncovered distance is a better indicator of the ride-sharing potential in our case. We will return to this issue in the next paragraph after introducing the concept of external mobility service. Figure 1 provides an illustrative example of how to compute the uncovered distance. Participant 1 (P1) has three trips (P1(1), P1(2), and P1(3)), traveling from 1 to 4 in the first one, from 4 to 3 in the second one, and from 3 to 1 in the third one. Participants 2, 3, and 4 (P2, P3, and P4) have two trips each, return trips from 1 to 4, 1 to 5, and 1 to 3, respectively. Each participant has a car with three passenger seats. Besides showing the trips for each participant, the legend also shows, in parentheses, the earliest

**Figure 1.** (Color online) An Illustrative Example of the Heuristic



departure time from the origin and the latest arrival time at the destination for each trip. For time intervals at nodes (i.e., the interval between a participant's earliest departure time and his or her latest arrival time at the node to arrive at the destination on time, shown in parentheses), the numbers in italic are inputs, and the rest are obtained via calculation. For instance, participant 4 can feasibly reach all the destinations by riding with 1, and hence the uncovered distance of 4 by 1 is 0 (i.e., $u_{41} = 0$). We use 3-tuples to store the information of uncovered distances in the heuristic. Thus, the uncovered distance of P4 by P1 is stored as (P4, P1, 0). We remark here that uncovered distances are not symmetric; for example, $u_{14} = 14 \neq u_{41}$.

The additional complexity from the return restrictions motivates us to consider the use of an external mobility service provider, such as a taxi, as a backup option for taking the unmatched riders who have left their cars. Although this is an extension of the RS-M&R problem presented before, we can still compare the performance of the heuristic with the ILP from Section 4.3 by setting the cost of an EMS sufficiently high. Sufficiently high in this setting means that a driver always prefers to use his own car when the ride-sharing plan includes the use of an EMS. The resulting optimal solution to the relaxed problem is then the same as the optimal solution to the RS-M&R. From an algorithmic standpoint, measuring one's ride-sharing potential based on uncovered distances can help to reduce the potential utilization of an EMS.

The remainder of this section is organized as follows. In Section 5.1, we introduce the observations that are used to determine candidates for meeting points and car parking points. In Section 5.2, we present our heuristic that iterates over the participants. To justify our design choices, we discuss alternative implementations in Section 5.3. These alternatives will be compared in Section 8 through extensive numerical experiments.

### 5.1. Determining Potential Locations for Meeting Points and Car Parking Points

Meeting points are the nodes where two or more persons start a shared ride. We use the concept of a time interval at a node as the possible time frame of being present at the node, given the earliest departure time, the latest arrival time, and the assignment of the shared rides. Our approach for determining potential meeting points depends on the following observations.

**Observation 1.** A node $i$ can be a *potential* meeting point for persons $p$ and $q$ if and only if (i) the intersection of the time intervals of $p$ and $q$ at $i$ is nonempty and (ii) $i$ is the starting point of a common arc in $SP_{pr}$ and $SP_{qw}$ for some $r$ and $w$.

Building on Observation 1, we derive Observation 2, which extends the concept of meeting points to the multiple-rider matchings.

**Observation 2.** A node $i$ can be a potential meeting point for a set of persons $Q \subseteq P$ if and only if $i$ is a potential meeting point for any subset of persons $Q' \subseteq Q$.

Our approach for determining potential car parking points depends on the following observation that comes on top of Observations 1 and 2.

**Observation 3.** Node $i$ is a *potential* car parking point for person $p$ if and only if (i) it is a potential meeting point for $p$ and, (ii) following the shortest paths, it will be visited at least twice by $p$ during the planning horizon.

According to Observation 3, a potential car parking point is a special type of potential meeting point where the role of a person can be changed from a driver to a rider. Note that home addresses are also potential car parking points according to Observation 3. Furthermore, note that Observation 3 also holds for the relaxed problem if we consider an EMS as a special participant of the system.

### 5.2. The Ride-Sharing Heuristic

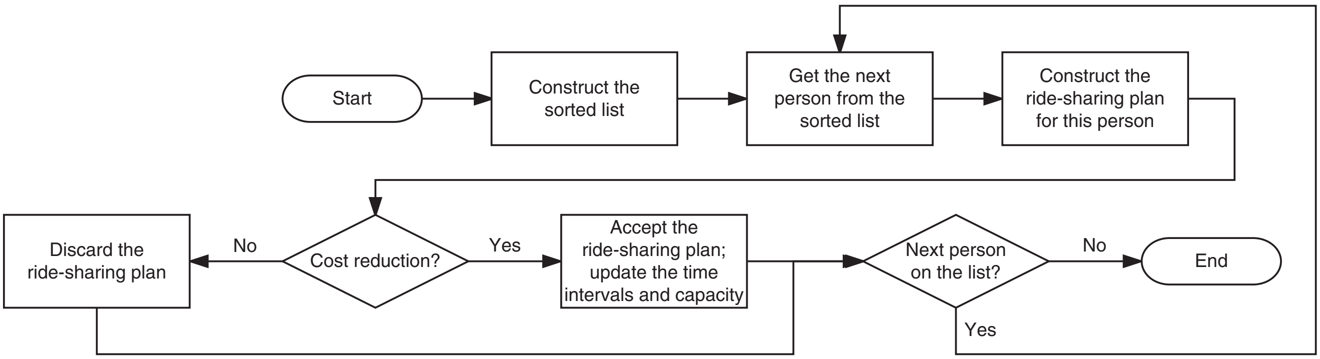
The ride-sharing heuristic that we propose is a constructive and iterative procedure. We construct a sorted list of participants according to their ride-sharing potential, which is measured by their *initial* minimum uncovered distances. Here, "initial" refers to the time we measure the minimum uncovered distance. A key feature of this heuristic is that the list is sorted only once, and will not change when the heuristic assigns drivers to riders. These changes in the minimum uncovered distance of a participant can happen when a potential driver of this participant is also assigned to other participants.

In each iteration, we select the next participant from the list as a potential rider and determine the ride-sharing plan for this person. This ride-sharing plan is accepted only if it leads to a reduction in total mobility cost. The overall structure of the heuristic is summarized in Figure 2. We will now further elaborate on the procedures of (i) constructing the sorted list and (ii) iteratively constructing the ride-sharing plan for each participant.

**5.2.1. Constructing the Sorted List.** The purpose of the list is to select the most promising participant who might be able to satisfy the return restrictions only by riding with others. The procedure involves three steps. We illustrate the steps using Figure 1.

***Step* 1.** For each participant $p$, we calculate the initial uncovered distance of $p$ by $q$ for all $q \neq p$. The results of the initial uncovered distances in the illustrative example are presented as 3-tuples: (P1, P2, 7), (P1, P3, 30), (P1, P4, 14), (P2, P1, 7), (P2, P3, 30), (P2, P4, 14), (P3, P1, 12), (P3, P2, 12), (P3, P4, 12), (P4, P1, 0), (P4, P2, 0), and (P4, P3, 16).

**Figure 2.** Overall Structure of the Heuristic




***Step* 2.** For each participant $p$, we sort these initial uncovered distances in nondescending order; that is, (P1, P2, 7), (P1, P4, 14), and (P1, P3, 30) for P1; (P2, P1, 7), (P2, P4, 14), and (P2, P3, 30) for P2; (P3, P1, 12), (P3, P2, 12), and (P3, P4, 12) for P3; and (P4, P1, 0), (P4, P2, 0), and (P4, P3, 16) for P4. Note that when there is a tie among the pairs, they are ranked lexicographically. Let $X_p$ denote the minimum initial uncovered distance of $p$. Thus, $X_1 = 7$, $X_2 = 7$, $X_3 = 12$, and $X_4 = 0$.

***Step* 3.** We sort the participants according to their minimum initial uncovered distance in nondescending order. The resulting sorted list in the illustrative example is P4, P1, P2, and P3.

As shown in Figure 3, we obtain a sorted list of potential riders (shown vertically in a rectangle). For each participant $p$ on this list, we also maintain a priority queue (shown horizontally) with respect to the minimum initial uncovered distance of $p$ by $q$ for all $q \neq p$.

**5.2.2. Constructing the Ride-Sharing Plan.** As shown in Figure 2, we construct the ride-sharing plan using an iterative procedure over the participants. In each iteration, we determine the ride-sharing plan for the next participant in the sorted list using a two-phase procedure. In Phase 1, we solve a variant of the set cover problem (SCP) and obtain a feasible ride-sharing plan by using an EMS to cover the uncovered distance in the solution. For initialization, this feasible solution can result in a higher total cost compared to the case without ride sharing. Phase 2 improves the ride-sharing plan using a subroutine to reduce the use of an EMS in the solution.

**Figure 3.** The Structure of a Sorted List

| | | | |
|---|---|---|---|
| P4 | (P4, P1, 0) | (P4, P2, 0) | (P4, P3, 16) |
| P1 | (P1, P2, 7) | (P1, P4, 14) | (P1, P3, 30) |
| P2 | (P2, P1, 7) | (P2, P4, 14) | (P2, P3, 30) |
| P3 | (P3, P1, 12) | (P3, P2, 12) | (P3, P4, 12) |

***Phase* 1. *Dynamic greedy cover algorithm.*** We start with a formal definition of the set cover problem. Then, we explain how to construct the ride-sharing problem as a variant of it. Finally, we outline the dynamic greedy cover algorithm (i.e., Algorithm 1), which is to construct a feasible solution for the RS-M&R with the use of an EMS.

The *set cover problem* is defined as follows. Let $\mathcal{U}$ be a universe set of $n$ elements $x_1, x_2, \ldots, x_n$, and let $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ be a collection of $m$ subsets of $\mathcal{U}$ such that $\bigcup_i S_i = \mathcal{U}$. The goal is to select as few subsets as possible from $\mathcal{S}$ such that their union covers $\mathcal{U}$. The literature proposes the greedy cover algorithm that selects sets according to the following rule: at each iteration, select the set that contains the largest number of uncovered elements. The greedy cover algorithm is shown to be the best-possible polynomial time approximation algorithm for the set cover problems (Feige 1998).

In our problem context, one universe set of elements is associated with each participant. Let $S_p(q)$ denote the set of arcs for all $r \in \mathcal{R}_p$ for which participant $p$ can ride with $q$ for $q \neq p$ (concerning spatiality, time, and capacity). The set $S_p(q)$ is equivalent to subset $S_i$ in the SCP. We also define $\mathcal{U}_p$ as a set of all arcs for which $p$ can ride with others for all $r \in \mathcal{R}_p$, and hence each arc of $p$ that can be covered by $q$ ($q \neq p$) is an element of $\mathcal{U}_p$. Clearly, $\mathcal{U}_p$ is a subset of $p$'s shortest paths during the planning horizon, and it is equivalent to $\mathcal{U}$ in the SCP.

There are two major differences between our problem and the set cover problem. First, a full cover is not always achievable, because the subsets $S_p(q)$ for all $q \neq p$ are dependent; that is, the selection of one subset may lead to infeasibility of other subsets. Thus, the goal in our problem context is to select as few subsets (i.e., $S_p(q)$) as possible, to cover the maximum distance for $p$ during the planning horizon. Having said that, the greedy strategy is still applicable to our problem. We select the subset that contains the minimum uncovered distance from the priority queue of $p$ until no further improvement can be achieved. The second difference is a continuation of the first one. The time interval of $p$ on an arc can be influenced by the previous assignments. Thus, the uncovered distance of a subset $S_p(q)$ increases over the greedy iterations in Algorithm 1. Consequently, we update the subsets and hence the priority queue of the current participant in each iteration.

Overall, the procedure includes three basic elements: (i) a priority queue pertaining to the minimum uncovered distance of the current potential rider, (ii) a monitor to record the arcs that are covered, and (iii) a monitor to record the total cost and the state (time intervals, capacity) of the current potential rider and the corresponding potential drivers. The overall procedure is shown in Algorithm 1.

**Algorithm 1** (The dynamic greedy cover algorithm)

**Input:** A potential rider $p$, and the corresponding priority queue $Q_p$ of potential drivers $q$
**Output:** A feasible ride-sharing plan for $p$
1: Initialize group $\mathcal{G} = \varnothing$
2: Initialize uncovered distance of $p$ = total distance of $p$
3: **while** {uncovered distance of $p > 0$ and priority queue $Q_p \neq \varnothing$} **do**
4: Remove the next 3-tuple $(p, q, u_{pq})$ from $Q_p$
5: Update the uncovered distance of $p$ by $q$ (i.e., $u_{pq}$)
6: **if** $u_{pq}$ has changed in line 5 **then**
7: Insert $(p, q, u_{pq})$ back to $Q_p$
8: **else**
9: Add $q$ to $\mathcal{G}$
10: Update the time intervals of the participants in $\mathcal{G}$
11: Update the capacity of $q$
12: Update the uncovered distance of $p$
13: **end if**
14: **end while**
15: Use EMS to cover all the uncovered arcs

In Algorithm 1, we make an assumption that if a participant takes the role of a rider, then this participant is not allowed to drive his or her own car at all. This simplification allows us to construct a "good" feasible ride-sharing plan for the current participant as a potential rider efficiently. This feasible solution is used as the input of the EMS reduction subroutine (see Algorithm 2) in Phase 2. This subroutine is designed to improve the solution by relaxing this assumption.

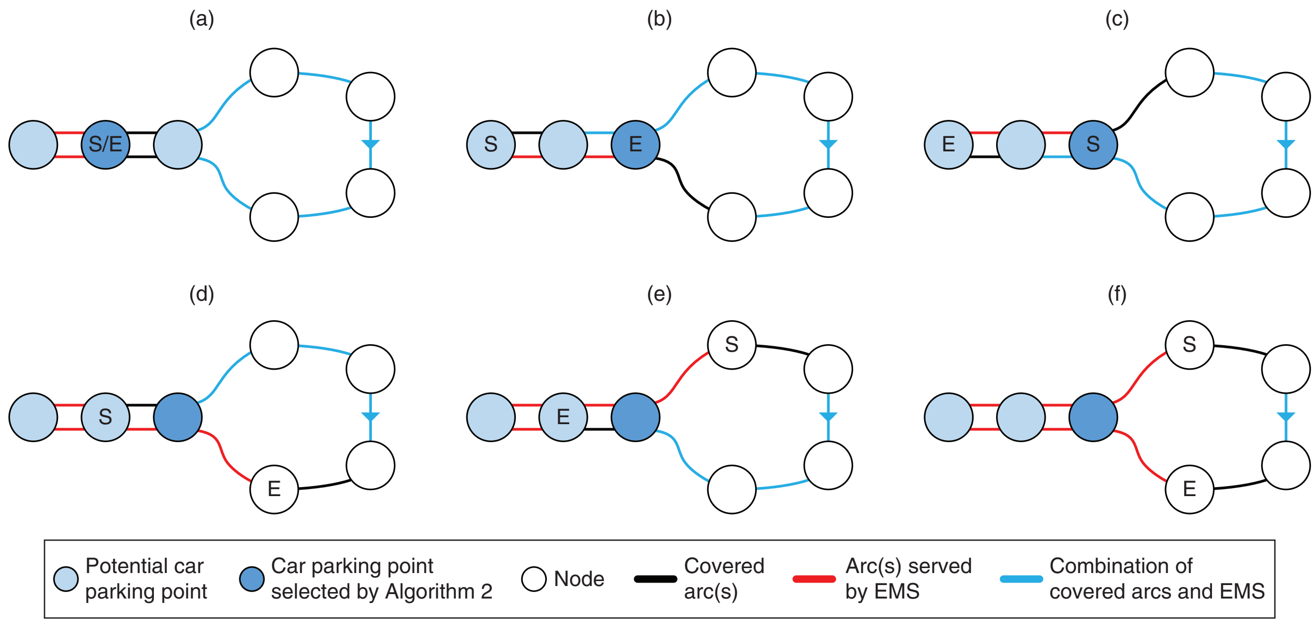
***Phase 2. EMS reduction subroutine.*** It is unlikely that the potential riders can be fully covered by their peers. Thus, the solution constructed by the dynamic greedy cover algorithm could include the use of an EMS. Assuming EMS is a much more costly option, we use a subroutine to minimize the use of EMS. The key of this subroutine is to determine the best location of the car parking point for the current potential rider $p$ so that the rides (i) before the first shared ride and (ii) after the last shared ride, which are initially served by EMS, can be substituted by self-driving to the largest extent to reduce the total cost. In other words, the goal of this subroutine is to reduce EMS use by slightly modifying the found ride-sharing plan. To this end, the subroutine is designed to minimize the total distance from the to-be-determined car parking point to where rider $p$ starts/stops riding with others.

Here we remind the reader that a potential car parking point is a node that will be visited at least twice during the planning horizon by a given participant. Overall, the procedure offers two ways of reducing the use of an EMS. First, as illustrated in Figure 4(a)–(c), when both the starting node (S) of the first shared ride and the ending node (E) of the last shared ride of participant $p$ are potential car parking point(s), the arcs initially served by an EMS from the first origin to the car parking point as well as from the car parking point to the final destination can be fully substituted by self-driving. Note that these two nodes can be either the same node (see Figure 4(a)) or two different nodes (see Figure 4, (b) and (c)). When they are different nodes, Algorithm 2 selects the node that leads to less use of an EMS as a car parking point. Second, as illustrated by Figure 4(d)–(f), when at least one of the two points (S or E) is not a potential car parking point, a subset of these arcs still need to be served by an EMS. We make two remarks pertaining to Figure 4. First, for the purpose of illustration, we consider only three potential car parking points in Figure 4. However, a participant can have more than three potential car parking points. Second, the ride-sharing plan resulting from Algorithm 1 may contain the use of an EMS between the first and the last shared rides, which we also omit from the figure to focus on the illustration of selecting the car parking point. The overall procedure is shown in Algorithm 2.

**Algorithm 2** (The EMS reduction subroutine)

**Input:** Output of Algorithm 1 for $p$
**Output:** The final ride-sharing plan of $p$
1: Find the starting node $v_s$ of the first shared ride
2: Find the ending node $v_e$ of the last shared ride
3: **if** both $v_s$ and $v_e$ are potential car parking points **then**
4: **if** $v_s = v_e$ **then**
5: Set it as selected car parking point of $p$
6: **else if** $v_s \neq v_e$ **then**
7: Remove the covered arc(s) in the shared ride(s) between them from the ride-sharing plan
8: Set the node that is further away from the home address as selected car parking point
9: **end if**

**Figure 4.** (Color online) An Illustrative Example of Selecting the Car Parking Point




10: **else**
11: Find the closest potential car parking point from either $v_s$ or $v_e$
12: Set it as selected car parking point of the current potential rider
13: **end if**
14: update the ride-sharing plan with self-driving from the first origin to the car parking point, and from the car parking point to the final destination

In Algorithm 2, we make two assumptions. First, we assume that EMS should be avoided at any cost (i.e., be replaced by changing the role of a rider to a driver) before the starting node of the first shared ride and/or after the ending node of the last shared ride as determined by Algorithm 1. This can be achieved by selecting the car parking point that results in the least amount of EMS use. We introduced this assumption to achieve comparable results from the ride-sharing heuristic and RS-M&R. Second, we assume that EMS use cannot be reduced for the remainder of the trips. However, eventually, only ride-sharing plans resulting in costs savings will be accepted by the heuristic (see Figure 2). Obviously, these two assumptions can be relaxed by changing the EMS reduction subroutine.

In closing this section, we make the following remark: Although the proposed solution approach evaluates the potential matches based only on mobility costs, other objectives are considered implicitly. As we greedily assign drivers to the potential rider based on the maximum distance each driver can cover, it potentially helps to reduce the number of transfers of the potential rider. To reduce the total deviation from the latest arrival time at the destinations, we set the departure time of each person at his latest possible departure time within the time interval of a feasible match. Although we do not consider it in our experiments, postprocessing may be used to improve the objective values of the total waiting time during the transfer and the total deviation from the latest arrival time at the destinations, by reoptimizing the time schedule of the given ride-sharing plan.

### 5.3. Discussion of Our Design Choices

In this section, we discuss our design choices and present a few alternatives. First, we consider two alternatives to study the way of selecting the current potential rider (i.e., the way we create the sorted list). By comparison with these two alternatives, which consider larger search spaces, our goal is to show that the design choices we make in Section 5.2 can deliver comparable results more efficiently. Furthermore, we study the added value of the EMS reduction subroutine.

In the proposed heuristic, the ride-sharing potential of a participant is measured by his or her initial minimum uncovered distance, which is an attribute that remains unchanged. This allows us to use a static sorted list to keep the ride-sharing potentials in order. However, the initial minimum uncovered distance does not fully reflect the real-time ride-sharing potential of a participant, since this potential highly depends on the previous assignments of his or her potential drivers. This dependency is caused by monotonically decreasing time-window lengths of the participants who have been assigned as drivers because of the previous ride-sharing assignments. As a result, the minimum uncovered distance of a potential rider who is still on the list may change, if this value results from pairing with a driver who has been assigned to riders in the previous iterations. Hence, a more sophisticated way of measuring a participant's ride-sharing potential is to use the actual minimum uncovered distance, which is computed based on the actual time feasibilities of all the participants. To this end, we construct a second priority queue to store this information, in which the priority is defined by the actual minimum uncovered distance of the potential riders. The resulting data structure becomes a two-dimensional priority queue: each row has the same structure as in Figure 3; the first element in each row is now ordered as a priority queue of the actual minimum uncovered distance of all potential riders. To maintain the structure of a priority queue, we follow the three steps (i.e., remove, update, insert) from Algorithm 1. When a cost-reducing ride-sharing plan is found for a rider (i.e., $p$ in Algorithms 1 and 2), the time intervals of the drivers who are assigned to the rider become more restricted. Therefore, the uncovered distance for each of these drivers with all the participants who have not been assigned any role needs to be recalculated and updated. The ranking of the unassigned participants' ride-sharing potential will be changed accordingly. The complexity of this implementation comes from the high dependency of the participants. For instance, a rider can hop with multiple drivers, and a driver can take multiple riders. The time intervals of all the directly and indirectly related participants need to be updated every time the status of any of them changes. As another benchmark, we also consider a *naïve* implementation, where the current potential rider is selected in a random manner (i.e., we use a randomized list instead of the sorted list). The heuristic is then restarted for a given number of times (random restarts), and we select the solution with the minimum total cost among all the repetitions. For convenience, in the remainder of this paper, we call the proposed heuristic with a static sorted list the 1*D greedy heuristic* and the alternative implementation with a two-dimensional priority queue the 2*D greedy heuristic*.

Finally, to test the efficiency of the EMS reduction subroutine (i.e., Phase 2), we also conduct a set of experiments where only Algorithm 1 is used.

## 6. An Illustrative Case: Significant BV

In this section, we apply the RS-M&R model on a case based on the commuter/business traffic of Significant BV, a consulting company that initiated this research. To this end, we use the rides driven by Significant personnel to evaluate the ride-sharing potential for this company. In Section 6.1, we present the case and discuss the characteristics of the data. Section 6.2 reports the results of applying the RS-M&R model to the rides driven by Significant personnel in March 2014.

### 6.1. Case Description

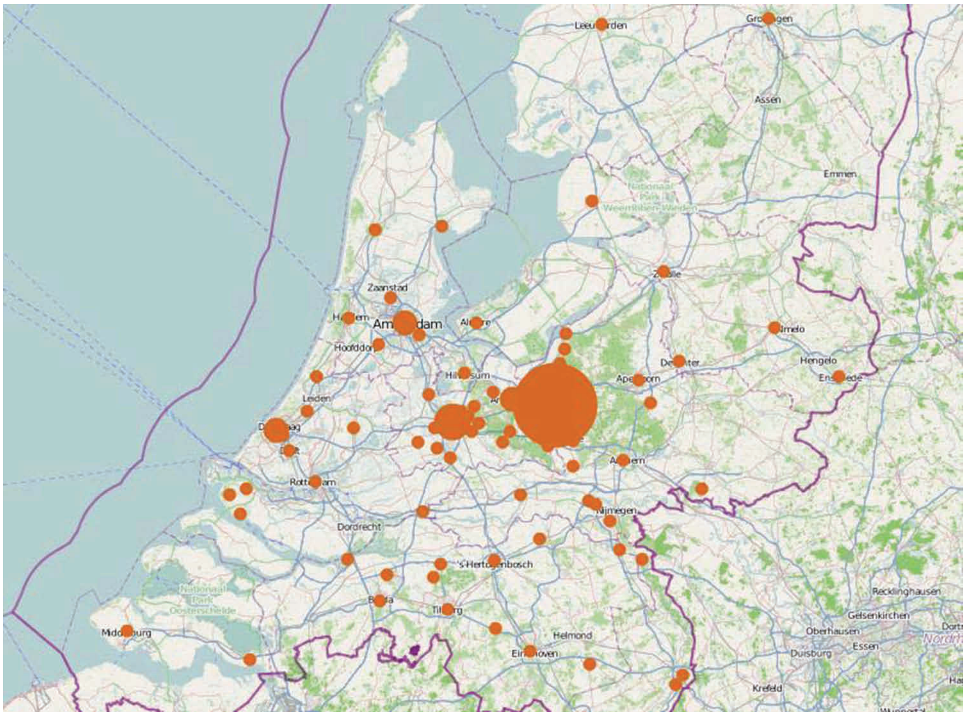
Significant BV is a Dutch consultancy and research firm, advising on organizational, operational, and procurement issues in the public domain. Currently, the company employs 42 consultants and researchers, all of whom are equipped with a company car. To reduce the costs and greenhouse gas emissions of the commuter/business traffic, Significant considers reducing the number of single-person trips by car via ride sharing among colleagues. However, employees lack insights into their colleagues' whereabouts, and thus ride sharing happens only occasionally for joint meetings. Our proposed automated matching methodology could offer a solution for this.

As input data, we collect data on Significant-related rides of all the employees in a period of four weeks using prepared forms filled out by the employees to keep track of their rides. The rides are defined on a city level; time schedules of the rides are defined on an hourly basis. The rides within the same city are excluded from the set of rides we use for the experiments. During these four weeks, 1,416 rides are considered. Eighty-seven percent of the these rides are from/to a home address; the other 13% are rides between the office in Barneveld and a client or those between two clients. Seventy-eight locations are visited in total. Figure 5 presents an overview of these locations; the size of a circle indicates the frequency of visits to this location. For instance, Barneveld is visited 430 times, Utrecht is visited 242 times, and Amsterdam is visited 97 times during the four weeks. Figure 6 shows the number of rides in different time intervals. Most of the rides are driven in the early morning or late afternoon, which is consistent with the high percentage of rides from/to home addresses. The main purpose of this illustrative case is to show the ride-sharing potential, even on such a small scale with only 42 participants and 71 trips per day on average.

The network used in this case study consists of the 78 cities shown in Figure 5 and the 358 carpool parking lots in the Netherlands.[1] The arc between each node pair represents the travel route chosen by Google Maps under the criterion of shortest driving time. To collect the travel time between each node pair, the Google Maps API and Bing Maps API are used. The Google Maps API is used to retrieve accurate coordinates of a location. The Bing Maps API is used to retrieve the travel time information for the 189,660 arcs.

Although most of the carpool parking lots are easily accessible from highways, a limited detour is still necessary. Therefore, every carpool parking lot that results in fewer than 5 minutes of additional driving distance if it is inserted into the route is assumed to be located on the shortest path. Similarly, we also assume that

**Figure 5.** (Color online) Origins and Destinations of Observed Trips



every city that leads to fewer than 10 minutes of additional driving distance if it is inserted into the route is located on the shortest path. The visiting order of the nodes along the shortest path is determined in a nondecreasing order according to the travel time between the origin and each node. This automated process of assigning nodes to the shortest paths may lead to undesirable outcomes that need to be adjusted (see Online Appendix B for detailed illustrations).

### 6.2. Experiments

In this section, we show the results of applying the RS-M&R model to the rides driven by Significant personnel in a period of four weeks. The value of the parameters are selected based on interviews with employees at Significant, which are summarized in Table 1.

Figure 7 shows that by sharing rides, Significant employees could have saved 7%–25% of the distances driven. The optimal solutions also show that among the 1,416 collected rides, 511 could have been saved by ride sharing, with 506 rides shared by two persons (i.e., one driver and one passenger), 2 rides shared by three persons, and 3 rides shared by four persons. In other words, two-person trips account for the vast majority of the shared rides. In addition, no one needs to wait during the trip according to the optimal ride-sharing plan.

Although the user group is very small, and the origins and destinations are rather scattered across the

**Figure 6.** (Color online) Distribution of the Persons Over the Time Intervals

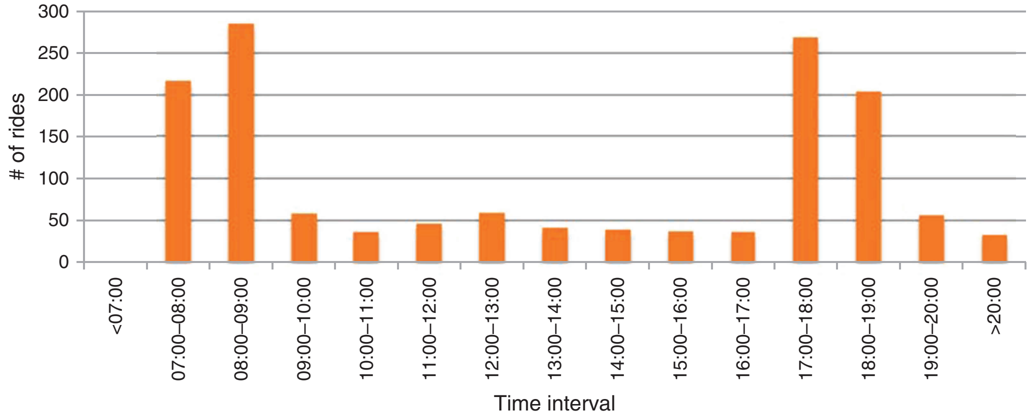


**Table 1.** Parameters of the Significant Case

| Parameter | Value |
|---|---|
| Max. waiting time at a transfer point | 10 min |
| Max. time deviation from the latest arrival time | 60 min |
| Capacity of passenger seats | 3 |
| Weight of the vehicle-miles $\alpha_1$ | 0.5 |
| Weight of the time deviation $\alpha_2$ | 0.01 |
| Weight of the waiting time $\alpha_3$ | 1 |
| Weight of the number of transfers $\alpha_4$ | 0.000001 |

Netherlands, the results show that Significant can still benefit from ride sharing by reducing the distance traveled by 7%–25%.

In the next section, the RS-M&R model is tested through extensive computational experiments on virtual networks. We aim at providing valuable insights into successfully implementing a ride-sharing service with the consideration of meeting points and return restrictions.

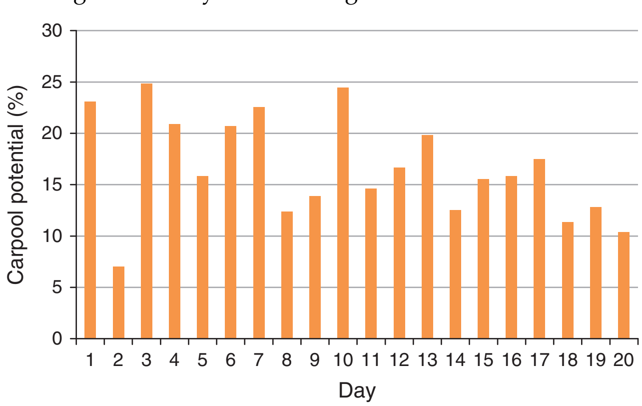
## 7. Experimental Settings

There are three factors that affect the possibility of ride sharing: the spatial network, the hot-spot density, and the participation density. Therefore, we design our problem instances to reflect these factors in three different settings, including the basic setting presented in Section 7.1, the high transfer point density (HTP) setting described in Section 7.2, and the high business traffic density setting (denoted by HBT setting) described in Section 7.3. Section 7.4 explains the method we use to compare the algorithmic performance, and Section 7.5 explains the experimental setting for investigating the benefits of using an EMS. The performance indicators are described in Section 7.6.

### 7.1. Basic Setting

Two different spatially distributed sets with 51 nodes generated from Solomon's (1987) 50 customer problem instances are considered: the scattered set (R101) and the clustered set (C101). Assuming a single-workplace problem, we retain the coordinates of the depot, which becomes the workplace, and of the customers, which are considered as locations for the cities and carpool places. Note that only cities are considered as potential home addresses, origins, and destinations for business trips, while carpool places are used only for transfers. We multiply the coordinates of these nodes by 3, resulting in an area of $210 \times 210$ kilometers (roughly the size of the Netherlands). The networks are generated based on the minimum spanning tree concept. To better reflect the connectivity of the Dutch intercity road network, we extend the minimum spanning tree by connecting those two nodes whose ratio between their Euclidean distance and the shortest distance on the current network is above a predetermined threshold (0.3 for the scattered set and 0.5 for the clustered set). The resulting graphs are depicted in Figure 8, where the arcs in red constitute the minimum spanning tree and the arcs in blue are added for better connectivity. The Euclidean distance is used to calculate the distances between the connected nodes. The average speed of the drivers is assumed to be 60 km/h.

**Figure 7.** (Color online) Percentages Saved of the Total Driving Distance by Ride Sharing

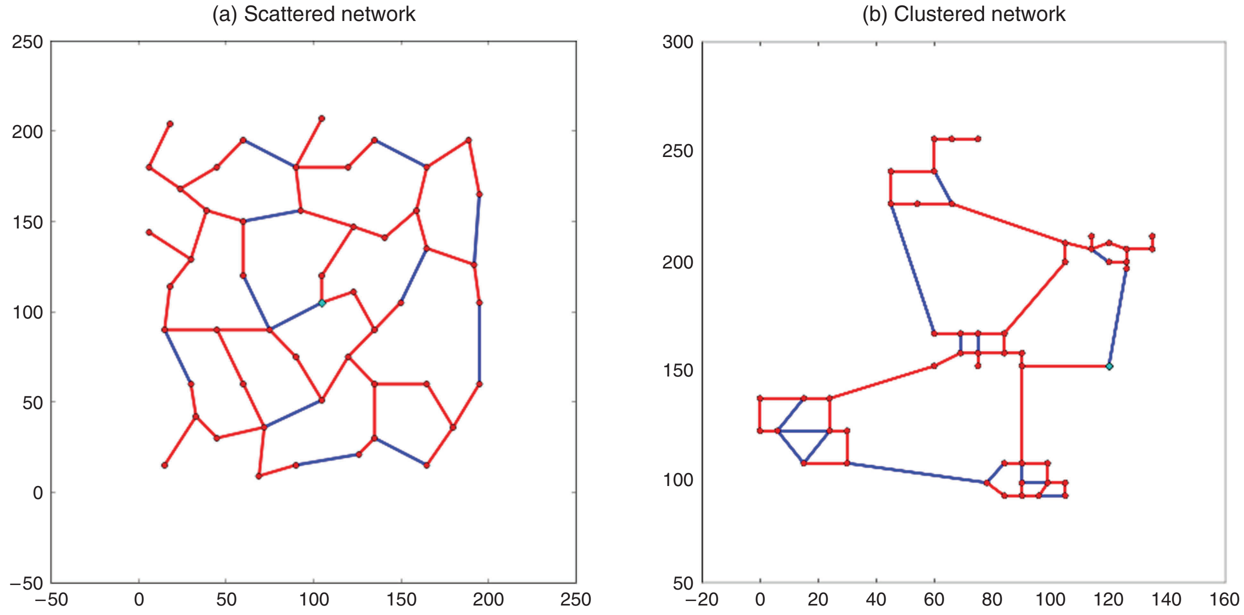



We consider three types of city nodes, the main difference of which is the associated probability of being an origin or a destination of a trip. Since all the trips belong to the commuter/business traffic of a company, the probability of the workplace being either the origin or the destination of these trips should be higher. We set it to 0.3 based on the data from Significant. The spatial distribution of the origins and destinations of individual trips depends on the socioeconomic statuses of the cities. For instance, when there are no significant demographic and socioeconomic differences within the region, it is possible to assume that the probability does not vary with the remaining 50 nodes. On the other hand, if such differences exist, some of the nodes may become hot spots for residential and business purposes, such as Utrecht, Amsterdam, and The Hague in the Significant case. It is natural to assign a higher probability to these city nodes. The number of hot spots is assumed to be one, two, or three, and the aggregate probability of any of the hot spots being the origin or the destination of a trip is 0.15—that is, with one hot spot, the probability of it being chosen is 0.15; with two hot spots, the probability of either one being chosen is 0.075; with three hot spots, the probability of any being chosen is 0.05. These observations lead to the following spatial distributions of trips:

- SU: Scatteredly distributed city/transfer nodes, uniformly distributed origins and destinations. We use the following probability function for this setting:

$$\Pr(n_{pr} = i) = \begin{cases} 0.3 & i \text{ is the workplace}, \\ \dfrac{0.7}{50} & \text{otherwise.} \end{cases}$$

**Figure 8.** (Color online) Networks




where $n_{pr}$ represents an end point of participant $p$'s ride $r$, either the origin $o_{pr}$ or the destination $d_{pr}$.

• SH: Scatteredly distributed city/transfer nodes with origin and destination hot spots. We use the following probability function for this setting:

$$\Pr(n_{pr} = i) = \begin{cases} 0.3 & i \text{ is the workplace,} \\ \dfrac{0.15}{h} & i \text{ is a hot spot,} \\ \dfrac{0.55}{50 - h} & \text{otherwise,} \end{cases}$$

where $h$ denotes the number of hot spots in an instance. We consider the settings with one, two, and three hot spots, which are denoted by SH1, SH2, and SH3, respectively.

• CU: Clustered city/transfer nodes, uniformly distributed origins and destinations, following the same probability function as in SU.

In total, we have 1,000 instances (20 scenarios, each including 50 instances) in the basic setting. Each scenario has a certain number of users and spatial distribution for the trips.

For each person, the number of trips was set to two, three, or four with probability 0.7, 0.2, or 0.1, respectively. The origins and destinations of the trips, except for the destination of the last trip, are generated based on the aforementioned probabilities in different settings. The destination of the last trip is the same as the origin of the first trip, representing the user's home location. Latest arrival times $l_{pr}$ for $r = 1$ are uniformly distributed between 480 and 540 (representing the time window between 8 a.m. and 9 a.m.); earliest departure times $e_{pr}$ for $r = r_{\max}$ are uniformly distributed between 1,020 and 1,140 (representing the time window between 5 p.m. and 7 p.m.). Latest arrival times of the remaining trips are uniformly distributed between 540 and 960 (representing the time window between 9 a.m. and 4 p.m.). Earliest arrival times of these trips are computed by subtracting a time slack from the corresponding latest arrival times. The time slack is set to 30. We assume that each person has a company car with capacity of three passenger seats. The weights $\alpha_1$, $\alpha_2$, $\alpha_3$, and $\alpha_4$ are the same as in the Significant case.

### 7.2. Effect of Transfer Point Density

To study the effect of transfer point density, we compare the basic setting with the HTP setting. The HTP setting contains half as many cities but the same number of transfer points as in the basic setting. The cities in the HTP setting form a subset of the transfer points. Only cities can be considered as locations for the origin or the destination of a trip, but both cities and transfer points can be used for transfers. Thus, a smaller number of cities represents an environment in which more trips may share the same origin or (and) destination. To allow for a fair comparison, the relative probability of a trip starting from (ending at) a hot spot compared to a regular city is the same as their counterpart in the basic setting. The set of 25 cities are randomly selected from the 50 nodes in each problem instance.

### 7.3. Effect of Business Trip Density

To access the impact of the business trip density on the performance of the ride-sharing system, we compare the basic setting with the HBT setting. In the HBT setting, the number of trips of a participant is randomly generated among two, three, or four with equal probabilities. Since two trips are always required for the commuter traffic, it implies that the number of business trips is zero, one, or two.

### 7.4. Algorithmic Comparison

We compare the computational performance of the proposed heuristic with different trip patterns and different numbers of participants. To show the efficiency of the proposed heuristic, we also consider a *naïve* heuristic for comparison. This naïve heuristic selects the current potential rider in a random manner, followed by Phase 2 of the proposed heuristic. To allow for a fair comparison with the ILP, we set the cost of EMS prohibitively high ($c_e = 1{,}000$) to avoid any use of it in the heuristics.

### 7.5. Benefits of the Potential Use of EMS

The return restriction may significantly reduce the success rate of ride sharing. It is likely that a participant has to travel with his own car because only a small part of his entire journey cannot be covered by any other driver. In this case, using an EMS for the small part may be beneficial. Thus, we want to investigate the potential benefit of allowing the use of an EMS as a backup option for taking the unmatched riders who have left their cars. To this end, we evaluate the mobility cost $c_e = 2$, which resembles the cost of using a taxi in the Netherlands, and $c_e = 0.5$, which resembles a situation when the community hires a mobility service provider to handle all the mobility demands. We also consider an intermediate cost setting $c_e = 1.25$, which aims at representing the cost of using a contracted EMS as an emergency backup. We use only the heuristic to solve these problem instances.

### 7.6. Performance Indicators

We evaluate and compare the solutions using the following metrics: (1) the percentage mileage savings; (2) the travel time increase, that is, the average waiting time per person during the entire day; and (3) the time deviation from the latest arrival time, that is, the average time difference (per person per ride) between a person's arrival time and the latest arrival time at the destination. In addition, we evaluate the efficiency of ride sharing using the following indicators: (4) the percentage car savings, that is, the number of saved cars as a fraction of the total number of participants; (5) the matching rate, that is, the vehicle-miles of the shared rides as a fraction of vehicle-miles when ride sharing is implemented; (6) the average number of riders in a shared ride; and (7) the average number of transfers needed for a matched rider to reach his destination. Note that we do not follow the commonly used definition of the matching rate, namely, the fraction of participants that are matched. This is because with the permission of transfers, a higher number of shared rides does not necessarily indicate a better performance of the ride-sharing system. Among these performance measures, the matching rate, the mileage savings, and the average number of riders indicate the success of ride sharing, while the others are related to the inconvenience of ride sharing. It is important to note that for the indicators of the success of ride sharing, a higher value is preferred. In contrast, for the indicators of the inconvenience of ride sharing, a lower value is preferred.

To conclude, Table 2 provides a summary of all experimental settings. Given these settings, we perform a number of experiments, as shown in Table 3.

**Table 2.** Instance Settings

| | |
|---|---|
| Basic setting | |
| No. of cities | 50 |
| Probability density of no. of trips | $\Pr(r=2)=0.7$, $\Pr(r=3)=0.2$, $\Pr(r=4)=0.1$ |
| Trip distribution | SU, SH3, SH2, SH1, CU |
| No. of participants | 40, 60, 80, 100 |
| HTP setting | |
| No. of cities | 25 |
| Probability density of no. of trips | $\Pr(r=2)=0.7$, $\Pr(r=3)=0.2$, $\Pr(r=4)=0.1$ |
| Trip pattern | SU, SH3, SH2, SH1, CU |
| No. of participants | 40, 60, 80, 100 |
| HBT setting | |
| No. of cities | 50 |
| Probability density of no. of trips | $\Pr(r=2)=\Pr(r=3)=\Pr(r=4)=\frac{1}{3}$ |
| Trip distribution | SU, SH3, SH2, SH1, CU |
| No. of participants | 40, 60, 80, 100 |

## 8. Numerical Results

Tests were performed on an Intel Core i5-6200U 2.30 GHz computer with 8 GB of RAM. The standard CPLEX 12.40 mixed integer programming solver in AIMMS was used for the ILP, and the heuristic was implemented in Java. In our results, we report averages over 50 instances. The corresponding ILPs ran with a time limit of two hours using AIMMS's default parameter settings. In most cases, the ILP found the optimal solution. For the instances in which the solver reached the time limit, they were terminated with a gap of at most 1.8%.

The remainder of this section is organized as follows. Section 8.1 studies the features of the RS-M&R, based on the optimal solutions obtained by the ILP in the basic setting. Sections 8.2 and 8.3 report the results of the HTP setting and the HBT setting, respectively, as compared to the results of the basic setting. In Section 8.4, we report the performance of the proposed heuristics. In Section 8.5, we report the benefits of introducing an EMS into the ride-sharing system.

**Table 3.** Characteristics of Instances in Each Experiment

| Experiment | Setting | $c_e$ | Solution method |
|---|---|---|---|
| Effect of trip patterns and no. of participants | Basic | 1,000 | ILP |
| Effect of transfer point density | HTP vs. basic | 1,000 | ILP |
| Effect of business trip density | HBT vs. basic | 1,000 | ILP |
| Heuristic performance | | | |
| Optimality gap | HTP and basic | 1,000 | ILP, 1D greedy, 2D greedy |
| Efficiency of Algorithm 2 | Basic | 1,000 | 1D greedy (Algs. 1 & 2, Alg. 1 only) |
| Selecting the next potential rider | Basic | 1,000 | 1D greedy, 2D greedy, naïve |
| Efficiency of randomization | Basic | 1,000 | Naïve(1, 5, 10 iterations) |
| Benefit of using an EMS | Basic | 0.5, 1.25, 2 | 1D greedy |

## 8.1. Effects of Trip Patterns and Participation Density

In this section, we study the effect of the number of participants in the system and their trip patterns on the system performance. Our experiments show that the average waiting time per person is negligible, less than one minute in all the tested scenarios. Thus, only the results of the other six indicators are presented. In general, we find that every performance indicator increases with the number of participants. The performance indicators that measure the success of ride sharing (i.e., the percentage mileage savings, the percentage car savings, and the matching rate) increase as the network concentration increases. However, the degree of network concentration has no significant impact on the performance measures that indicate the inconvenience of ride sharing (i.e., the time deviation from the latest arrival time, the average number of riders, and the average number of transfers). The detailed experimental results are shown in Tables 4 and 5. We use $\rho$ to represent the number of participants in the tables.

Table 4 presents the resulting performance indicators used in the objective function of the RS-M&R, including the percentage mileage savings (denoted by MileSav), the time deviation from the latest arrival time (denoted by TimeDev), and the percentage car savings (denoted by CarSav) in the basic setting. As expected, MileSav increases when more people participate. Comparing the cases in the scattered network (i.e., SU, SH3, SH2, and SH1), the MileSav is higher when the origins and the destinations of the trips are more concentrated. The reason is that it is more likely for two or even more persons to share a ride if they have the same origin or destination. The CU case offers the highest MileSav across all cases. Although the origins and destinations are evenly distributed across the cities, the clusters in the clustered network provide a natural concentration of them. Since the road connections between clusters are limited, it results in an increased number of shared rides among the trips between different clusters. We also see that ride sharing creates a promising opportunity to reduce the number of cars needed to satisfy the mobility demand (positive values of CarSav). This opportunity increases when more people participate. It might seem counterintuitive that when the number of participants is small (i.e., $\rho = 40$ and 60), SU leads to a slightly higher CarSav than SH3. The fact that SU has a higher percentage of the shared rides with two riders, whereas SH3 has a higher percentage of the shared rides with one and three riders, suggests that the number of cars saved by sharing rides with two passengers in SU is larger than the number of cars saved by sharing rides with one or three passengers in SH3 when the number of participants is rather small. Larger-scale ride-sharing systems may better realize the ride-sharing potential of the highly concentrated network/trip patterns.

We observe that TimeDev fluctuates in different trip patterns, because it is more closely related to the earliest departure times and the latest arrival times of the persons who are assigned to share a ride. However, we see that TimeDev increases with the number of participants. Such an increase in TimeDev results from an increased dependency of the participants. A feasible

**Table 4.** Basic Setting: Results of the ILP with Varying Number of Participants and Trip Patterns

| | MileSav (%) | | | | TimeDev (minutes) | | | | CarSav (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\rho = 40$ | 60 | 80 | 100 | 40 | 60 | 80 | 100 | 40 | 60 | 80 | 100 |
| SU | 8.09 | 12.27 | 15.05 | 18.28 | 2.53 | 3.52 | 4.34 | 5.28 | 5.60 | 9.63 | 10.83 | 13.72 |
| SH3 | 8.69 | 13.07 | 17.00 | 19.62 | 2.55 | 3.67 | 4.54 | 5.42 | 5.55 | 9.42 | 12.65 | 14.76 |
| SH2 | 10.14 | 14.41 | 17.74 | 20.82 | 2.90 | 3.78 | 4.69 | 5.56 | 6.65 | 10.63 | 12.85 | 16.72 |
| SH1 | 10.70 | 15.44 | 18.70 | 22.60 | 2.79 | 3.78 | 4.67 | 5.55 | 8.30 | 12.57 | 14.72 | 18.20 |
| CU | 13.76 | 18.83 | 23.99 | 27.84 | 3.43 | 4.48 | 5.62 | 6.72 | 6.80 | 10.60 | 13.72 | 15.68 |
| Average | 10.28 | 14.80 | 18.50 | 21.83 | 2.84 | 3.85 | 4.77 | 5.71 | 6.58 | 10.57 | 12.95 | 15.82 |

**Table 5.** Basic Setting: Ride-Sharing Efficiency with Varying Number of Participants and Trip Patterns

| | Matching rate (%) | | | | Avg no. of riders | | | | Avg no. of transfers | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\rho = 40$ | 60 | 80 | 100 | 40 | 60 | 80 | 100 | 40 | 60 | 80 | 100 |
| SU | 8.88 | 14.13 | 17.81 | 22.49 | 1.32 | 1.40 | 1.48 | 1.52 | 0.15 | 0.22 | 0.25 | 0.29 |
| SH3 | 9.71 | 15.17 | 20.68 | 24.53 | 1.31 | 1.38 | 1.52 | 1.59 | 0.16 | 0.21 | 0.26 | 0.29 |
| SH2 | 11.39 | 17.00 | 21.69 | 26.47 | 1.35 | 1.50 | 1.55 | 1.64 | 0.17 | 0.23 | 0.26 | 0.28 |
| SH1 | 12.25 | 18.42 | 23.20 | 29.39 | 1.38 | 1.55 | 1.59 | 1.68 | 0.14 | 0.20 | 0.25 | 0.28 |
| CU | 16.13 | 23.45 | 31.90 | 38.88 | 1.43 | 1.54 | 1.68 | 1.72 | 0.22 | 0.25 | 0.30 | 0.33 |
| Average | 11.67 | 17.63 | 23.06 | 28.35 | 1.36 | 1.47 | 1.56 | 1.63 | 0.17 | 0.22 | 0.26 | 0.29 |

matching requires the time coordination of all the corresponding participants, and thus some matched participants might need to execute their trips earlier to accommodate others.

Table 5 presents the results of the additional performance indicators, including the matching rate, the average number of riders, and the average number of transfers. These indicators also increase with the number of participants across all trip patterns. Furthermore, we observe a higher matching rate when the concentration of the trips is higher. Compared to the other three trip patterns in the scattered network, SH1 results in a higher value of the matching rate. However, the difference in the average number of riders is marginal, and there is no clear pattern in the number of transfers.

Interestingly, for a given number of participants, CU results in a higher MileSav and a higher matching rate, whereas SH1 leads to a higher CarSav (see Tables 4 and 5). The results suggest that the natural concentration in CU offers a bigger potential for the mileage savings, but this potential depends on the frequent usage of meeting points. Because of the evenly distributed origins and destinations, participants in CU more likely need to travel with their own car to a meeting point to benefit from ride sharing.

We now analyze and quantify the benefits of allowing multiple transfers in a ride-sharing system. Table 6 shows the percentages of matched riders with zero, one, two, three, and four transfers in the optimal solution for different trip patterns and different numbers of participants. For SU with 40 participants, 82.19% of the riders can ride with the same driver for the entire trip, 16.41% need one transfer, and the remaining 1.40% need two transfers. When there are 100 participants, the percentage of riders who do not need any transfer decreases by 15 percentage points; the numbers of riders with one transfer and with two or more transfers increase by 10 percentage points and 5 percentage points, respectively. Riders with a match involving transfers, as well as their drivers, need to plan and execute their trips more carefully to ensure that they arrive at the transfer points in time. This may be considered an inconvenience, but, on the other hand, the corresponding MileSav, CarSav, and matching rate improve significantly. This observation also holds for the other four trip patterns. We also see that CU leads to the maximum utilization of transfers. By comparing to the averages among the trip patterns of scatteredly distributed nodes (i.e., SU, SH3, SH2, and SH1), CU leads, on average, to more than 6 percentage points fewer riders with no transfer, but more than 4 percentage points more riders with one transfer, and more than 2 percentage points more riders with two or more transfers. This suggests that the possibility of transfers is the most beneficial in the clustered network. When the number of participants is 100 in CU, not only does the percentage of riders with no transfer drop, the percentage of riders with one transfer also drops. The riders with at least two transfers account for 8.68% of the matched riders. The limited connectivity between different city

**Table 6.** Average Percentages of Riders in Matches with Zero, One, Two, Three, and Four Transfers in the Basic Setting

| Trip patterns | Number of participants | No. of transfers for a rider (%) 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| SU | 40 | 82.19 | 16.41 | 1.40 | 0.00 | 0.00 |
| | 60 | 74.59 | 21.79 | 3.37 | 0.25 | 0.00 |
| | 80 | 71.18 | 23.75 | 4.72 | 0.28 | 0.08 |
| | 100 | 67.07 | 26.11 | 5.98 | 0.80 | 0.03 |
| SH3 | 40 | 79.98 | 19.06 | 0.84 | 0.12 | 0.00 |
| | 60 | 75.52 | 21.42 | 2.88 | 0.18 | 0.00 |
| | 80 | 70.37 | 24.49 | 4.54 | 0.60 | 0.00 |
| | 100 | 67.04 | 26.50 | 5.78 | 0.63 | 0.05 |
| SH2 | 40 | 80.51 | 17.71 | 1.78 | 0.00 | 0.00 |
| | 60 | 74.49 | 21.79 | 3.43 | 0.28 | 0.00 |
| | 80 | 69.97 | 24.61 | 4.66 | 0.72 | 0.04 |
| | 100 | 68.21 | 24.87 | 5.83 | 1.02 | 0.06 |
| SH1 | 40 | 83.44 | 15.17 | 1.18 | 0.21 | 0.00 |
| | 60 | 78.75 | 17.37 | 3.39 | 0.49 | 0.00 |
| | 80 | 72.37 | 22.72 | 4.12 | 0.79 | 0.00 |
| | 100 | 68.32 | 25.59 | 5.31 | 0.68 | 0.11 |
| CU | 40 | 73.52 | 22.57 | 3.52 | 0.29 | 0.10 |
| | 60 | 70.88 | 24.27 | 4.45 | 0.40 | 0.00 |
| | 80 | 65.28 | 27.27 | 6.35 | 0.94 | 0.17 |
| | 100 | 54.76 | 23.72 | 7.44 | 1.12 | 0.12 |

**Table 7.** Results for Having at Most Zero, One, and Two Transfers

| | MileSav (%) | | | CarSav (%) | | | Matching rate (%) | | | TimeDev | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| SU | 83.05 | 97.88 | 99.92 | 72.69 | 94.50 | 99.38 | 80.99 | 97.52 | 99.86 | 75.23 | 96.57 | 99.94 |
| SH3 | 82.71 | 98.24 | 99.99 | 70.06 | 93.10 | 93.74 | 80.55 | 97.96 | 99.99 | 77.92 | 97.34 | 99.99 |
| SH2 | 84.18 | 98.20 | 99.86 | 83.07 | 96.61 | 99.44 | 81.94 | 97.82 | 99.82 | 77.39 | 97.81 | 99.84 |
| SH1 | 86.72 | 98.51 | 99.87 | 86.48 | 97.30 | 99.20 | 84.58 | 98.21 | 99.84 | 80.61 | 97.72 | 99.77 |
| CU | 84.28 | 98.14 | 99.95 | 70.09 | 94.34 | 99.99 | 81.19 | 97.70 | 99.91 | 77.70 | 95.77 | 99.38 |
| Average | 84.19 | 98.19 | 99.92 | 76.48 | 95.17 | 98.35 | 81.83 | 97.84 | 99.89 | 77.77 | 97.04 | 99.79 |

*Notes.* The results are measured as proportions of the base case scenarios. They are based on the scenarios with 60 participants in the base case.

clusters in CU creates the trouble of having increased number of transfers, but, as expected, it also facilitates ride sharing.

To quantify the benefits of allowing multiple transfers, Table 7 compares the solution for the basic setting without any limit on the number of transfers to the solutions for the basic setting with at most zero, one, and two transfers. Only the results for the scenario with 60 participants are presented. The results are measured as a proportion of the base case scenarios; that is, 100% represents the results of the original model where there is no limitation on the number of transfers. We see that the absence of transfer results in a substantial reduction in the mileage savings, the number of cars saved, and the matching rate. On average, the mileage savings decrease noticeably, by more than 15%; the matching rate and the number of cars saved decrease even more. This implies that by limiting the number of transfers, the shared rides with relatively short distance are the most likely to be removed from the optimal ride-sharing plan. Accordingly, meeting points become more important with a limited number of transfers. This is because matched riders will probably need to drive themselves to the meeting points where the shared rides with longer shared distances start. As a consequence, the number of cars staying at the home addresses decreases. Furthermore, the average time deviation for participants decreases by more than 22%. This reduction is due to the decrease of the interdependency between drivers and matched riders. Perhaps more importantly, the results also suggest that most of the benefits of the proposed ride-sharing model can be achieved with at most one transfer. With single transfers, the three performance measures with respect to the success of ride sharing drop slightly, by no more than 5%. Thus, when taking into account the inconvenience associated with transfers, as well as the effort of coordinating the ride-sharing system, it is more desirable to allow only one transfer in the implementation.

In closing this section, we remark that although it is conceivable that the riders would prefer as few transfers as possible, recent technological developments, such as GPS and automated driving, might ease the hassles. Also in implementation environments with longer travel distances, multiple transfers are more likely to be acceptable.

### 8.2. Effect of Transfer Point Density

In this section, we report the impact of the number of transfer points on the system performance in Tables 8 and 9. The results are measured as absolute differences from the basic setting, where all the transfer points are considered as cities.

Table 8 presents the results of MileSav, TimeDev, and CarSav. For all the trip patterns, the HTP setting leads to higher potentials in reducing mileage and the number of cars needed. On average, we find that the benefit

**Table 8.** Effect of Transfer Point Density Regarding MileSav, TimeDev, and CarSav

| | MileSav (%) | | | | TimeDev (minutes) | | | | CarSav (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\rho = 40$ | 60 | 80 | 100 | 40 | 60 | 80 | 100 | 40 | 60 | 80 | 100 |
| SU | 1.83 | 2.80 | 3.36 | 4.65 | 0.42 | 0.52 | 0.58 | 0.69 | 1.30 | 1.57 | 3.87 | 4.52 |
| SH3 | 3.57 | 5.00 | 5.21 | 6.37 | 0.58 | 0.84 | 0.66 | 0.77 | 4.40 | 5.45 | 5.49 | 7.54 |
| SH2 | 4.04 | 3.96 | 6.15 | 6.84 | 0.43 | 0.57 | 0.66 | 0.81 | 5.65 | 5.77 | 7.94 | 7.88 |
| SH1 | 7.17 | 5.98 | 7.67 | 8.40 | 0.79 | 0.68 | 0.79 | 1.12 | 6.80 | 7.63 | 9.30 | 10.50 |
| CU | 2.44 | 3.87 | 4.10 | 3.45 | 0.30 | 0.82 | 0.81 | 0.52 | 1.70 | 2.97 | 5.61 | 5.50 |
| Average | 3.81 | 4.32 | 5.30 | 5.94 | 0.50 | 0.69 | 0.70 | 0.78 | 3.97 | 4.68 | 6.44 | 7.19 |

*Notes.* The results of MileSav and CarSav are measured as absolute percentage points differences from the base case scenarios. The results of TimeDev are measured as absolute differences.

**Table 9.** Effect of Transfer Point Density Regarding Matching Rate, Average Number of Riders, and Average Number of Transfers

| | Matching rate (%) | | | | Avg no. of riders | | | | Avg no. of transfers | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\rho = 40$ | 60 | 80 | 100 | 40 | 60 | 80 | 100 | 40 | 60 | 80 | 100 |
| SU | 2.29 | 3.86 | 4.89 | 7.38 | 0.00 | 0.02 | 0.02 | 0.05 | 0.01 | −0.02 | 0.00 | 0.00 |
| SH3 | 4.47 | 7.09 | 8.11 | 10.74 | 0.03 | 0.09 | 0.06 | 0.11 | −0.03 | −0.01 | −0.02 | −0.01 |
| SH2 | 5.43 | 5.78 | 9.99 | 11.96 | 0.03 | 0.05 | 0.12 | 0.08 | −0.02 | −0.02 | −0.02 | −0.02 |
| SH1 | 9.99 | 9.25 | 13.03 | 16.03 | 0.12 | 0.16 | 0.23 | 0.22 | 0.01 | −0.01 | −0.01 | −0.04 |
| CU | 3.48 | 6.20 | 7.34 | 6.90 | 0.02 | 0.04 | 0.02 | 0.04 | 0.00 | 0.00 | −0.01 | −0.01 |
| Average | 5.13 | 6.44 | 8.67 | 10.60 | 0.04 | 0.07 | 0.09 | 0.10 | −0.01 | −0.01 | −0.01 | −0.02 |

*Notes.* The results of matching rate are measured as absolute percentage points differences from the base case scenarios. The results of the average number of riders and average number of transfers are measured as absolute differences.

of introducing more transfer points, from the perspectives of MileSav and CarSav, increases in the number of participants. We also find that the HTP setting results in less than a one minute increase in TimeDev.

The results on the matching rate, the average number of riders, and the average number of transfers are presented in Table 9, measured as absolute differences from the basic setting. We find that the matching rate is more sensitive to the number of participants in the high transfer point density scenarios. This is shown by the fact that the overall/average difference in matching rate increases in the number of participants. The results also reveal that the matching rate is more sensitive to the concentration of trip distributions; that is, the difference in matching rate also increases as the trip pattern varies from SU to SH3 to SH2 to SH1. Compared to the basic setting, the performance of SU, SH3, and SH2 is much more distinguishable when the density of the transfer points is higher.

Surprisingly, we observe no big difference in the average number of transfers for different numbers of participants and different trip patterns. The increase in the average number of riders in a shared ride is at most 0.23. These results point to the fact that the major price of having higher matching rates (as well as MileSav and CarSav shown in Table 8) is the slight increase in TimeDev (i.e., less than one minute) among the participants. In other words, more concentrated origins and destinations are capable of creating higher ride-sharing potentials, with slightly more flexible and tolerant participants.

### 8.3. Effect of Business Trip Density

In this section, we report on the effect of the business traffic density on the performance of the system. We evaluate the indicators (i) from the perspectives of the system, (ii) for the commuter trips only, and (iii) for the business trips only. Only the results for the scenario with 60 participants in SU, SH1, and CU are presented, because the results of SU and SH1 provide the lower and upper bounds of the results of SH2 and SH3, respectively. The results are shown in Table 10.

As expected, the business traffic density has a positive impact on the mileage savings, the matching rate, and the average number of riders for the business traffic. More surprising is the fact that the increase of business trip density has a negative impact on the mileage savings, the matching rate, and the car occupancy in the commuter traffic. To some extent, this result may be a consequence of the choice of objective hierarchy: minimizing the waiting time is twice as important as minimizing the vehicle-miles. In the basic setting, because of the scarceness of business trips, we see a larger percentage of mixture between commuter traffic and business traffic within a shared ride, especially between the early return commuter traffic and the late business traffic. The potential drawbacks of these mixed matchings are the increase in waiting time and TimeDev. When the number of business trips increases, these participants are more likely to be better off by sharing rides within the business traffic. Implicitly, this results in a reduction of the number of participants that the commuter traffic can share rides with, and thus leads to reductions in MileSav, the matching rate, and car occupancy. Given that the commuter traffic is dominant over the business traffic in the basic setting, the diminishing commuter traffic density in the HBT setting also worsens these indicators from the system's standpoint. This is because the improvement gained from the business traffic is not enough to offset the rollback from the commuter traffic from the vehicle-miles perspective.

These findings stress the importance of having high participation from the viewpoint of the two subsystems, that is, the commuter traffic and the business traffic. Similar conclusions can be observed in the scenarios with 40, 80, and 100 participants.

### 8.4. Heuristics Performance

In this section, we report the algorithmic performance with different trip patterns and different numbers of participants. Within a time limit of two hours using AIMMS's default parameter settings, all the instances with 40 and 60 persons can be solved to optimality.

**Table 10.** Effects of Business Trip Density

| | SU | | | SH1 | | | CU | | |
|---|---|---|---|---|---|---|---|---|---|
| | Basic | HBP | % diff. | Basic | HBP | % diff. | Basic | HBP | % diff. |
| System | | | | | | | | | |
| MileSav (%) | 12.27 | 7.68 | −37.41 | 15.44 | 10.64 | −31.09 | 18.83 | 13.89 | −26.23 |
| Matching rate (%) | 14.13 | 8.36 | −40.84 | 18.42 | 11.99 | −34.91 | 23.45 | 16.24 | −30.75 |
| Avg no. of riders | 1.40 | 1.22 | −12.79 | 1.55 | 1.31 | −15.47 | 1.54 | 1.34 | −12.88 |
| TimeDev | 3.52 | 2.39 | −32.20 | 3.78 | 2.88 | −23.76 | 4.48 | 3.59 | −19.83 |
| Avg no. of transfers | 0.22 | 0.15 | −30.73 | 0.20 | 0.15 | −22.85 | 0.25 | 0.21 | −15.96 |
| Commuter | | | | | | | | | |
| MileSav (%) | 14.10 | 8.91 | −36.81 | 17.68 | 12.27 | −30.60 | 21.43 | 15.54 | −27.48 |
| Matching rate (%) | 16.62 | 9.85 | −40.73 | 21.68 | 14.12 | −34.87 | 27.62 | 18.57 | −32.77 |
| Avg no. of riders | 1.11 | 0.67 | −39.11 | 1.21 | 0.72 | −40.88 | 1.20 | 0.73 | −39.43 |
| TimeDev | 3.97 | 2.58 | −34.95 | 4.28 | 3.02 | −29.45 | 5.03 | 3.64 | −27.56 |
| Avg no. of transfers | 0.22 | 0.17 | −22.79 | 0.21 | 0.18 | −13.89 | 0.26 | 0.24 | −7.26 |
| Business | | | | | | | | | |
| MileSav (%) | 2.27 | 4.88 | 114.98 | 3.01 | 7.16 | 137.87 | 4.98 | 10.45 | 109.84 |
| Matching rate (%) | 2.37 | 5.18 | 118.57 | 3.17 | 7.79 | 145.74 | 5.41 | 11.85 | 119.04 |
| Avg no. of riders | 0.18 | 0.32 | 81.64 | 0.23 | 0.38 | 63.90 | 0.23 | 0.39 | 69.81 |
| TimeDev | 1.19 | 1.94 | 64.03 | 1.18 | 2.60 | 120.80 | 1.61 | 3.46 | 114.68 |
| Avg no. of transfers | 0.01 | 0.07 | 702.38 | 0.01 | 0.08 | 1,095.52 | 0.03 | 0.12 | 275.81 |

*Note.* The results are based on the scenarios with 60 participants in the base case.

In Table 11, we report the number of instances in every tested scenario where the solver reached the time limit. Overall, when the network is more concentrated, more problem instances reach the time limit before finding the optimal solution. The run times of the ILP and the 1D greedy and 2D greedy heuristics in solving the different scenarios are summarized in Table 12.

We see that the run time in solving the ILP increases superlinearly with respect to the number of participants. The scenarios with a higher concentration of origins and destinations, that is, the HTP settings and CU and SH1 in general, are much more difficult to solve. The run times of these scenarios are also more sensitive to an increase in the number of participants. In contrast, the run times of the greedy heuristics increase linearly with respect to the number of participants.

**Table 11.** Number of Instances Terminated Before Finding the Optimal Solution (Basic, HTP)

| Trip patterns | No. of cities | No. of participants | | | |
|---|---|---|---|---|---|
| | | $\rho = 40$ | 60 | 80 | 100 |
| SU | 50 | 0 | 0 | 0 | 0 |
| | 25 | 0 | 0 | 0 | 3 |
| SH3 | 50 | 0 | 0 | 0 | 0 |
| | 25 | 0 | 0 | 1 | 11 |
| SH2 | 50 | 0 | 0 | 0 | 5 |
| | 25 | 0 | 0 | 1 | 19 |
| SH1 | 50 | 0 | 0 | 1 | 4 |
| | 25 | 0 | 0 | 13 | 42 |
| CU | 50 | 0 | 0 | 3 | 25 |
| | 25 | 0 | 0 | 7 | 31 |

These run times are insensitive to the trip patterns: for different trip patterns on the scattered network (i.e., SU, SH3, SH2, SH1), the run times of the same number of participants are very similar. The run times of CU increase faster in the number of participants. As expected, the 2D greedy heuristic is slower than the 1D greedy heuristic. Although the differences are small, they increase in the number of participants.

Table 13 presents the optimality gaps of the 1D and 2D greedy heuristics. Overall, the performance of the two greedy heuristics is very close; neither consistently outperforms the other. Provided the 2D greedy heuristic requires a longer run time but results in a similar performance, we will focus our discussion on the 1D greedy heuristic. For every trip pattern, the optimality gaps of the 1D greedy heuristic increase linearly with respect to the number of participants. In either setting, the optimality gap increment of CU from 40 to 100 participants is approximately four percentage points higher than the optimality gap increment of SH1, which is the largest among SU, SH3, SH2, and SH1. The results suggest that when the trip pattern is more concentrated, the 1D greedy heuristic performs worse than the 2D greedy heuristic. This is because the potential for system optimization grows as the number of participants increases. Even so, the greedy heuristic can be used to obtain a good feasible ride-sharing plan that provides a much better result compared to the situation where ride sharing is not implemented.

Figure 9 provides further information regarding the implication of a "good solution" in practice. Specifically, we depict the mileage savings obtained by the 1D greedy heuristic. We see that the mileage savings

**Table 12.** Summary of the Run Time

| | | Run time (sec) | | | | | |
|---|---|---|---|---|---|---|---|
| | | Basic | | | HTP | | |
| Trip patterns | No. of participants | ILP | 1D greedy | 2D greedy | ILP | 1D greedy | 2D greedy |
| SU | 40 | 8.96 | 0.40 | 0.41 | 9.37 | 0.38 | 0.40 |
| | 60 | 20.39 | 0.51 | 0.53 | 24.30 | 0.51 | 0.54 |
| | 80 | 45.31 | 0.62 | 0.69 | 72.03 | 0.64 | 0.69 |
| | 100 | 137.92 | 0.76 | 0.84 | 940.17 | 0.77 | 0.88 |
| SH3 | 40 | 8.89 | 0.38 | 0.40 | 8.83 | 0.40 | 0.42 |
| | 60 | 19.97 | 0.51 | 0.53 | 30.25 | 0.52 | 0.56 |
| | 80 | 52.15 | 0.63 | 0.69 | 306.55 | 0.65 | 0.70 |
| | 100 | 310.99 | 0.76 | 0.84 | 2,245.20 | 0.79 | 0.91 |
| SH2 | 40 | 9.09 | 0.40 | 0.41 | 9.23 | 0.40 | 0.43 |
| | 60 | 23.46 | 0.51 | 0.56 | 23.40 | 0.53 | 0.57 |
| | 80 | 84.52 | 0.64 | 0.71 | 661.73 | 0.65 | 0.71 |
| | 100 | 1,194.26 | 0.76 | 0.85 | 3,419.80 | 0.81 | 0.91 |
| SH1 | 40 | 9.66 | 0.41 | 0.45 | 16.49 | 0.43 | 0.46 |
| | 60 | 25.27 | 0.52 | 0.56 | 98.07 | 0.55 | 0.59 |
| | 80 | 278.85 | 0.64 | 0.70 | 2,647.70 | 0.68 | 0.76 |
| | 100 | 1,161.22 | 0.77 | 0.86 | 6,456.44 | 0.82 | 0.95 |
| CU | 40 | 11.67 | 0.45 | 0.47 | 11.41 | 0.45 | 0.48 |
| | 60 | 38.13 | 0.57 | 0.62 | 103.09 | 0.58 | 0.62 |
| | 80 | 877.07 | 0.73 | 0.82 | 1,661.06 | 0.73 | 0.82 |
| | 100 | 4,843.45 | 0.89 | 1.01 | 5,600.56 | 0.89 | 1.04 |

increase in the number of participants. For 100 participants, the mileage savings range from 11.93% to 16.46% in the basic setting, and from 15.42% to 21.35% in the HTP setting, depending on the trip patterns. Without doubt, the reduction obtained by the proposed heuristic can be translated into a substantial cost reduction for the company. In addition, the 1D greedy heuristic obtains the largest mileage savings in CU in the basic setting, but, on the other hand, SH1 ranks first in the HTP setting. Interestingly, this observation is consistent with the optimal solutions of the ILP. This suggests that, to some extent, the 1D greedy heuristic can capture the dynamic of the RS-M&R problem on an ordinal level.

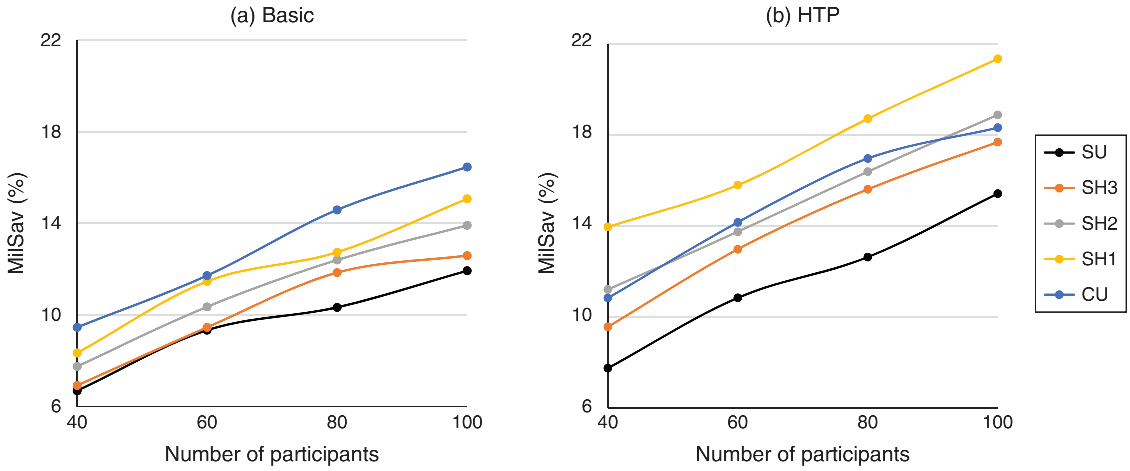
**Table 13.** Summary of the Optimality Gaps

| | | Optimality gap (%) | | | |
|---|---|---|---|---|---|
| | | Basic | | HTP | |
| Trip patterns | No. of participants | 1D greedy | 2D greedy | 1D greedy | 2D greedy |
| SU | 40 | 1.51 | 1.56 | 2.29 | 2.30 |
| | 60 | 3.33 | 3.33 | 4.92 | 4.97 |
| | 80 | 5.51 | 5.56 | 7.00 | 7.13 |
| | 100 | 7.77 | 7.77 | 9.71 | 10.08 |
| SH3 | 40 | 1.82 | 1.71 | 3.00 | 2.97 |
| | 60 | 4.10 | 4.24 | 6.13 | 6.18 |
| | 80 | 6.22 | 6.22 | 8.34 | 8.55 |
| | 100 | 8.76 | 8.74 | 11.17 | 11.46 |
| SH2 | 40 | 2.61 | 2.64 | 3.36 | 3.37 |
| | 60 | 4.58 | 4.64 | 5.59 | 5.56 |
| | 80 | 6.50 | 6.65 | 9.74 | 9.78 |
| | 100 | 8.69 | 8.78 | 11.97 | 12.34 |
| SH1 | 40 | 2.69 | 2.67 | 5.02 | 5.33 |
| | 60 | 4.65 | 4.57 | 7.25 | 7.37 |
| | 80 | 7.29 | 7.34 | 10.47 | 10.75 |
| | 100 | 9.79 | 9.96 | 14.08 | 14.38 |
| CU | 40 | 4.97 | 4.95 | 6.30 | 6.32 |
| | 60 | 8.73 | 8.80 | 10.94 | 10.82 |
| | 80 | 12.53 | 12.74 | 15.41 | 15.49 |
| | 100 | 15.85 | 15.78 | 18.93 | 19.30 |

Next, we examine the efficiency of the proposed alternative design choices, which include the EMS reduction subroutine and the naïve implementation with a varying number of restarts. We report the results, for 60 participants in SU, SH1, and CU, in Tables 14 and 15. As expected, the 1D greedy heuristic without the EMS reduction subroutine (i.e., using only Algorithm 1) takes less time to run. However, the differences are negligible. In negligible run time, the EMS reduction subroutine can effectively reduce the optimality gap by 3.65 percentage points in SU (from 6.98% to 3.33%), 3.54 percentage points in SH1 (from 8.19% to 4.65%), and 5.36% percentage points in CU (from 14.09% to 8.73%). Such improvements, to some extent, also imply the benefits of using meeting points in the proposed ride-sharing systems.

The naïve heuristic with a single iteration (with the EMS reduction subroutine) performs very closely to the 1D greedy heuristic. With five iterations, the naïve heuristic already outperforms the 1D greedy heuristic in the tested scenarios of 60 participants. The efficiency

**Figure 9.** (Color online) Mileage Savings Obtained by the 1D Greedy Heuristic




gain from increasing the number of iterations monotonically decreases. The results imply that with a limited number of participants, a smarter sorting does not have a big impact (1D and 2D greedy heuristics) on the algorithmic performance. Regarding run times, we see that the run times of the naïve implementation with one iteration are more than twice the run times of the 1D heuristic. This suggests that a more careful selection of potential riders may result in smaller chances of constructing the matches that have no cost reduction (recall that the matches with no cost reduction need to be discarded). However, the run time per iteration decreases in the number of iterations because of the constant time required for constructing the graph.

Based on the numerical results of the smaller-size instances, we find that the 1D greedy heuristic can obtain a good solution with the shortest run times, while the naïve heuristic can provide better solutions after a certain number of iterations. The recommendation between these two heuristics is not straightforward. This decision needs to be made based on the size of the problem instances and the time available to run the heuristics.

Finally, we discuss the behavior of these two heuristics on larger-size instances that could better reflect other real-world cases for large companies or communities of a considerable number of companies. In particular, we use the two heuristics to solve the problems instances of 500 and 1,000 participants. The results pertaining to mileage savings and run times are given in Tables 16 and 17, respectively.

As shown in Table 16, the mileage savings of 500 participants obtained by the 1D greedy heuristic vary from 27.27% (in SU) to 30.36% (in CU). For ride-sharing systems of 1,000 participants, it achieves up to 36.06% mileage savings. The naïve heuristic needs fewer than 5 iterations to acquire a better solution, except for the scenario of 1,000 participants in CU, where 10 iterations are required. The added value of using even more iterations is limited. Table 17 shows that it takes less than four seconds for the 1D greedy heuristic to solve the problem instances of 500 participants, and less than eight seconds to solve the problem instances of 1,000 participants. Roughly speaking, the naïve heuristic needs three times more run time to achieve the same mileage savings. However, the naïve heuristic has the potential to further improve the solutions with even more iterations.

To conclude, the EMS reduction subroutine is a cost-efficient method to improve the solution quality. Moreover, both the 1D greedy heuristic and the naïve heuristic with a small number of iterations are able to quickly provide good feasible solutions resulting in more than 15%, 27%, and 33% mileage savings for 100, 500, and 1,000 participants, respectively. The choice between these two solution methods, as well as the number

**Table 14.** Optimality Gaps (%) of Alternative Implementation Choices (60 Participants)

| Trip patterns | 1D greedy (Alg. 1 + Alg. 2) | 1D greedy (Alg. 1 only) | Naïve (1 ite.) | Naïve (5 ite.) | Naïve (10 ite.) |
|---|---|---|---|---|---|
| SU | 3.33 | 6.98 | 3.34 | 2.56 | 2.40 |
| SH1 | 4.65 | 8.19 | 4.65 | 3.40 | 3.07 |
| CU | 8.73 | 14.09 | 8.89 | 7.20 | 6.80 |

**Table 15.** Run Times in Seconds of Alternative Implementation Choices (60 Participants)

| Trip patterns | 1D greedy (Alg. 1 + Alg. 2) | 1D greedy (Alg. 1 only) | Naïve (1 ite.) | Naïve (5 ite.) | Naïve (10 ite.) |
|---|---|---|---|---|---|
| SU | 0.51 | 0.50 | 1.22 | 2.30 | 3.20 |
| SH1 | 0.52 | 0.50 | 1.22 | 2.36 | 3.27 |
| CU | 0.57 | 0.56 | 1.28 | 2.54 | 3.62 |

**Table 16.** Mileage Savings (%) of Larger-Size Problem Scenarios

| No. of participants | Trip patterns | 1D greedy | Naïve (1 ite.) | (5 ite.) | (10 ite.) | (15 ite.) | (20 ite.) |
|---|---|---|---|---|---|---|---|
| 500 | SU | 27.27 | 27.11 | 27.85 | 28.05 | 28.23 | 28.35 |
| | SH1 | 29.50 | 29.21 | 30.28 | 30.52 | 30.53 | 30.71 |
| | CU | 30.36 | 29.92 | 30.76 | 31.03 | 31.14 | 31.45 |
| 1,000 | SU | 33.40 | 33.06 | 33.87 | 33.92 | 34.13 | 34.20 |
| | SH1 | 35.65 | 35.34 | 35.98 | 36.16 | 36.25 | 36.30 |
| | CU | 36.06 | 35.22 | 35.83 | 36.10 | 36.31 | 36.39 |

**Table 17.** Run Times in Seconds of Larger-Size Problem Scenarios

| No. of participants | Trip patterns | 1D greedy | Naïve (1 ite.) | (5 ite.) | (10 ite.) | (15 ite.) | (20 ite.) |
|---|---|---|---|---|---|---|---|
| 500 | SU | 3.44 | 5.19 | 14.29 | 23.26 | 39.55 | 41.07 |
| | SH1 | 3.60 | 5.22 | 14.56 | 24.23 | 33.53 | 42.88 |
| | CU | 3.89 | 5.82 | 16.59 | 27.65 | 38.69 | 50.26 |
| 1,000 | SU | 6.75 | 8.81 | 31.33 | 58.90 | 88.42 | 114.09 |
| | SH1 | 7.22 | 9.53 | 33.12 | 62.56 | 92.01 | 120.54 |
| | CU | 7.80 | 11.57 | 37.93 | 71.67 | 115.44 | 139.79 |

of iterations for the naïve heuristic, should be made according to the decision maker's preferred solution quality and run time. Finally, the naïve heuristic with multiple iterations shows the potential of using an improvement heuristic.

### 8.5. Benefits of Using an EMS

In this section, we report the potential benefit of allowing the use of an EMS as a backup option for taking the unmatched riders who have left their cars. In principle, solutions obtained by the heuristics in the previous experiments in Section 8.4 could have used an EMS. However, we used a prohibitively high EMS cost to avoid any use of it. Table 18 presents the results of the total cost savings and the mileage savings. These results are obtained by the 1D greedy heuristic, measured as percentage deviation from the results with $c_e = 1{,}000$. As expected, the benefit of an EMS is negatively correlated with its cost. We also see that the benefit of an EMS is highest for the CU case and smallest for the SU case. This result is explained in the next paragraph by carefully examining the composition of the mobility cost.

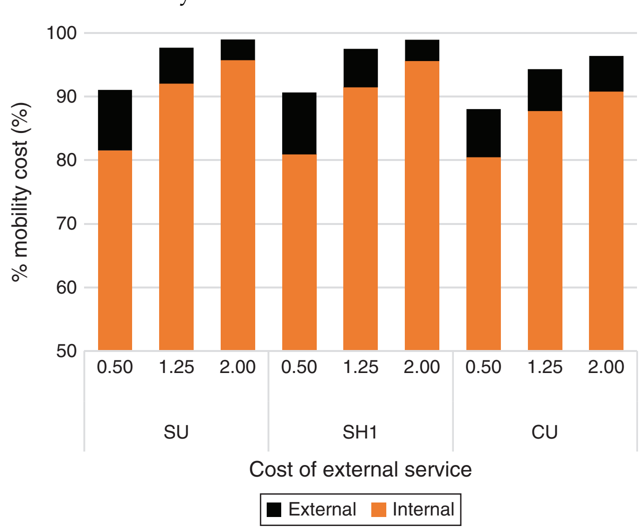
Figure 10 describes the composition of the mobility cost in more detail. The total costs are measured as percentages of the results with $c_e = 1{,}000$. The fraction of total external mobility cost decreases slightly in CU but decreases much more noticeably in the scattered network (i.e., SU and SH1); that is, the effect of a decrease in the vehicle-miles provided by an EMS is less than the impact of an increase in its unit cost. The results also show that the use of an EMS indeed reduces the total mobility costs. In particular, an EMS is most valuable in the clustered network, which points to the fact that

**Table 18.** Benefits of Using an EMS, Measured as a Percentage Deviation from the Results of $c_e$=1,000

| | Saved mobility cost (%) | | | Saved miles (%) | | |
|---|---|---|---|---|---|---|
| | $c_e = 0.5$ | $c_e = 1.25$ | $c_e = 2$ | $c_e = 0.5$ | $c_e = 1.25$ | $c_e = 2$ |
| SU | 8.92 | 2.30 | 1.01 | 88.09 | 55.84 | 34.08 |
| SH1 | 9.36 | 2.47 | 1.06 | 73.31 | 47.89 | 27.96 |
| CU | 11.96 | 5.66 | 3.63 | 91.78 | 73.95 | 60.10 |

*Note.* The results are based on the scenarios of 60 participants in the base case.

**Figure 10.** (Color online) Division of the Internal and External Mobility Costs



more participants in the CU fail to share rides because of a short disconnection along the journey.

## 9. Conclusion

In this paper, we consider the ride-sharing problem within a closed community of companies that agree to share the calendars of their employees. In particular, we introduce return restrictions in the ride-sharing problem to meet companies' need for their commuter/business traffic. We provide a general ILP for the ride-sharing problem with meeting points, return restrictions, and the option for riders to transfer between drivers.

Because of the high computational complexity of the problem, we propose a greedy heuristic. The basic idea of this heuristic is to guarantee the return restrictions of the participants who have the most ride-sharing potential, by greedily assigning other participants who can share rides with them taking the spatial, time, and capacity constraints into account. This heuristic can also be used to solve an extended problem that allows the use of an external mobility service as a backup option.

The proposed ILP is illustrated using a real-life ride-sharing problem of Significant BV, a Dutch consultancy and research firm. Although the user group is small, and the origins and destinations of the trips are rather scattered across the Netherlands, the results show that Significant can still benefit from ride sharing by a reduction of 7%–25% in vehicle miles.

The performance of the ILP and the heuristic are also tested on a large number of virtual problem instances. The numerical results show that ride sharing by using the proposed ILP can substantially improve a number of critical performance indicators, that is, the mileage savings (thus the mobility cost), the number of cars needed, the matching rate, and the occupancy rate of the cars. On average, the price that has to be paid to achieve these performance increases is minor: participants will (i) spend no more than one extra minute on a ride, (ii) arrive at their destination no more than eight minutes before the corresponding latest arrival time of a trip, and (iii) require, on average, fewer than 0.4 transfers to reach their destination. Overall, our results demonstrate that ride sharing creates more benefits when the origins and the destinations of the trips are more concentrated. Such a concentration relates to participation density, city density, hot-spot density, and city clusters. The results also suggest that it may be desirable to consider the use of an EMS. Finally, we show that the proposed heuristics, that is, the 1D greedy heuristic and the naïve heuristic with multiple iterations, are able to provide good feasible solutions quickly, which offers opportunities for the further development of improvement heuristics.

Future research can be done along three lines: (i) extending the mathematical model to incorporate the option of using an EMS, which allows us to investigate the true benefit of EMS; (ii) developing improvement heuristics to improve the performance in larger-scale problem instances; and (iii) extending the model by introducing an objective related to the resilience of the system, given that ride-sharing systems are much more integrated with the introduction of the return restriction.

## Endnote

[1] We retrieve the names and the coordinates of these parking lots from www.carpoolplein.nl.

## References

Agatz N, Erera A, Savelsbergh M, Wang X (2011) Dynamic ride-sharing: A simulation study in metro Atlanta. *Procedia - Soc. Behav. Sci.* 17(1):1450–1464.

Agatz N, Erera A, Savelsbergh M, Wang X (2012) Optimization for dynamic ride-sharing: A review. *Eur. J. Oper. Res.* 223(2):295–303.

Aissat K, Oulamara A (2014) A priori approach of real-time ridesharing problem with intermediate meeting locations. *J. Artificial Intelligence Soft Comput. Res.* 4(4):287–299.

Berbeglia G, Cordeau JF, Laporte G (2010) Dynamic pickup and delivery problems. *Eur. J. Oper. Res.* 202(1):8–15.

Bruck BP, Incerti V, Iori M, Vignoli M (2017) Minimizing $CO_2$ emissions in a practical daily carpooling problem. *Comput. Oper. Res.* 81:40–50.

Delhomme P, Gheorghiu A (2016) Comparing French carpoolers and non-carpoolers: Which factors contribute the most to carpooling? *Transportation Res. Part D: Transport Environ.* 42:1–15.

Drews F, Luxen D (2013) Multi-hop ride sharing. *Sixth Annual Sympos. Combin. Search* (AAAI Press, Palo Alto, CA).

European Environment Agency (2013) Evaluating 15 years of transport and environmental policy integration term 2015: Transport indicators tracking progress towards environmental targets in Europe. Accessed September 29, 2016, http://www.eea.europa.eu/publications/term-report-2015.

Feige U (1998) A threshold of ln *n* for approximating set cover. *J. ACM* 45(4):634–652.

Furuhata M, Dessouky M, Ordóñez F, Brunet M, Wang X, Koenig S (2013) Ridesharing: The state-of-the-art and future directions. *Transportation Res. Part B: Methodological* 57:28–46.

Gruebele P (2008) Interactive system for real time dynamic multi-hop carpooling. Report, Global Transport Knowledge Partnership, Geneva, Switzerland.

Herbawi W, Weber M (2011) Evolutionary multiobjective route planning in dynamic multi-hop ridesharing. Merz P, Hao JK, eds. *Evolutionary Computation in Combinatorial Optimization*. Lecture Notes in Computer Science, Vol. 6622 (Springer, Berlin), 84–95.

Herbawi W, Weber M (2012) The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm. 2012 *IEEE Congress Evolutionary Comput.* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), 1–8.

Hosni H, Naoum-Sawaya J, Artail H (2014) The shared-taxi problem: Formulation and solution methods. *Transportation Res. Part B: Methodological* 70:303–318.

International Energy Agency (2005) Saving oil in a hurry: Measures for rapid demand restraint in transport. Accessed September 29, 2016, https://www.iea.org/publications/freepublications/publication/savingoil.pdf.

International Transport Forum (2015) A new paradigm for urban mobility: How fleets of shared vehicles can end the car dependency of cities. Accessed September 29, 2016, http://www.itf-oecd.org/sites/default/files/docs/cop-pdf-03.pdf.

Kuntzky K, Wittke S, Herrmann C (2013) Car and ride sharing concept as a product service system—Simulation as a tool to reduce environmental impacts. *The Philosopher's Stone for Sustainability* (Springer, Berlin), 381–386.

Lee A, Savelsbergh M (2015) Dynamic ridesharing: Is there a role for dedicated drivers? *Transportation Res. Part B: Methodological* 81:483–497.

Mahmoudi M, Zhou X (2016) Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Res. Part B: Methodological* 89:19–42.

Naoum-Sawaya J, Cogill R, Ghaddar B, Sajja S, Shorten R, Taheri N, Tommasi P, et al. (2015) Stochastic optimization approach for the car placement problem in ridesharing systems. *Transportation Res. Part B: Methodological* 80:173–184.

Savelsbergh M, Van Woensel T (2016) 50th anniversary invited article city logistics: Challenges and opportunities. *Transportation Sci.* 50(2):579–590.

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35(2): 254–265.

Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2015) The benefits of meeting points in ride-sharing systems. *Transportation Res. Part B: Methodological* 82:36–53.

Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2016) Making dynamic ride-sharing work: The impact of driver and rider flexibility. *Transportation Res. Part E: Logist. Transportation Rev.* 91: 190–207.

Stradling SG (2007) Determinants of car dependence. Garling T, Steg L, eds. *Threats from Car Traffic to the Quality of Urban Life: Problems, Causes, Solutions* (Emerald Publishing, Bingley, UK), 187–204.

van Dender K, Clever M (2013) Recent trends in car usage in advanced economies—Slower growth ahead? Summary and conclusions. International Transport Forum Discussion Papers, No. 2013/09, OECD Publishing, Paris.

Wang X, Dessouky M, Ordonez F (2016) A pickup and delivery problem for ridesharing considering congestion. *Transportation Lett.* 8(5):259–269.

Winter S, Nittel S (2006) Ad hoc shared-ride trip planning by mobile geosensor networks. *Internat. J. Geographical Inform. Sci.* 20(8): 899–916.