

# A Partition-Based Match Making Algorithm for Dynamic Ridesharing

Dominik Pelzer, Jiajian Xiao, Daniel Zehe, Michael H. Lees, Alois C. Knoll, *Member, IEEE*, and Heiko Aydt

**Abstract**—Ridesharing offers the opportunity to make more efficient use of vehicles while preserving the benefits of individual mobility. Presenting ridesharing as a viable option for commuters, however, requires minimizing certain inconvenience factors. One of these factors includes detours which result from picking up and dropping off additional passengers. This paper proposes a method which aims to best utilize ridesharing potential while keeping detours below a specific limit. The method specifically targets ridesharing systems on a very large scale and with a high degree of dynamics which are difficult to address using classical approaches known from operations research. For this purpose, the road network is divided into distinct partitions which define the search space for ride matches. The size and shape of the partitions depend on the topology of the road network as well as on two free parameters. This allows optimizing the partitioning with regard to sharing potential utilization and inconvenience minimization. Match making is ultimately performed using an agent-based approach. As a case study, the algorithm is applied to investigate the potential for taxi sharing in Singapore. This is done by considering about 110 000 daily trips and allowing up to two sharing partners. The outcome shows that the number of trips could be reduced by 42% resulting in a daily mileage savings of 230 000 km. It is further shown that the presented approach exceeds the mileage savings achieved by a greedy heuristic by 6% while requiring 30% lower computational efforts.

**Index Terms**—Network partitioning, optimization algorithm, dynamic ridesharing, taxi sharing, multi-agent systems.

## I. INTRODUCTION

**I**N recent years, urbanization and economic growth have led to a rapid increase in the demand for mobility causing many urban transportation systems to operate at their capacity limits. As the potential for scaling-up the infrastructure is limited and subject to spatial and economic constraints, solutions for a more efficient utilization of the existing resources is required.

Manuscript received March 6, 2014; revised July 5, 2014 and December 3, 2014; accepted February 27, 2015. This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence and Technological Enterprise (CREATE) program. The Associate Editor for this paper was B. De Schutter.

D. Pelzer, J. Xiao, D. Zehe, and H. Aydt are with the Modeling and Optimization of Architectures and Infrastructure Group, TUM CREATE, Singapore 138602 (e-mail: dominik.pelzer@tum-create.edu.sg; jiajian.xiao@tum-create.edu.sg; daniel.zehe@tum-create.edu.sg; heiko.aydt@tum-create.edu.sg).

M. H. Lees is with the Section Computational Science of the Faculty of Science, University of Amsterdam, 1012 Amsterdam, The Netherlands (e-mail: M.H.Lees@uva.nl).

A. C. Knoll is with the Department of Informatics, Technische Universität München, 80333 Munich, Germany (e-mail: knoll@in.tum.de).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2015.2413453

One possibility for mitigating the strain on urban infrastructures while preserving the benefits of individual mobility is the introduction of mobility services which leverage on the growing availability of real-time information. An example of such individualized mobility services is real-time ridesharing which includes different variants such as peer-to-peer ride sharing, taxi sharing and smart on-demand buses.

One key challenge common to all variants of real-time ridesharing is to maximize the capacity utilization of vehicles while minimizing the inconvenience resulting from detours. As discussed in Section II, especially for the case of large-scale systems with dynamic transportation demand and vehicle availability, practically implementable methods leading to good results have not yet been established. Therefore, in Section III, an approach is presented which addresses this issue by partitioning the road network into regions that satisfy certain inconvenience constraints. The partitioning reduces the size of the search space in which the match making algorithm described in Section IV attempts to find matches. Two free parameters involved in the partitioning process allow optimizing the partitioning with regard to maximizing the outcome of the match making procedure as described in Section V.

As an application, this method is employed in Section VI for quantifying the potential for taxi sharing in Singapore. This application scenario is motivated by the fact that despite the large number of taxis in many megacities (28 000 in Singapore in 2013 [1]), supply bottlenecks during peak hours tend to lead to significantly reduced service quality. Given the availability of efficient match making algorithms, these shortfalls could be mitigated if passengers with overlapping routes would make use of the possibility to share rides.

To assess the performance of the partition-based approach, Section VII provides a comparison to a greedy heuristic building on the results of the Singapore case study. In Section VIII, the results are discussed and conclusions as well as an outlook on future work are ultimately given in Section IX.

## II. RELATED WORK

Dynamic ridesharing systems are related to a class of problems known as *vehicle routing problems* (VRP). These problems exhibit a large variety of different characteristics and have been subject to intensive research in recent decades.

### A. General Context

A number of taxonomies and reviews have attempted to develop classifications of the various types of problems. A

taxonomic review on the VRP was published by Eksioglu *et al.* [2] which on one hand provides an overview of previous classifications [3]–[9] and on the other suggests a comprehensive categorization scheme aiming to achieve complete coverage for all varieties of VRPs. Review papers on VRPs were published by Laporte [10] who gives an overview of exact algorithms, heuristics and metaheuristics, by Vidal *et al.* [11] whose focus are heuristics and by Pillac *et al.* [12] who investigate *dynamic* VRPs (D-VRP). Surveys on the class of *pickup and delivery problems* (PDP) are provided by Berbeglia *et al.* [13], [14], Savelsbergh *et al.* [15] as well as Parragh *et al.* [16], [17]. PDPs are problems where goods need to be picked up at a certain location and be delivered to another one which makes them conceptually related to ridesharing problems. This is particularly the case for the dynamic variant of the problem where transportation requests are generated over time in contrast to the static case where the entire problem is known beforehand. Methods addressing these dynamics can be found in [18]–[24].

A generalization of PDPs are *dial-a-ride problems* (DARP) which include additional inconvenience constraints that need to be considered when transporting passengers rather than goods [25]. DARPs consist of creating vehicle routes and schedules for passengers with a specific origin-destination pair and certain requirements regarding pickup and drop off time. Their objective is to minimize the transportation costs under a set of constraints. The formulation of the optimization problem can take a large variety of different forms, depending on what attributes are considered relevant for the quality of service. These attributes can for example be waiting time, maximum ride time or the difference between actual and desired arrival time. A comprehensive review on quality of service aspects and a categorization of previous work can be found in [26]. Just as PDPs, DARPs can either have a *static* or a *dynamic* character. Furthermore, it can be distinguished between a simple *single-vehicle* DARP and the more common *multi-vehicle* DARP [25]. Exact solutions based on dynamic programming for small instances of the static single-vehicle DARP had been developed early [27], [28] and were adapted to the dynamic variant of the problem as well [27]. Significantly more research was carried out on variants of the multi-vehicle DARP. For the static variant, starting from the solution proposed by Jaw *et al.* [29], a great number of heuristics have been developed which are explained in further detail in [16], [17], [25]. The dynamic multi-vehicle DARP was addressed by a smaller number of studies, including the work on heuristics in [30]–[32] where instances of up to a several hundred users could be solved. It, however, needs to be mentioned that most so-called dynamic DARPs are only dynamic in a sense that in addition to a static case a fraction of requests may occur or be canceled during operation time. Purely dynamic problems, in contrast, are comparably new and primarily arise from the increasing ubiquity of smart mobile devices [33]. With regard to DARPs, in [25] it is concluded that the number of solution techniques for the static problem is fairly abundant with methods able to solve instances with several hundred users. In contrast, dynamic problems remain hard to address and are therefore considered to deserve greater research efforts.

## B. Ridesharing Systems

Despite certain similarities, dynamic ridesharing systems differ from conventional DARPs in various aspects as pointed out in Agatz *et al.* [34]. They consider one distinguishing factor to be the supply of drivers and vehicles who in the typical DARP belong to a company while they are independent entities in a ridesharing system. This causes additional dynamics since the appearance of drivers is subject to uncertainty. Furthermore, new and more heterogeneous constraints may arise from individual preferences of different drivers. Also, vehicles in ride-share systems are not operated from depots but have unique origin-destination pairs so that routing decisions need to be evaluated as deviations from a driver's direct path. Methods developed for DARPs are therefore not universally applicable to ridesharing systems.

Surveys specifically on ridesharing were recently published by Agatz *et al.* [34] with a focus on optimization techniques and Furuhata *et al.* [35] who adopt a more practical perspective. Furuhata introduces a classification framework for ridesharing services which includes business functions such as pricing strategies and payment methods as well as issues of trust and privacy. While they identify a number of soft factors that pose challenges to the practical implementation of ridesharing concepts, their view on technical issues concludes that central research questions lie in the development of real-time bidding agents which implement automated rideshare exchange markets. This issue has recently been addressed by research presented in [36] and [37] which will be briefly discussed at a later stage.

With regard to optimization, Agatz *et al.* arrive at two main conclusions which are well in accordance with most findings on other types of VRPs mentioned above. The first aspect is that existing approaches may not be capable of solving realistic-size instances which is why a clear need for fast and practically implementable approaches is identified. A second aspect concerns scalability issues of existing centralized approaches. In this regard, it is concluded that centralized methods may not be feasible for addressing ridesharing problems in larger urban areas. Therefore, a need for decentralized approaches which reduce the size of the search space is derived. Addressing these two aspects as well as the challenges arising from the dynamics of the problem are the main contributions of the approach presented in this paper.

## C. Categories and Examples of Ridesharing Systems

Ridesharing problems can be classified according to a number of different criteria. According to Agatz *et al.*, problem variants can be distinguished by the number of drivers and riders involved. This leads to four different categories, encompassing *single rider—single driver*, *single rider—multiple driver*, *multiple rider—single driver* and *multiple rider—multiple driver* assignments. While the approach presented in this paper is not generally limited to a particular category, the case study in Section VI presents a single rider—single driver assignment. This is, in fact, an assignment problem which is discussed in greater detail in the review conducted in [38].

Another classification is described by Yan *et al.* [39] who distinguish three types of models based on the deviation of passengers' routes. These models are 1) the *many origins to one destination* (MOOD) model, 2) the *one origin to many destinations* (OOMD) model, and 3) the *many origins to many destinations* (MOMD) model. In addition, a fourth basic model can be defined which is 4) the *one origin to one destination* (OOOD) model.

There have been a number of commercial applications implementing the OOOD model. Most of these applications operated in a way that users could either create new rides by providing their trip details and wait for others to join or choose to join an existing trip themselves. One crucial limitation of these implementations was that the match making had to be conducted almost manually thus limiting their attractivity. An improvement in this sense is the vHike system presented by Stach [40] which transmits information via Bluetooth to potentially matchable people nearby.

Due to their greater flexibility, implementations of the MOMD model have undergone a recent increase in popularity. An example for an MOMD model can be found in [41] where a discrete event model is used to identify the best match and route by minimizing the deviations from desired departure and arrival times. Other examples in this context are startups like Sidecar,<sup>1</sup> Flinc,<sup>2</sup> and Carma<sup>3</sup> which facilitate dynamic ridesharing using location-aware mobile devices.

As for examples of an OOMD model, Chen [42] presents a system in *vehicular ad-hoc networks* (VANET) which aims to optimize fuel-savings. One characteristic of this system which establishes a relation to the method presented in this work is that it considers the road network topology by conducting match making based on road network partitioning. This means that the road network is divided into sub-areas in the form of a grid and the system matches passengers not only traveling to the same destination sub-area, but also to all sub-areas which share the borders with the destination sub-area.

Car pooling applications which are often organized to facilitate the commute to work are implementations of the MOOD model. An example for both an exact and a heuristic algorithm addressing this problem can be found in [43]. There is also some work which combines OOMD and MOOD models. For example, Tao [44] implements a taxi sharing system where the match making phase consists of enumerating all possible combinations from one origin to all destinations. The combination which leads to the shortest path is then chosen as the solution.

Gidofalvi and Pedersen [45] use a mathematical approach to solve the match making problem. Given a set of sharing requests, a maximum taxi-share size  $k$  and a minimum savings requirement, their algorithm is aimed at finding combinations of passengers which yield the maximum financial benefit. The algorithm first tries to find the best combination of sharing partners in a group of  $k$  for every request, where 'best' is defined as sharing the longest common route. For all these combinations, the algorithm further tries to find the one with maximum overall

savings. Since taxi fares are simply approximated by using the linear distance between two locations, this may, however, lead to sub-optimal results in practice. This is because cost typically depends not only on the linear distance but also on the time traveled, which is highly traffic dependent.

d'Orey [46] presents a taxi sharing algorithm with two parts, namely a customer algorithm and a taxi algorithm. The customer algorithm receives user-defined parameters such as ride time or distance and sends the information to a selected range of taxis. The taxi algorithm then calculates the optimal route with minimum distance.

Geisberger *et al.* [47] describe a fast detour computation algorithm to minimize the additional travel distance resulting from sharing a ride. Once a sharing request is generated, it is compared to all existing requests to find the combination with the shortest detour. This is done by performing a routing calculation whenever a new sharing request appears. Since the algorithm is aimed at finding an optimal solution, the required exhaustive search through all possible solutions may become a bottleneck in case of a large search space though.

#### D. Agent-Based Approaches

While most approaches are implemented as classical centralized optimization problems, a way of better accounting for the dynamics of ridesharing systems may be the use of agent-based methods as also employed in this work. One example of an agent-based approach was investigated by Mes *et al.* [48] who compare an agent-based vehicle scheduling technique based on *Vickrey auctions* with a classical look-ahead heuristic. In their simulation study with about 20 automated guided vehicles, they demonstrate that the multi-agent approach is less sensitive to fluctuations of demand and supply and that it achieves higher service levels than classical heuristics, particularly in the case of greater dynamics. With regard to agent-based approaches, the question arises how limited information provided to individual agents influences the overall system performance. This issue is addressed by Winter and Nittel [49] who consider a system where autonomous agents are matched based on short-range communication. In this context it is shown that a system built on local communication does not perform significantly worse than a system which has access to global information.

A frequently encountered issue regarding ridesharing systems is the question of fair pricing. In [36], Kamar and Horwitz present a system where prices are negotiated according to a Vickrey auction mechanism. This system dynamically creates shared plans and calculates fair splits of payments. A drawback, however, is that users do not have any influence on their assignment which may limit the system's attractiveness to travelers. Also, its suitability for large-scale problems has not yet been investigated.

The first drawback is resolved by the approach of Kleiner *et al.* [37], who propose a dynamic ridesharing system based on auctions with a second-price scheme. In contrast to the approach presented by Kamar and Horwitz, this system is able to consider individual user preferences regarding sharing partners and personal valuations for specific rides. By tuning the maximum allowed detour, the system allows to tradeoff the

<sup>1</sup><http://www.side.cr>

<sup>2</sup><http://www.finc.de>

<sup>3</sup><http://www.carmacarpool.com>

minimization of total vehicle kilometers with the probability of successful matches and thus facilitates achieving a critical mass. Drawbacks of the concept are that it has only been implemented as a single rider—single driver assignment and that investigations in a large-scale system are still pending.

### E. Partitioning Approaches

One common problem of most existing ride matching techniques is their lack of scalability. Resolving this issue requires approaches to either parallelize and balance the computing workload or to reduce the search space.

Early attempts to reduce the size of the problem were conducted in the context of PDPs. One example for this approach is the work by Bodin and Sexton [50] who formed clusters of users to be served by the same vehicle. This idea was further developed by Dumas *et al.* [51] and later by Desrosiers *et al.* [52], Ioachim *et al.* [53] and Borndörfer *et al.* [54] who used *mini-clusters* of users to be served within the same time frame in the same area.

Another way of reducing the problem size is to partition the road network into different regions. A number of approaches using this principle to balance computing workload is discussed by Möhring *et al.* [55] who compare several partitioning schemes with regard to speeding up the Dijkstra algorithm. These schemes partition the road network into grid, Quad-tree, Kd-tree and METIS [56] structures with the aim of balancing the number of vertices or edges in each partition. Another partition-based approach for PDPs is presented in [22]. They define a route as a sequence of 3-D boxes with the rectangular base representing the geographical area and the height defining the time span at which the zone is serviced by a vehicle. New pickups or drop offs are added to an existing zone if they are spatially and temporally close thus expanding the size of the zone. Once a zone exceeds a predefined size, it is split into two. In [24], the fuzzy C-means (FCM) method presented in [57] is employed to divide an area into multiple, homogeneous and non-overlapping sub-areas. The fuzzy zones are created based on historical data on origin-destination pairs so that they serve for estimating the probability of the occurrence of new ride requests.

An approach which is aimed at parallelizing the simulation is presented by Wei *et al.* [58] who partition the network with the objective of balancing the workload for each partition. Another road network partitioning technique for the purpose of employing a hierarchical path computation algorithm is described by Gonzalez *et al.* [59]. In their work, roads are classified according to different categories such as highways, main roads, neighborhood roads etc. which divide the road network into areas belonging to different hierarchical levels. Another example for this approach is a model developed by Zeng *et al.* [60]. It is based on partitioning the road network into a grid of rectangular tiles. The granularity of the grid is individually adapted to each single sharing case with four different possible resolutions ranging from  $(1 \times 1)$  km to  $(10 \times 10)$  km. A route can then be expressed as a sequence of tiles which is referred to as a *corridor*. For match making, only passengers whose origin and destination lie inside the corridor of an existing trip are

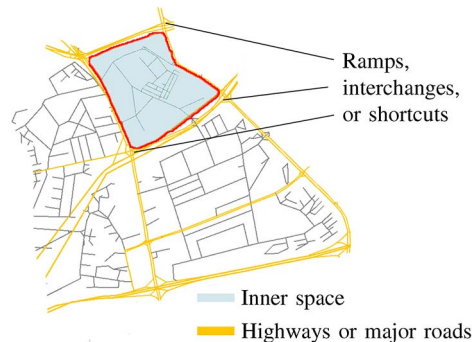


Fig. 1. Example of a partition where a set of minor roads is surrounded by major roads.

considered. While this approach is effective in regard to the reduction of the search space, the choice of partition granularity is somewhat arbitrary. The search space may therefore be either too large resulting in too many solutions or too small which may lead to good matches being filtered out. Furthermore, due to their fixed quadratic shape, partitions do not account for varying densities of the network so that the number of nodes, links and ride requests may highly vary among different partitions. This drawback common to a variety of partition-based methods motivates the approach presented in this paper which is discussed in greater detail in the following section.

## III. NETWORK PARTITIONING ALGORITHM

The approach presented in this paper builds on the method described by Zeng [60]. In contrast to their work, however, a partitioning procedure is defined which aims to create partitions of which the shape represents the topology of the road network. Due to the existence of two free parameters, the outcome of the partitioning process can be tuned with regard to certain objectives such as maximizing ride matches and minimizing inconvenience caused by detours. Based on the partitioned road network, a passenger's route is described as a corridor consisting of a sequence of partitions. A match making algorithm finally compares the corridors of different passengers and identifies a match in case the destination of one passenger is located within the other passenger's corridor. Due to the partitioning based on road network topology, matching partners can be expected to reside in areas of the network which can be conveniently reached so that the search space contains more desirable solutions than in case of a partitioning consisting of a simple rectangular grid.

One characteristic of a road network is its hierarchical structure which distinguishes different road categories, ranging from highways to minor roads. Typically, a number of roads that belong to a high category encloses a set of roads belonging to a lower category. The partitioning algorithm exploits this network characteristic by distinguishing three types of roads, i.e., highways, major roads and minor roads. As illustrated in Fig. 1, it then divides the network into partitions consisting of minor roads which are enclosed by sets of major roads.

Defining partitions simply by using major roads is, however, not always precise. This is illustrated in Fig. 1 where one



Fig. 2. Buffer operation results in a bounding shape with a width defined by  $\alpha$ .

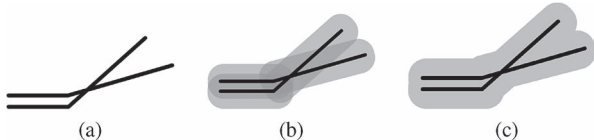


Fig. 3. Illustration of the buffer and union operation. Given a set of major roads (a), a buffer operation creates a set of individual bounding shapes (b). These bounding shapes are then combined by a union operation to obtain a single boundary containing all major roads (c).

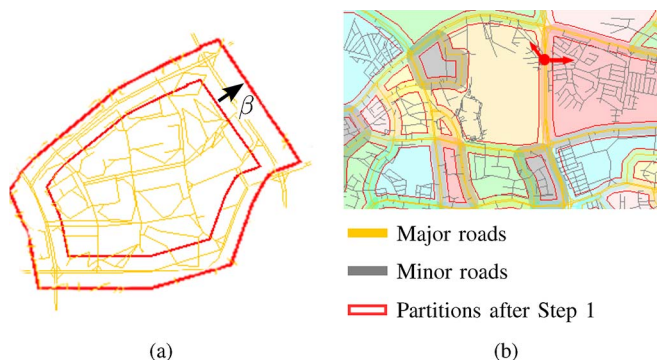


Fig. 4. Partition is scaled according to the parameter  $\beta$  (a). This results in partition overlaps which allow to match passengers from adjacent partitions (b).

partition is surrounded by multiple parallel roads on either side. Apart from the colored partition, this would create a number of undesired small partitions in between those parallel roads. To achieve a properly partitioned network, the partitioning process therefore consists of three distinct steps which are explained in the following paragraphs.

*a) Step 1—Basic Partitioning:* As illustrated in Fig. 2, the first step employs a buffer operation which generates a bounding shape around every major road. Subsequently, overlapping bounding shapes are combined by performing a union operation as illustrated in Fig. 3. The inner edge of a bounding shape then defines a partition. The width of the buffer zones depends on a parameter  $\alpha$  which determines the granularity of the partitioning and which will be used for partitioning optimization in Section V.

*b) Step 2—Partition Scaling:* As a result of Step 1, adjacent partitions are separated by a buffer zone of minimal width  $2 \cdot \alpha$ . To include this zone into the partitioned network, partitions need to be expanded by  $\beta$  as illustrated in Fig. 4(a). The minimum expansion to eliminate areas not contained in any partition is given by  $\beta = \alpha$ . With regard to ride matching, however, also an expansion by  $\beta > \alpha$  resulting in partition overlaps may be beneficial. As shown in Fig. 4(b), these overlaps may allow travelers residing near partition edges to be matched with travelers in a neighboring partition and thus increase their chances for a match. The degree of the overlap

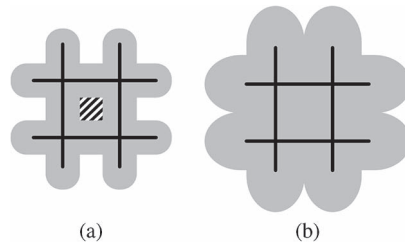


Fig. 5. Illustration of how a partition (patterned area) is defined by four major roads and their corresponding buffer zones (a). If  $\alpha$  is too large, the partition may disappear (b).

can be tuned by the parameter  $\gamma = \beta - \alpha$  which will be subject to the optimization discussed in Section V.

*c) Step 3—Fixing of Artifacts:* In a real-world road network, partitions may be of very different sizes. As illustrated in Fig. 5, small partitions could be lost if the value of  $\alpha$  is large. In case this is not compensated by a sufficiently high  $\beta$ , some roads may not be contained in the partitioned network. To resolve this issue, the buffer operation is also applied to the roads which are not included in any partition. In case the resulting buffer intersects with an existing partition, it is merged into this partition using the union operation. If no overlaps are found, a new partition is created. Other roads that may overlap with this new partition are then merged into it using the union operation.

#### IV. MATCH MAKING ALGORITHM

The purpose of the match making algorithm is to identify combinations of travelers who could share a ride. For matching purposes, travelers are distinguished by three different categories. The first category consists of those travelers who are just about to start their trip and are looking for a ride. The second category comprises passengers who have started their trip already but have not yet been matched. These travelers are looking for another passenger to share their ride. Travelers belonging to the third category have been matched already and are not available for further matches. The match making algorithm is invoked for every traveler  $k$  of the first category who is about to start a trip. If another traveler  $j$  belonging to the first or second category is found who fulfills the matching criteria defined below, both travelers are matched and a new corridor is calculated. If this causes the shared vehicle to reach its capacity limit, both travelers are assigned to the third category. If no match is found, agent  $k$  starts his own trip and thus switches to category two. In the following, the criteria for identifying valid matches are explained in further detail.

##### A. Corridor Subset Criterion

To be eligible for a match, both travelers need to reside in the same partition by the time one agent is about to start a trip. Furthermore, the routes of both travelers need to coincide which requires the destination partition of one traveler being a member of the corridor of the other. For each newly created agent, the algorithm therefore iterates over all travelers in the same partition and compares their respective corridors.

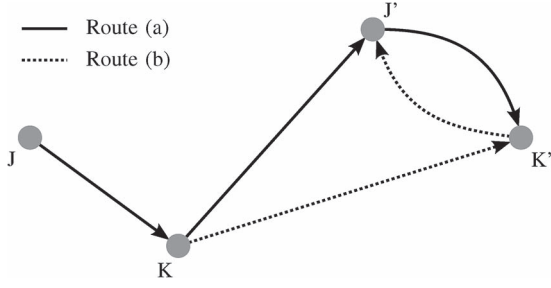


Fig. 6. Example of a traveler  $j$  and a passenger  $k$  sharing a ride.

### B. Inconvenience Criterion

As illustrated in Fig. 6, inconvenience arises from detours required for picking up or dropping off passengers. As shown in Fig. 6, there are two routing options for each pair of agents. Since agent  $j$  has started the ride already, the first trip consists of traveling from his current location  $J$  to the pickup location  $K$  of agent  $k$ . Option (a) is then to first drop off agent  $j$  at his destination  $J'$  and subsequently continue to agent  $k$ 's destination  $K'$ . Conversely, option (b) swaps destinations  $K'$  and  $J'$  so that  $k$  is dropped off before  $j$ .

- (a)  $J \rightarrow K \rightarrow J' \rightarrow K'$   
 (b)  $J \rightarrow K \rightarrow K' \rightarrow J'$ .

While both options may be feasible for taxi sharing, ridesharing involving a private vehicle may only allow for option (b) as option (a) would require agent  $j$  to hand his vehicle to agent  $k$ .

The different options cause different inconvenience to the travelers involved. With option (a), a passenger  $j$  traveling from  $J$  to  $J'$  sharing a ride with another person  $k$  traveling from  $K$  to  $K'$  needs to take a detour for picking up  $k$ . At the same time, passenger  $k$  has to wait until  $j$  reaches his pickup location and ultimately has to accept a detour for dropping off  $j$ . Option (b) imposes an extra detour on the itinerary of  $j$  and removes the detour from  $k$ 's route.

The resulting inconvenience in either case can be quantified by the deviation of the new route from the direct way and may be measured in various units such as extra travel time or additional driving distance. Choosing travel time as the measure of interest, a match is only considered feasible if the extra time related to a shared ride remains below a certain inconvenience limit  $I_{\max}$  for both sharing partners. Solution (a) can therefore only be feasible for  $j$  if

$$t_{\text{pickup}}^{J \rightarrow K} + t_{\text{shared}}^{K \rightarrow J'} - t_{\text{direct}}^{J \rightarrow J'} \leq I_{\max} \quad (1)$$

with  $t$  denoting the travel time between the locations denoted in the superscript. As agent  $k$  needs to wait for being picked up and ultimately has to accept a detour via  $J'$ , a second condition for the feasibility of this solution is that

$$t_{\text{pickup}}^{J \rightarrow K} + t_{\text{shared}}^{K \rightarrow J'} + t_{\text{dropoff}}^{J' \rightarrow K'} - t_{\text{direct}}^{K \rightarrow K'} \leq I_{\max}. \quad (2)$$

Following the same argumentation, solution (b) is only feasible if for agent  $j$  the entire trip only exceeds the duration of the

direct trip  $J \rightarrow J'$  by  $I_{\max}$  so that

$$t_{\text{pickup}}^{J \rightarrow K} + t_{\text{shared}}^{K \rightarrow K'} + t_{\text{dropoff}}^{K' \rightarrow J'} - t_{\text{direct}}^{J \rightarrow J'} \leq I_{\max}. \quad (3)$$

The condition that the waiting time  $t_{\text{pickup}}^{J \rightarrow K}$  of agent  $k$  also needs to be below  $I_{\max}$  is implicitly contained in (3).

### C. Mileage Savings Criterion

From a system perspective, a match is desirable if the total driving distance of the shared ride  $d_{\text{match total}}$  is shorter than the sum of the direct individual trips  $d_{\text{direct}}^{K \rightarrow K'}$  and  $d_{\text{direct}}^{J \rightarrow J'}$ . Hence, a last feasibility condition for any option (a) or (b) is that

$$d_{\text{match total}}^{(a,b)} < d_{\text{direct}}^{K \rightarrow K'} + d_{\text{direct}}^{J \rightarrow J'}. \quad (4)$$

It may be the case that both options (a) and (b) lead to feasible solutions and that there are multiple agents having a feasible match with agent  $k$ . Among all feasible solutions, one is chosen which maximizes the driving distance savings. These savings are expressed by the difference between the sum of the distances of the direct individual trips  $d_{\text{direct}}^{K \rightarrow K'}$ ,  $d_{\text{direct}}^{J \rightarrow J'}$  and the total length of the shared ride  $d_{\text{match total}}$ , leading to the condition

$$\text{maximize} \left( d_{\text{direct}}^{K \rightarrow K'} + d_{\text{direct}}^{J \rightarrow J'} - d_{\text{match total}} \right). \quad (5)$$

## V. OPTIMIZATION METHODOLOGY

The main metrics of interest are the mileage savings  $S$  and the sharing potential  $R$ . The mileage savings  $S$  indicate how many kilometers of driving are saved by sharing rides. They are determined by the accumulated daily mileage  $D$  of all agents traveling individually as compared to the corresponding mileage  $D'$  when employing ridesharing:

$$S = \frac{D - D'}{D}. \quad (6)$$

In turn, the sharing potential is expressed by the number of matches  $M$  divided by the entire number of trips  $N$ :

$$R = \frac{M}{N}. \quad (7)$$

As a match reduces the number of trips by one,  $R$  also denotes the relative trip reduction achieved by ridesharing.

The objective is to maximize  $S$  while keeping the inconvenience  $I_k$  imposed on each passenger  $k$  below a certain limit  $I_{\max}$ . The probability for finding a match depends on number, size distribution, and shape of the partitions which are in turn a function of  $\alpha$  and  $\gamma$ . A large value for  $\alpha$  is expected to yield larger partitions which lead to an increased size of the search space. This implies a higher sharing potential which in turn can be expected to yield higher mileage savings. The same applies to large  $\gamma$  which cause partition overlaps that also increase the pool of potentially matchable travelers. While the number of possible matches increases with  $\alpha$  and  $\gamma$ , the growing partition size also leads to an increasing share of matches with detours greater than the inconvenience limit  $I_{\max}$ . Finding the best solution in this growing search space therefore also comes with

increased computational efforts. Finding an optimal partition configuration while keeping inconvenience below a certain limit translates into the optimization problem

$$\underset{\alpha, \gamma}{\text{maximize}} \quad S(\alpha, \gamma) \quad (8)$$

$$\text{subject to} \quad I_k \leq I_{\max} \quad \forall k. \quad (9)$$

The presented approach aims at identifying a partitioning of the road network which during system operation yields a good exploitation of the sharing potential under the given limitations. For this purpose, a great number of different partition configurations with each configuration corresponding to a specific tuple  $(\alpha, \gamma)$  are created. By means of simulation, these configurations are then classified according to the optimization problem defined above. The optimal partition configuration can finally be used for both further in-depth simulation studies as well as real-time ridesharing system operation.

## VI. CASE STUDY FOR SINGAPORE

As an application to a real-world scenario, this methodology was employed to investigate the potential for taxi sharing in the city of Singapore. The simulation was performed using the agent-based tool SEMSim [61]. SEMSim allows simulating a large number of individual vehicles on the scale of an entire city. Trips are generated according to an origin-destination matrix which contains data on starting time, starting location and destination of all trips. These trips are generated in real-time so that the problem is purely dynamic. When a new agent is created,  $N$  is incremented by one. At the same time, the system checks whether there is another agent in the same partition who could serve as a sharing partner according to the above defined criteria. If a sharing partner is found,  $M$  is incremented by one; otherwise the newly created agent starts his own trip. At the same time, route lengths are calculated using the standard Dijkstra algorithm to update the total mileage. To account for limited vehicle capacities, each agent cannot be matched more than once. In this sense, the investigated problem is a single rider—single driver problem. According to the definition of the sharing potential in (7), the fraction of shared rides out of the entire number of trips is then given by  $2 \cdot R$ .

In the first step, a road network of Singapore was partitioned according to the method described in Section III. Initially, for  $(\alpha, \gamma)$  a range  $\alpha \in [20, 600]$  m and  $\gamma \in [0, 1000]$  m with a resolution of 20 m was chosen. The computations were executed on a standard computer equipped with a 3.2 GHz quad core Intel Xeon CPU, and 10 GB memory was used.

Fig. 7 shows the dependency of the number of partitions from  $\alpha$ . The number of partitions decreases with increasing  $\alpha$  since smaller partitions get aggregated at larger  $\alpha$  values. As can be observed from an analysis of partition size distributions, large  $\alpha$  lead to configurations where boundaries between partitions are lost so that a small number of very large partitions emerges. These partitions may expand over a large part of the network and contain many of the smaller partitions as islands. Due to their size, these partitions are not desirable since they contain a large number of ultimately infeasible solutions. The remain-

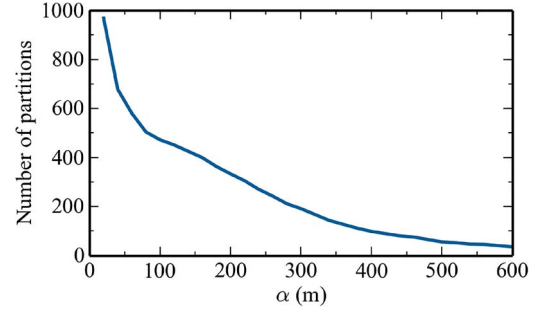


Fig. 7. Number of partitions depending on  $\alpha$ .

ing considerations are therefore restricted to an interval  $\alpha \in [60, 180]$  m. The lower bound of this interval results from the observation that smaller values lead to a network fragmentation into regions of very small size which cannot be expected to yield a sufficient number of matches.

The input for the travel demand is based on real-world traffic data for Singapore. Building on this data, 50 000 agents which perform a total number of about 110 000 daily trips were simulated. As travel demand significantly fluctuates during the course of a day, a period of 24 hours was investigated. For this calculation, the parameters  $(\alpha, \gamma)$  were restricted to an interval  $\alpha \in [60, 180]$  m. Since an increase of  $\gamma$  can be expected to yield a growing sharing potential up to a certain threshold,  $\gamma$  was successively increased up to a value where the sharing potential saturated. As a tradeoff between computing time and accuracy, the resolution of  $\alpha$  was set to 40 m and the  $\gamma$  resolution to 80 m.

The result of the simulation is shown in Fig. 8. The scale of  $\gamma$  is cut off at a value where the sharing potential does not exhibit any further increase. As shown in Fig. 8(a), increasing values for  $\alpha$  at low  $\gamma$  values lead to a significant sharing potential increase. This is because increasing  $\alpha$  causes small partitions, which do not make any relevant contribution to the sharing potential, to be merged to bigger ones. Towards larger  $\alpha$  values, this increase, however, also comes at the cost of a significant run time increase caused by greater partition size.

At larger  $\gamma$ , the sharing potential is governed by  $\gamma$  so that the value of  $\alpha$  does not exhibit any considerable effect. Increasing  $\gamma$  expands partitions so that small partitions either get large enough to contribute to the sharing potential or get overlaid by others. This increase saturates at a maximum sharing potential of 42% around  $\gamma = 600$  m where a further partition size increase does not lead to any additional matches. The  $\gamma$  dependency of the run time is low so that configurations with large  $\gamma$  and moderate  $\alpha$  values are preferable both regarding sharing potential and computing time.

A similar dependency is observed for the mileage savings depicted in Fig. 8(b).  $S$  ranges from 11% at low  $\alpha$  and  $\gamma$  up to 20% for large  $\gamma$  values. This means that the best configuration reduces the mileage driven by all vehicles in one day by 230 000 km compared to a scenario without ridesharing.

Fig. 9 shows sharing potential and mileage savings as a function of the time of the day for the configurations where sharing potential and mileage savings are the highest. It can be seen that both  $R$  and  $S$  qualitatively follow the number of

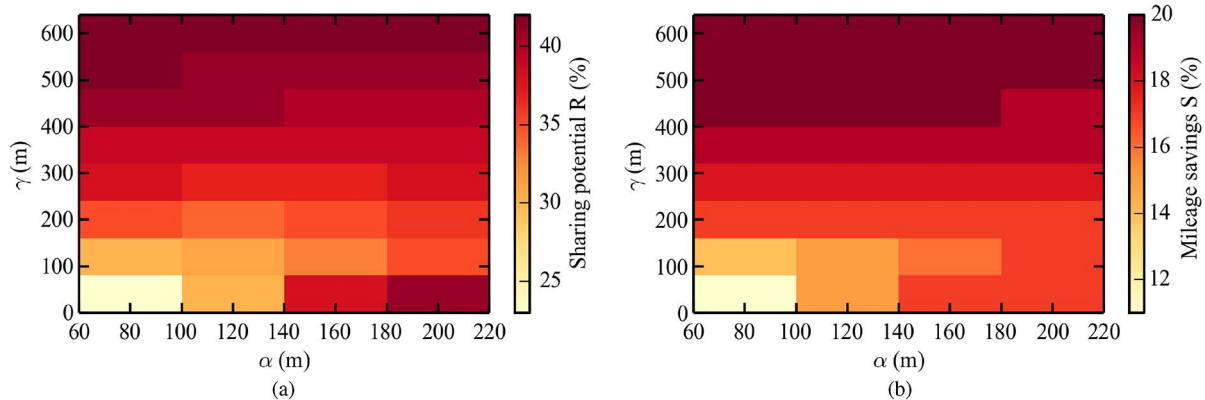


Fig. 8. Simulation results for the sharing potential (a) and the mileage savings (b) depending on  $\alpha$  and  $\gamma$  at a maximum detour of 10 minutes per trip.

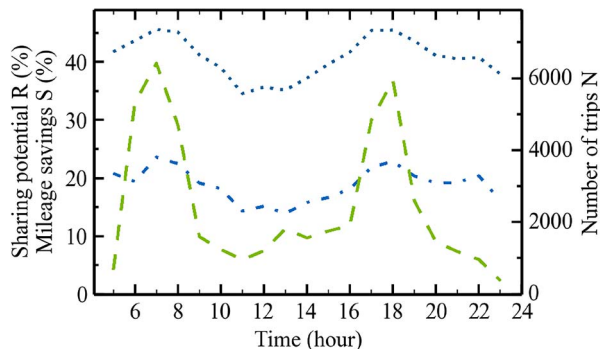


Fig. 9. Maximum sharing potential (dotted line), maximum mileage savings (dashed dotted line) and number of hourly trips (dashed line) depending on the time of the day.

started trips. This is expected since a larger pool of rides also increases chances for a match. The increase is, however, not proportional to the trip number. The maximum sharing potential of 46% and the maximum mileage savings of 23% are achieved in peak hours at 7am and 6pm but still remain at a notable level above 36% and 14%, respectively during off-peak hours. This shows that despite comparably low trip numbers, matching chances remain considerably high.

In principle, different times of the day exhibit typical mobility patterns where the average flows of travelers have different directions. It could therefore be expected that certain partition configurations might be more beneficial at certain times. Choosing an optimal partition configuration for each hour of the day, however, is observed to have a negligible effect on the sharing potential. The most favorable configurations with regard to the whole day therefore also belong to the most favorable ones in each individual time interval.

The computing time required by the partitioning algorithm shows a strong dependency on the choice of  $\alpha$ . With small  $\alpha$  values, a fine-grained partitioning is created. This means that the number of roads which need to be fixed in Step 3 discussed in Section III is small. The opposite is the case for larger  $\alpha$  which lead to a greater number of roads that initially are not contained in the partitioned network. As the buffer operation fixing this issue is computationally expensive, the computing time increases with  $\alpha$ . In the interval  $\alpha \in [60, 180]$  m, the

partitioning algorithm for one configuration terminates within a time frame of one to ten minutes. As the partitioning can be performed a priori and therefore needs to be done only once, the only time critical part is the match making which is conducted during system operation. This step, however, only needs a very short time of about 1 ms per match so that real-time capability also in large-scale systems is ensured.

## VII. COMPARISON TO GREEDY HEURISTIC

Given the large scale and the dynamics of the problem, comparing the presented method to classical approaches known from operations research literature is not feasible. To still provide an intuitive understanding of the performance of the partition-based approach, a comparison to a simple greedy heuristic was performed. For this purpose, this heuristic was integrated into the simulation framework and applied to the setup of the case study described above.

### A. Greedy Algorithm

The concept of this heuristic is to search for possible matching partners in the vicinity of an agent and to choose the best among all feasible options. When a new agent  $k$  is created, the first step is to identify all feasible matches within a circle with radius  $r$  around the agent. In accordance with the partitioning approach, this is done by assessing the feasibility and optimality of a solution by the criteria defined in Sections IV-B and IV-C. The main difference between this approach and the partition-based method is that in this case the search space is not determined by partitions adapted to the structure of the road network. The resulting implications on mileage savings  $S$ , sharing potential  $R$ , and computing time are discussed in the following section.

### B. Comparison of Results

For comparing both methods, the best partition configuration identified in Section VI was compared to the greedy approach using different search radii  $r$ . With regard to the mileage savings  $S$  it was distinguished between savings related to the first



TABLE I  
SHARING POTENTIAL AND TOTAL MILEAGE SAVINGS ACHIEVED BY THE  
PARTITION-BASED METHOD AND THE GREEDY HEURISTIC

Method	$r$ (km)	$R$ (%)	$S_1$ (%)	$S_2$ (%)
Partition-based	-	42	19.4	20.1
Greedy heuristic	12	46	15.9	17.1
Greedy heuristic	6	46	16.5	17.6
Greedy heuristic	5	46	16.8	17.9
Greedy heuristic	4	45	17.2	18.2
Greedy heuristic	3	45	17.6	18.6
Greedy heuristic	2.5	45	17.8	18.7
Greedy heuristic	2	44	18.0	18.9
Greedy heuristic	1.5	43	18.0	18.8
Greedy heuristic	1	41	17.1	18

feasible match according to (1)–(4),  $S_1$ , and savings resulting from the best match according to (1)–(5),  $S_2$ .

As shown in Table I, the sharing potential achieved by the greedy approach increases with the search radius and ultimately saturates at a value of  $R = 46\%$ . This is slightly higher than achieved by the partition-based approach where a maximum sharing potential of 42 is obtained. This can be expected since given a sufficiently high search radius, the greedy method will certainly identify any feasible match while some matches may be missed by the partition-based method.

Despite the lower sharing potential, the partition-based approach performs better with regard to mileage savings. In relative terms, the maximum savings of the partition-based method are 6% higher than those achieved by the greedy approach. In absolute numbers, this corresponds to 14 600 km lower daily mileage achieved by the partition-based method. This can be explained by the fact that the greedy approach may sometimes find matching partners which are comparably far apart so that they are not considered by the partition-based method. Matching these partners makes them unavailable for later matches where a better sharing partner could be found. While reducing  $r$  could be expected to remedy this issue, it is observed that the further decrease of  $r$  also reduces the sharing potential which in turn also leads to lower mileage savings. In this sense, the partition-based approach tends to find the ‘better’ matches in terms of system performance.

Expectedly, the mileage savings  $S_2$  are greater than  $S_1$  for both methods. For the greedy approach, the differences between the first feasible and the best solution range from 4.4% to 7.5% relative, for the partition-based method a 3.6% difference is observed. These generally low numbers indicate that the feasibility criteria (1)–(4) already perform a reasonable preselection of solutions. The corridor subset comparison conducted by the partition-based method, however, appears to perform better since the difference between  $S_1$  and  $S_2$  remains lower than for the greedy approach.

With regard to run time, the partition-based method is typically faster than the greedy approach. At  $r = 2$  km where the greedy approach performs best in terms of system mileage savings, its run time is about 30% higher than for the partition-based approach. This is because the greedy method generally involves a larger number of computationally expensive routing operations while the partition-based approach first reduces the size of the search space by comparing partition corridor subsets. While this difference can still be considered fairly small, it can

be expected to increase when considering higher numbers of agents or more than two matching partners.

## VIII. DISCUSSION

The outcome of the partitioning process shows a well-partitioned network in accordance with the topology of the road network. While increasing either  $\alpha$  or  $\gamma$  yields higher sharing potentials and mileage savings, choosing sufficiently high values for  $\gamma$  makes the outcome independent of  $\alpha$  as long as  $\alpha$  is kept in a certain range. In this sense, the choice for  $\alpha$  is crucial for achieving a network which is neither fragmented into a large number of small partitions nor consists of a small amount of large partitions. Given a reasonable choice for  $\alpha$ , increasing  $\gamma$  then maximizes sharing potential and mileage savings. Raising  $\gamma$  beyond a threshold does not yield any further gains but increases the run time of the algorithm.

With  $R = 42\%$  and  $S = 20\%$  corresponding to absolute savings of 230 000 km per day, the results reveal a significant potential for transportation system efficiency improvements through ridesharing. Since in this case only single rider—single driver assignments were considered, these numbers could be further increased by allowing multiple matching partners. Also, compared to 967 000 daily taxi trips in Singapore [1], the amount of about 110 000 trips per day considered in this study is fairly low. As it was shown that higher numbers of agents increase sharing potential while detours can be expected to be decreased, this is assumed to yield further mileage savings.

The comparison between the partition-based and the greedy method shows that the partition-based approach performs well in identifying feasible matches. At the same time, the higher mileage savings of the partition-based method show that the corridor subset criterion is a good measure for search space size reduction. By allowing more than two passengers to be matched, the number of routing requests would considerably increase so that the run time advantage of the corridor subset criterion would play a significantly greater role.

One more benefit of the partition-based method is not reflected in these results. In many cases, moving from one sub-region of the road network into another involves a transfer via major roads. During peak hours, travel speeds on these roads are often significantly decreased. Furthermore, transfers from one network sub-region into another may require passing a greater number of traffic lights than a trip within the same sub-region. Both aspects may lead to a significant increase in travel times. The partition-based approach aims at finding matches in the same sub-region of the network, thus avoiding these delays.

A factor inherent to the partition-based approach which could lead to a sharing potential underestimation is that certain ‘good’ matches may not be counted. There may be trip configurations at which the pickup point of a potential sharing partner is outside of the corridor while the detour to the drop off point is very small, or vice versa. In this case, the total inconvenience of pickup and drop off may be smaller than for a trip which fulfills the corridor criterion. This issue could be resolved by modifying the match making process. For this purpose, a configuration with smaller partitions could be used. If, for instance, the starting point of two potential sharing partners

would be in one of these small partitions, this would indicate that the expected pickup detour is small. In this case, a longer detour for the drop off would be justified. This can be realized by also considering partitions in the vicinity of the corridor as potential drop off locations. With this combination, the total expected detour would still remain below the desired limit. The small sharing potential difference between the greedy heuristic and the partition-based approach, however, indicates that the effect of this issue is very small.

An aspect which could further improve sharing potential while decreasing run time is to unify partition size with regard to expected detours. This could be achieved by refining the partitioning algorithm in a way such that partitions in which a significant number of trips exceed the inconvenience limit are sub-divided while small partitions are re-aggregated to larger areas. Such a procedure could ultimately yield configurations in which the variance of expected inconvenience among all partitions would be constrained to a narrow interval which is most beneficial regarding both sharing potential maximization and inconvenience minimization.

## IX. CONCLUSION AND FUTURE WORK

In this paper, a match making algorithm for dynamic ridesharing based on network partitioning was presented. The aim of this approach is to minimize the mileage driven in the entire system while keeping the inconvenience caused by detours for picking up and dropping off passengers below a certain limit. Addressing this problem on a large scale requires reducing the space in which the algorithm searches for matches. This is achieved by partitioning the road network into distinct regions which represent certain sub-structures of the road network. Consequently, routes can be described as corridors defined by sequences of these partitions. Passengers can then be matched if the destination partition of one passenger lies within the other one's corridor. This approach significantly reduces the number of computationally expensive route calculations enabling the method to address systems at a very large scale involving a great number of agents.

As a case study, the method was applied to investigate a dynamic taxi sharing problem consisting of about 110 000 daily trips in Singapore. This was done by employing a microscopic traffic simulation using real-world mobility demand data. The case study confirmed the applicability of the algorithm and allowed conclusions on a reasonable value range for the two optimization parameters. The results demonstrated that, in the case of a single rider—single driver problem and a maximum accepted detour per traveler of 10 minutes, the number of trips could be reduced by 42%. This would reduce the accumulated daily system mileage by 230 000 km.

To provide an intuitive understanding of the performance of the developed approach, a comparison to a greedy heuristic was conducted. The results show that for a single rider—single driver problem, the partition-based approach yields 6% higher mileage savings in 30% lower run times. Further benefits of the partition-based method regarding computational efforts are expected in case of higher numbers of agents or when considering multiple rider—single driver problems.

Future work will consist of further improving the partitioning algorithm and the match making procedure. As a possible improvement of the partitioning algorithm, the creation of partitions with equal properties in regard to expected detours was identified. This can be achieved by splitting up partitions where generally longer detours occur or by merging partitions where detours are below the accepted threshold. The match making procedure can be further developed in various ways. One improvement was identified to be the consideration of potential sharing partners in neighboring partitions which may be missed in the current implementation. Another improvement is to also consider multiple rider—single driver assignments which would yield an increase in sharing potential exploitation. Ultimately, individual user preferences and fair pricing strategies will be considered which will allow accounting for the tradeoff between overall traffic system performance and individual costs and benefits.

## REFERENCES

- [1] Statistics in Brief, Land Transport Authority, Singapore, 2013.
- [2] B. Eksioglu, A. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, Nov. 2009.
- [3] L. Bodin, "A taxonomic structure for vehicle routing and scheduling problems," *Comput. Urban Soc.*, vol. 1, no. 1, pp. 11–29, Jan. 1975.
- [4] L. Bodin and B. Golden, "Classification in vehicle routing and scheduling," *Networks*, vol. 11, no. 2, pp. 97–108, 1981.
- [5] M. Desrochers, J. Lenstra, and M. Savelsbergh, "A classification scheme for vehicle routing and scheduling problems," *Eur. J. Oper. Res.*, vol. 46, no. 3, pp. 322–332, Jun. 1990.
- [6] M. Desrochers, C. Jones, J. Lenstra, M. Savelsbergh, and L. Stougie, "Towards a model and algorithm management system for vehicle routing and scheduling problems," *Decision Support Syst.*, vol. 25, no. 2, pp. 109–133, Mar. 1999.
- [7] H. Min, V. Jayaraman, and R. Srivastava, "Combined location-routing problems: A synthesis and future research directions," *Eur. J. Oper. Res.*, vol. 108, no. 1, pp. 1–15, Jul. 1998.
- [8] W. Powell, J. Shapiro, and H. Simao, "A representational paradigm for dynamic resource transformation problems," *Ann. Oper. Res.*, vol. 104, no. 1–4, pp. 231–279, 2001.
- [9] H. Psaraftis, "Dynamic vehicle routing: Status and prospects," *Ann. Oper. Res.*, vol. 61, no. 1, pp. 143–164, 1995.
- [10] G. Laporte, "Fifty years of vehicle routing," *Transp. Sci.*, vol. 43, no. 4, pp. 408–416, 2009.
- [11] T. Vidal, T. Crainic, M. Gendreau, and C. Prins, "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis," *Eur. J. Oper. Res.*, vol. 231, no. 1, pp. 1–21, Nov. 2013.
- [12] V. Pillac, M. Gendreau, C. Guéret, and A. Medaglia, "A review of dynamic vehicle routing problems," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 1–11, Feb. 2013.
- [13] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: A classification scheme and survey," *TOP*, vol. 15, no. 1, pp. 1–31, Jul. 2007.
- [14] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 8–15, Apr. 2010.
- [15] M. W. P. Savelsbergh and M. Sol, "The general pickup and delivery problem," *Transp. Sci.*, vol. 29, no. 1, pp. 17–29, 1995.
- [16] S. Parragh, K. Doerner, and R. Hartl, "A survey on pickup and delivery problems," *J. für Betriebswirtschaft*, vol. 58, no. 1, pp. 21–51, Apr. 2008.
- [17] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations," *J. für Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, Apr. 2008.
- [18] M. Savelsbergh and M. Sol, "Drive: Dynamic routing of independent vehicles," *Oper. Res.*, vol. 46, no. 4, pp. 474–490, Aug. 1998.
- [19] A. Fabri and P. Recht, "On dynamic pickup and delivery vehicle routing with several time windows and waiting times," *Transp. Res. B, Methodol.*, vol. 40, no. 4, pp. 335–350, May 2006.

- [20] M. Gendreau, F. Guertin, J.-Y. Potvin, and R. Séguin, "Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries," *Transp. Res. C, Emerging Technol.*, vol. 14, no. 3, pp. 157–174, Jun. 2006.
- [21] S. Mitrović-Minić, R. Krishnamurti, and G. Laporte, "Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows," *Transp. Res. B, Methodol.*, vol. 38, no. 8, pp. 669–685, Sep. 2004.
- [22] S. Mitrović-Minić and G. Laporte, "Waiting strategies for the dynamic pickup and delivery problem with time windows," *Transp. Res. B, Methodol.*, vol. 38, no. 7, pp. 635–655, Aug. 2004.
- [23] C. Cortés, A. Núñez, and D. Sáez, "Hybrid adaptive predictive control for a dynamic pickup and delivery problem including traffic congestion," *Int. J. Adaptive Control Signal Process.*, vol. 22, no. 2, pp. 103–123, 2008.
- [24] D. Sáez, C. E. Cortés, and A. Núñez, "Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering," *Comput. Oper. Res.*, vol. 35, no. 11, pp. 3412–3438, Nov. 2008.
- [25] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: Models and algorithms," *Ann. Oper. Res.*, vol. 153, no. 1, pp. 29–46, Sep. 2007.
- [26] J. Paquette, J.-F. Cordeau, and G. Laporte, "Quality of service in dial-a-ride operations," *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1721–1734, May 2009.
- [27] H. N. Psaraftis, "Dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem," *Transp. Sci.*, vol. 14, no. 2, pp. 130–154, 1980.
- [28] H. N. Psaraftis and N. Harilaos, "Exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows," *Transp. Sci.*, vol. 17, no. 3, pp. 351–357, 1983.
- [29] J.-J. Jaw, A. Odoni, H. Psaraftis, and N. Wilson, "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows," *Transp. Res. B, Methodol.*, vol. 20, no. 3, pp. 243–257, Jun. 1986.
- [30] O. Madsen, H. Ravn, and J. Rygaard, "A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives," *Ann. Oper. Res.*, vol. 60, no. 1, pp. 193–208, 1995.
- [31] D. Teodorovic and G. Radivojevic, "A fuzzy logic approach to dynamic dial-a-ride problem," *Fuzzy Sets Syst.*, vol. 116, no. 1, pp. 23–33, Nov. 2000.
- [32] L. Coslovich, R. Pesenti, and W. Ukovich, "A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem," *Eur. J. Oper. Res.*, vol. 175, no. 3, pp. 1605–1615, Dec. 2006.
- [33] S. Hartwig and M. Buchmann "Empty seats traveling: Next-generation ridesharing and its potential to mitigate traffic- and emission problems in the 21st century," Nokia Res. Center, Espoo, Finland, Tech. Rep., 2007.
- [34] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *Eur. J. Oper. Res.*, vol. 223, no. 2, pp. 295–303, Dec. 2012.
- [35] M. Furuhata *et al.*, "Ridesharing: The state-of-the-art and future directions," *Transp. Res. B, Methodol.*, vol. 57, pp. 28–46, Nov. 2013.
- [36] E. Kamar and E. Horvitz, "Collaboration and shared plans in the open world: Studies of ridesharing," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 187–194.
- [37] A. Kleiner, B. Nebel, and V. Ziparo, "A mechanism for dynamic ride sharing based on parallel auctions," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 266–272.
- [38] D. Pentico, "Assignment problems: A golden anniversary survey," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 774–793, Jan. 2007.
- [39] S. Yan, C.-Y. Chen, and Y.-F. Lin, "A model with a heuristic algorithm for solving the long-term many-to-many car pooling problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1362–1373, Dec. 2011.
- [40] C. Stach, "Saving time, money and the environment—Vhike a dynamic ride-sharing service for mobile devices," in *Proc. IEEE Int. Conf. PERCOM Workshops*, 2011, pp. 352–355.
- [41] A. Di Febraro, E. Gattorna, and N. Sacco, "Optimization of dynamic ridesharing systems," *Transp. Res. Rec.*, vol. 2359, pp. 44–50, Oct. 2013.
- [42] P.-Y. Chen, J.-W. Liu, and W.-T. Chen, "A fuel-saving and pollution-reducing dynamic taxi-sharing protocol in vanets," in *Proc. 72nd IEEE VTC—Fall*, 2010, pp. 1–5.
- [43] R. Baldacci, V. Maniezzo, and A. Mingozzi, "An exact method for the car pooling problem based on Lagrangean column generation," *Oper. Res.*, vol. 52, no. 3, pp. 422–439, Jun. 2004.
- [44] C.-C. Tao, "Dynamic taxi-sharing service using intelligent transportation system technologies," in *Proc. Int. Conf. WiCom, Netw. Mobile Comput.*, 2007, pp. 3209–3212.
- [45] G. Gidofalvi and T. B. Pedersen, "Cab-sharing: An effective, door-to-door, on-demand transportation service," in *Proc. 6th Eur. Congr. Intell. Transp. Syst. Serv.*, 2007, pp. 1–8.
- [46] P. d'Orey, R. Fernandes, and M. Ferreira, "Empirical evaluation of a dynamic and distributed taxi-sharing system," in *Proc. 15th ITSC*, 2012, pp. 140–146.
- [47] R. Geisberger, D. Luxen, S. Neubauer, P. Sanders, and L. Volker, "Fast detour computation for ride sharing," Karlsruhe Inst. Technol., Karlsruhe, Germany, Tech. Rep., 2009.
- [48] M. Mes, M. van der Heijden, and A. van Harten, "Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems," *Eur. J. Oper. Res.*, vol. 181, no. 1, pp. 59–75, 2007.
- [49] S. Winter and S. Nittel, "Ad hoc shared-ride trip planning by mobile geosensor networks," *Int. J. Geograph. Inf. Sci.*, vol. 20, no. 8, pp. 899–916, Jul. 2006.
- [50] L. D. Bodin and T. R. Sexton, "The multi-vehicle subscriber dial-a-ride problem," *TIMS Studies Manage. Sci.*, vol. 2, pp. 73–86, 1986.
- [51] Y. Dumas, J. Desrosiers, and F. Soumis, "Large scale multi-vehicle dial-a-ride problems," Les Cahiers du GERAD, Montreal, QC, Canada, 1989.
- [52] J. Desrosiers, Y. Dumas, F. Soumis, S. Taillefer, and D. Villeneuve, "An algorithm for mini-clustering in handicapped transport," Cahiers du GERAD, Montreal, QC, Canada, 1991.
- [53] I. Ioachim, J. Desrosiers, Y. Dumas, M. M. Solomon, and D. Villeneuve, "A request clustering algorithm for door-to-door handicapped transportation," *Transp. Sci.*, vol. 29, no. 1, pp. 63–78, 1995.
- [54] R. Borndörfer, M. Grötschel, F. Klostermeier, and C. Küttner, "Telebus berlin: Vehicle scheduling in a dial-a-ride system," in *Computer-Aided Transit Scheduling*, ser. Lecture Notes in Economics and Mathematical Systems, N. Wilson, Eds. Berlin, Germany: Springer-Verlag, 1999, vol. 471, pp. 391–422.
- [55] R. H. Möhring, H. Schilling, B. Schütz, D. Wagner, and T. Willhalm, "Partitioning graphs to speed up dijkstra's algorithm," *Exp. Efficient Algorithms*, vol. 3503, pp. 189–202, 2005.
- [56] G. Karypis and V. Kumar, "Metis—Unstructured graph partitioning and sparse matrix ordering system, Version 2.0," Univ. Minnesota, Minnesota, MN, USA, Tech. Rep., 1995.
- [57] J. C. Bezdek, *Fuzzy Mathematics in Pattern Classification*. Ithaca, NY, USA: Cornell Univ., 1973.
- [58] D. Wei, F. Chen, and X. Sun, "An improved road network partition algorithm for parallel microscopic traffic simulation," in *Proc. 2010 Int. Conf. MACE*, 2010, pp. 2777–2782.
- [59] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag, "Adaptive fastest path computation on a road network: A traffic mining approach," in *Proc. 33rd Int. Conf. VLDB*, 2007, pp. 794–805.
- [60] Y. Zeng, M. Szczygiel, and B. Honary, "Car-share: Making the right connection ride matching," in *Proc. 12th Annu. PGNet*, 2011, pp. 1–5.
- [61] Y. Xu, H. Aydt, and M. Lees, "Semsim: A distributed architecture for multi-scale traffic simulation," in *Proc. ACM/IEEE/SCS 26th Workshop PADS*, 2012, pp. 178–180.



**Dominik Pelzer** received the postgraduate degree (Diplom-Physiker) in physics and economics from Heidelberg University, Heidelberg, Germany.

He is currently working as a Research Associate at TUM CREATE, Singapore. Prior to this he was employed as a Research Assistant at the Fraunhofer-Institute for Solar Energy Systems (ISE) in Freiburg, Germany. His research areas include energy systems and urban transportation infrastructures which he is investigating from a complex systems perspective.



**Jiajian Xiao** received the B.Eng. degree in computer science from Shanghai Jiaotong University, Shanghai, China, in 2011 and a M.Sc. degree in informatics from Technische Universität München, Munich, Germany, in 2013. He is currently pursuing the Ph.D. degree at Technische Universität München.

He is a Software Developer at TUM CREATE, Singapore. His research interests include interactive high performance computer simulation and simulation-based optimization.



**Daniel Zehe** received the B.Sc. and M.Sc. degrees in computer systems in engineering from Magdeburg University, Magdeburg, Germany, in 2011 and 2012, respectively. He is currently pursuing the Ph.D. degree at Technische Universität München, Munich, Germany.

He is currently working as a Research Associate at TUM CREATE, Singapore. Prior to this, he worked at BMW Research and Technology. His fields of research include multi-resolution modeling for traffic systems and cloud computing for large-scale agent-based simulation.



**Michael H. Lees** received the Ph.D. degree from the School of Computer Science at the University of Nottingham, Nottingham, U.K.

Currently he is an Assistant Professor in the Section Computational Science, University of Amsterdam, Amsterdam, Netherlands. Prior to this he was an Assistant Professor in the School of Computer Engineering of Nanyang Technological University (NTU), Singapore. He has post-doctoral experience at NTU, the University of Nottingham and the University of Birmingham, U.K. His research

interests are in modeling and simulation of large scale complex systems, he is particularly interested in understanding the effects human behavior and individual behavioral interactions have on system level dynamics.

Dr. Lees is currently workshop chair of the International Conference on Computational Science (ICCS) and an editor of the Journal of Computational Science (JOCS).



**Alois C. Knoll** (M'88) received the Dipl.Ing. (M.Sc.) degree in electrical/communications engineering from the Universität Stuttgart, Stuttgart, Germany, in 1985 and the Ph.D. degree (summa cum laude) in computer science from the Technische Universität (TU) Berlin, Berlin, Germany, in 1988.

He served on the faculty of the computer science department of TU Berlin until 1993, when he qualified for teaching computer science at a university (habilitation). He then joined the Universität Bielefeld, where he became a Full Professor and Director

of the research group Technical Informatics until 2001. From 2001 to 2004 he was a group head and a member of the board of directors of the Fraunhofer-Institute for Autonomous Intelligent Systems. Since autumn 2001 he has been a professor of Computer Science at the Computer Science Department of the Technische Universität München (TUM), Munich, Germany. He is also on the board of directors of the Central Institute of Medical Technology at TUM (IMETUM); from 2004 to 2006 he was Executive Director of the Institute of Computer Science at TUM. His research interests include cognitive, medical and sensor-based robotics, multi-agent systems, data fusion, adaptive systems and multimedia information retrieval.

Dr. Knoll has published over 200 technical papers and guest-edited international journals. He has participated (and has coordinated) several large scale national collaborative research projects. He initiated and was the program chairman of the First IEEE/RAS Conference on Humanoid Robots, he was general chair of IEEE Humanoids2003 and general chair of Robotik 2004, the largest German conference on robotics, and he served on several other organizing committees. He is a member of the German Society for Computer Science.



**Heiko Aydt** received the Dipl.Ing. (FH) degree in information technology from Esslingen University of Applied Sciences, Esslingen, Germany, the M.Sc. degree from the Royal Institute of Technology (KTH) in Stockholm, Sweden, and the Ph.D. degree in computer science from Nanyang Technological University (NTU) in Singapore.

Prior to his doctoral studies, he gained experience as Software Engineer in the automotive industry. In December 2006, he joined the Parallel and Distributed Computing Centre at NTU as Research Associate where he worked on projects concerned with simulation-based optimization and agent-based crowd simulation.

In December 2011, he joined TUM CREATE as Research Fellow where he assumed the role of Principal Investigator for the 'Modeling and Optimization of Architectures and Infrastructure' group in December 2012. His research is concerned with analyzing the potential impact of electromobility on the infrastructure and the environment from a complex systems perspective. His current research interests are agent-based simulation, complex adaptive systems, symbiotic simulation, evolutionary computing and simulation-based optimization.