



**HAL**  
open science

# A Shareability Clustering Method to Solve the Dynamic Ride-Sharing Problem Considering Network Congestion

Negin Alisoltani, Mostafa Ameli, Mahdi Zargayouna, Ludovic Leclercq

► **To cite this version:**

Negin Alisoltani, Mostafa Ameli, Mahdi Zargayouna, Ludovic Leclercq. A Shareability Clustering Method to Solve the Dynamic Ride-Sharing Problem Considering Network Congestion. TRB 2022, Transportation Research Board 101st Annual Meeting, Jan 2022, Washington, D.C, United States. hal-03680835

**HAL Id: hal-03680835**

**<https://hal.science/hal-03680835>**

Submitted on 29 May 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 **A SHAREABILITY CLUSTERING METHOD TO SOLVE THE DYNAMIC**  
2 **RIDE-SHARING PROBLEM CONSIDERING NETWORK CONGESTION**

3  
4  
5  
6 **Negin Alisoltani (corresponding author)**

7 🏠 GeoTwin SAS, Paris, France


8 🏠 Univ. Gustave Eiffel, COSYS-GRETTIA, Paris, France

9  <https://orcid.org/0000-0003-0458-3074>

10 ✉ [negin.alisoltani@geotwin.io](mailto:negin.alisoltani@geotwin.io)

11  
12 **Mostafa Ameli**

13 🏠 Univ. Gustave Eiffel, COSYS-GRETTIA, Paris, France

14  <https://orcid.org/0000-0002-2470-6812>

15 ✉ [mostafa.ameli@univ-eiffel.fr](mailto:mostafa.ameli@univ-eiffel.fr)

16  
17 **Mahdi Zargayouna**

18 🏠 University Gustave Eiffel, COSYS, GRETTIA, Paris, France

19  <https://orcid.org/0000-0002-1221-8977>

20 ✉ [mahdi.zargayouna@univ-eiffel.fr](mailto:mahdi.zargayouna@univ-eiffel.fr)

21  
22 **Ludovic Leclercq**

23 🏠 Univ. Gustave Eiffel, Univ. Lyon, ENTPE, LICIT, Lyon, France

24  <https://orcid.org/0000-0002-3942-6354>

25 ✉ [ludovic.leclercq@univ-eiffel.fr](mailto:ludovic.leclercq@univ-eiffel.fr)

26  
27  
28  
29 Word Count: 5949 words + 5 table(s) × 250 = 7199 words

30  
31  
32  
33 Paper submitted for presentation at the 101<sup>th</sup> Annual Meeting Transportation Research Board,  
34 Washington D.C., January 2022 to AHB45 committee on "Advances in modeling and traffic  
35 management for large-scale urban network"

36  
37  
38 Submission Date: May 29, 2022

**1 ABSTRACT**

2 One of the most challenging facets of dynamic ride-sharing problems is the automated matching  
3 process, especially in large-scale networks where the number of received requests is huge. Match-  
4 ing users to trips is very challenging in real-time since it must happen very quickly while it must  
5 ensure high quality solutions. In this paper, we rely on an exact solving method to ensure the  
6 quality of the solution, while using AI-based techniques to limit the number of requests that we  
7 feed to the solver. More precisely, we propose a clustering method based on a new space-time  
8 shareability function to put the most shareable trips inside separate clusters. Each cluster is then  
9 managed with a proposed heuristic framework in order to solve the matching problem inside each  
10 cluster. To validate our proposal, we employ the proposed method on the network of Lyon city in  
11 France, with half-million requests in the morning peak from 6 to 10 AM. The results demonstrate  
12 that the algorithm can provide high-quality solutions in a short time for large-scale problems.

13

14

15 *Keywords:* Dynamic ride-sharing, Clustering, Shareability function, Optimal fleet management.

## 1 INTRODUCTION

2 Ridesharing originated as a general concept where individual travelers share a vehicle for a trip  
3 and split travel costs with others that have similar itineraries and time schedules (Furuhata et al.,  
4 1). The spread of mobile devices and the development of Global Positioning System (GPS) make  
5 it possible for all the transport operators to adapt in real-time the transportation supply to travel  
6 demand (Zargayouna et al., 2). These options make it possible to have access to the vehicles' posi-  
7 tion at any time and perform the matching process of ride-sharing in real-time. These possibilities  
8 have led to the development and progress of a new type of ride-sharing which is called real-time  
9 ride-sharing, also known as dynamic ride-sharing or *ad hoc* ride-sharing. Dynamic ride-sharing  
10 refers to a system that supports an automatic ride-matching process between participants on very  
11 short notice or even en-route (Agatz et al., 3).

12 The first issue in this application is matching the passengers and the fleet vehicles by solv-  
13 ing a fleet management problem. Effective and efficient optimization technology that matches  
14 drivers and riders in real-time is one of the necessary components for a successful dynamic ride-  
15 share system (Agatz et al., 4). Matching users to trips is very challenging in real-time since it must  
16 happen very quickly. Clustering methods enhance the scalability of matching methods by reducing  
17 the search space, making it possible to parallelize and balance the computing workload, and as a  
18 consequence speeding up the matching computations.

19 In this paper, we propose a clustering method based on a “Shareability Function” (SF) that  
20 considers all the trips' possible matching situations. Two trips can be shared either in parallel (pas-  
21 sengers are in the same vehicle at some point) or in sequence (Alisoltani et al., 5). The shareability  
22 function computes the extra travel time that the vehicle has to spend to service each matching situa-  
23 tion, compared to the situation where each trip is serviced independently without sharing. Then the  
24 shareability function for each two trips is the minimum computed value among the three different  
25 situations (parallel, sequence and independence). So, when the shareability function is low for two  
26 specific requests, it means that they have high potential to be shared efficiently. We propose a clus-  
27 tering method based on this new function to put the most shareable trips in separate clusters. We  
28 propose a heuristic method to solve the matching problem inside each cluster. The final algorithm  
29 can provide high quality solutions in a short time for large-scale problems.

30 It is crucial to consider the interactions between the system and the network traffic in dy-  
31 namic ride-sharing system evaluation. Also, it is important to consider other vehicles like personal  
32 cars besides the ride-sharing service vehicles in the network. In this paper, we define two differ-  
33 ent models to simulate the functioning of the proposed dynamic ride-sharing system: The “plant  
34 model”, based on Macroscopic Fundamental Diagram (MFD), is used to simulate the real traffic  
35 conditions and considers both service vehicles and personal vehicles in the network; The “pre-  
36 diction model”, based on the current mean speed, is used to calculate the travel times during the  
37 assignment process (Alisoltani et al., Alisoltani et al., 5, 6).

38 To validate our approach, we employ the proposed method on the real data of Lyon city in  
39 France with half-million requests in the morning peak from 6 to 10 AM. To demonstrate the clus-  
40 tering method's performance, we have compared the proposed method with two other clustering-  
41 based methods from the literature (spatial and temporal clustering). The results show that the  
42 method proposed in this paper can make a significant improvement in the quality of the solution  
43 and the computation time.

44 Since ride-shares are established on-demand, a ride-sharing system is similar to other on-  
45 demand forms of passenger transit such as taxis and dial-a-ride services like airport shuttles (Fu-

1 [ruhata et al., 1](#)). The clustering method proposed in this study can be used for other similar systems.  
2 The remainder of this paper is structured as follows. Next section is the literature review  
3 on ride-sharing and clustering methods. In the third section, we explain our clustering method for  
4 dynamic ride-sharing. Then we define our travelers matching algorithm and describe the two simu-  
5 lation models used as plant and prediction models respectively. In section "Experiments" we present  
6 the case study we used and provide the results of our experiments. The last section concludes this  
7 paper.

## 8 LITERATURE REVIEW

9 In this section, we review the latest studies on different solution methods and clustering approaches  
10 for real-time and dynamic ride-sharing.

11 Dynamic ride-sharing problem addresses short-term matching or even en-route matching  
12 ([Agatz et al., 3](#)). This fact makes the assignment problem more complex. In some studies, re-  
13 searchers try to narrow the feasible solution space to make the computations faster and be able to  
14 assign the vehicles to the requests that are coming at each time to the system. For example, [Liu](#)  
15 [et al. \(7\)](#) present a method to tighten travelers time windows and eliminate unnecessary variables  
16 and constraints to narrow the solution space.

17 Recently [Mourad et al. \(8\)](#) has presented a survey of models and algorithms for optimizing  
18 shared mobility, and they have shown that one of the most critical problems in the solution for  
19 these systems is computation time and the solutions quality.

20 Compared to these approaches, this paper proposes a method that combines the benefits  
21 of exact solving in terms of solution quality and the benefits of heuristics and meta-heuristics in  
22 terms of computation time and scalability. On the one side, we present an algorithm based on  
23 branch-and-bound that can provide the optimal solution for small instances of the problem. On the  
24 other side, we introduce a method based on clustering that groups the requests and put the most  
25 shareable trips in the same clusters. The exact algorithm is executed with the requests inside each  
26 cluster. A rolling horizon approach is introduced to handle the trip requests in real-time. We show  
27 that the clustering method provides high-quality solutions while reducing the computation time  
28 significantly.

29 The number of request for the mobility service at every time is huge in the large-scale prob-  
30 lems and the complexity of the solution method for the matching problem increases exponentially  
31 by small increase in the service demand. It is substantial to consider that the patterns of demands  
32 and the patterns of supplies are spatially and temporally dependent ([Wang et al., 9](#)). Consider-  
33 ing these dependencies, clustering methods can be applied to scale down the problems and make  
34 the computations faster. In the literature, there are clustering methods to handle large-scale prob-  
35 lems, like dividing the time into several time slots or dividing the space into several clusters, road  
36 segments, or cells ([Davis et al., Qi and Liu, 10, 11](#))

37 To address the ride-matching problem in large-scale configurations, [Pelzer et al. \(12\)](#) pro-  
38 pose to partition the road network into distinct regions which represent certain sub-structures of  
39 the road network. [Zuo et al. \(13\)](#) propose a clustering-based request matching and route plan-  
40 ning algorithm whose basic operations are merging requested trips on road networks. [Qiang and](#)  
41 [Shuang-Shuang \(14\)](#) propose an algorithm that uses the dataset of taxi get-off points and performs  
42 a clustering of taxis on urban roads. They compare their method with classical clustering methods.  
43 However, the taxi clustering data in their study are conducted in a static environment. In [Chen et al.](#)  
44 [\(15\)](#), all pickup points are partitioned into several clusters, the vehicles dispatching and the ride-

1 sharing problem are solved in each cluster. Bard and Jarrah (16) show that an appropriate solution  
 2 for large-scale problems is clustering the demand nodes and downsizing the network. To speed up  
 3 the computation, Some researches try to limit the feasible region with clustering methods. In these  
 4 methods, the demand nodes in the network are usually divided into geographically dense clusters  
 5 (Özdamar and Demir, Sáez et al., 17, 18). Li and Chung (19) propose an extended insertion algo-  
 6 rithm in conjunction with a Tabu search method, and a cluster-first-route-second approach to find  
 7 heuristic solutions.

8 Santi et al. (20) introduce the notion of a shareability network to quantify the spatial and  
 9 temporal compatibility of individual trips in a dynamic environment. In this network, two trips are  
 10 shareable if they would incur a delay of no more than five minutes. Then, Vazifeh et al. (21) address  
 11 the minimum fleet problem in on-demanded urban mobility. They enrich the idea of shareability  
 12 network to model the sharing of vehicles instead of rides. In these clustering methods, the trips  
 13 are clustered based solely on the situation of the origin points. However, in ride-sharing, other  
 14 combinations of trips should be considered. In this paper, we propose the concepts of "sequential  
 15 index" and "Shareability index" to assess the possibility of serving two trips with the same car  
 16 in sequence or by sharing the trips. Our proposal employs a method that reduces the number of  
 17 required vehicles.

18 Under the assumption of one person per ride, Alonso-Mora et al. (22) show that 98% of the  
 19 taxi rides currently served by over 13,000 taxis could be served with just 3,000 taxis of capacity  
 20 four in New York City. The proposed clustering method in this paper, seeks the potential trips to  
 21 be shared in the same vehicle and makes sure that the matching algorithm can use the maximum  
 22 capacity of a vehicle to serve the requests for a specific route.

23 In a recent study, Yu et al. (23) consider the combination of the ride-sharing and public  
 24 transportation services and formulate a mixed integer programming model for the multimodal  
 25 transportation planning problem. They propose a heuristic approach (i.e., angle-based clustering  
 26 algorithm) and compare its efficiency with the exact solution for different settings. They show  
 27 that the clustering algorithm works well both in small and large settings. However, it is crucial to  
 28 consider the interactions between the system and the network traffic in the ride-sharing problem.  
 29 In this paper, we define two distinct models for the dynamic ride-sharing to calculate the travel  
 30 times during the assignment and simulate real traffic conditions.

## 31 CLUSTERING METHOD FOR DYNAMIC RIDE-SHARING

### 32 Shareability function

33 To perform the clustering on the requests ( $N_t$ ) received by the system at time  $t$ , we define the  
 34 "Shareability Function" ( $SF_{i,j}$ ) between request  $i$  and request  $j$  ( $\forall i, j \in N_t$ ). This function computes  
 35 the difference between the travel time when the two trips are shared and the travel time to serve  
 36 each trip individually, for each pair of requests.

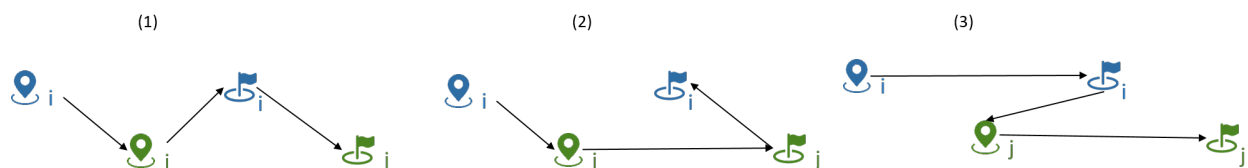


FIGURE 1 : Trip situations

1 In case of sharing, we consider three different situations for each pair of trips (Fig. 1). In the  
 2 first situation (1), two trips can be shared, and the vehicle first drops off the first passenger, and then  
 3 it goes to the second passenger drop off (destination) point. In this situation, the travel time for the  
 4 first passenger ( $Tp_i$ ) would be the summation of her/his waiting time ( $Ws_i$ ), the travel time between  
 5 her/his origin and the next passenger origin ( $TOO_{i,j}$ ) and the travel time between the next passenger  
 6 origin and her/his destination ( $TOD_{j,i}$ ). Similarly, the travel time for the second passenger ( $Tp_j$ )  
 7 would be the summation of her/his waiting time ( $Ws_j$ ), the travel time between her/his origin to  
 8 the first passenger's destination ( $TOD_{j,i}$ ) and the travel time from the first passenger's destination  
 9 and her/his destination ( $TDD_{i,j}$ ). So,  $SF_{i,j}$  (shareability function for trips  $i$  and  $j$ ) when two trips  
 10 have the first situation can be computed as in equation 1.

$$\begin{aligned} Tp_i + Tp_j &= Ws_i + TOO_{i,j} + Ws_j + TOD_{j,i} + TDD_{i,j} \\ \forall i, j \in N \\ SF_{i,j}^1 &= Tp_i + Tp_j - (TOD_{i,i} + TOD_{j,j} + W_i + W_j) \end{aligned} \quad (1)$$

11 In situation (2), the second passenger is served by the service vehicle while the first passen-  
 12 ger is on board. Thus, the travel time for the second passenger ( $Tp_j$ ) is the same as when served  
 13 individually (summation of the waiting time and the time from the origin to the destination) and  
 14 the travel time for the first passenger is the travel time of all the links from the first stop point ( $O_i$ )  
 15 to the last one ( $D_j$ ):

$$\begin{aligned} Tp_i + Tp_j &= Ws_i + TOO_{i,j} + Ws_j + TOD_{j,j} + TDD_{j,i} \\ \forall i, j \in N \\ SF_{i,j}^2 &= Tp_i + Tp_j - (TOD_{i,i} + TOD_{j,j} + W_i + W_j) \end{aligned} \quad (2)$$

16 In the third situation (3) that we consider for two trips, the trips are not shared, but the ve-  
 17 hicle can serve two passengers sequentially (Alisoltani et al., 5). This situation must be considered  
 18 in the shareability index in order to encourage putting these trips in the same group while solving  
 19 the optimization problem. The travel time for both passengers in this situation is the same as when  
 20 they are served individually. But the vehicle travel time can decrease if the travel time between  
 21 the first destination and the second origin is less than the summation of the travel time between the  
 22 first origin and the closest depot and the travel time between the start depot and the second origin.

$$\begin{aligned} Tp_i + Tp_j &= Ws_i + TOD_{i,i} + Ws_j + TOD_{j,j} \\ \forall i, j \in N \\ SF_{i,j}^3 &= Tp_i + Tp_j - (TOD_{i,i} + TOD_{j,j} + W_i + W_j) \\ SF_{i,j}^3 &= Ws_i + Ws_j - (W_i + W_j) \end{aligned} \quad (3)$$

23 The best situation for each two trips is the situation with minimum  $SF$ . So the algorithm  
 24 chooses the condition that the additional travel time is minimum for sharing each pair of trips:

$$SF_{i,j} = \min\{SF_{i,j}^1, SF_{i,j}^2, SF_{i,j}^3\} \quad (4)$$

## 1 Clustering based on a dissimilarity function

2 After computing the shareability function, we have the function value for each pair of requests that  
3 creates a “shareability matrix”. The shareability matrix is a dissimilarity matrix for the received  
4 requests: the higher the value, the least likely the two requests will be served together. In our  
5 approach, this matrix is used in the clustering process. We perform the clustering using the com-  
6 puted dissimilarity matrix. When we make clusters based on the  $SF$ , we put in the same cluster  
7 the trip requests that have more potential to be shared (the trips that have lower  $SF$ ). There are two  
8 main categories of algorithms for clustering based on the shareability matrix, both are potentially  
9 relevant for our large-scale ride-sharing application:

- 10 1. Partitional clustering algorithms (Celebi, 24) cluster the data into  $k$  clusters. One of the  
11 usual algorithms for partitioned clustering is k-means clustering. K-means clustering is  
12 simple, fast, and flexible.
- 13 2. Hierarchical clustering methods in which the clusters are arranged in a tree-like struc-  
14 ture. Hierarchical clustering can be divided into Agglomerative hierarchical clustering  
15 (AHC) and divisive clustering (Vora et al., 25).

16 In Qiang and Shuang-Shuang (14), the authors have compared the hierarchical clustering and k-  
17 means clustering for urban taxi carpooling in a static environment. They show that compactness  
18 and separation are almost the same for hierarchical and k-means clustering for large cluster sizes.  
19 However, in dynamic large-scale problems, the results might be different. Besides, computation  
20 time becomes the critical criterion in this context. Thus, in our experiments, we implement both  
21 clustering methods and we assess their performance, considering both solutions quality and com-  
22 putation time.

### 23 *Multidimensional scaling and k-means clustering*

24 K-Means method is a partitional clustering approach for decomposing the problem into indepen-  
25 dent subsets. It defines clusters of data based on their similarity, minimizing within-cluster vari-  
26 ances. In the clustering procedure, the preferred number of clusters ( $k$ ) should be specified in the  
27 algorithm before execution. The most common algorithm uses an iterative refinement technique.  
28 A common initialization step associates each observation (data point) with a cluster, and computes  
29 the set of cluster centroids as the mean of the observations of the cluster. Afterward, iterations will  
30 serve to optimize the clusters.

31 k-means clustering takes place based on the distance between points. Based on the study  
32 in Paea and Baird (26), to be able to apply the appropriate k-mean clustering method, we need to  
33 convert the dissimilarity matrix into a distance matrix. Therefore, we use the multidimensional  
34 scaling method and change the shareability matrix into a distance matrix (Wang and Boyer, 27).  
35 Note that we have implemented the method using the *mathtoolbox* in C++).

36 After extracting the distance matrix, we are able to create the same size clusters for the  
37 data received at every assignment time step and put the most shareable trips in separate clusters  
38 using the modified k-means clustering method. This k-mean clustering is used to obtain clusters  
39 in preferred sizes (Ganganath et al., 28). Accordingly, considering the objective function value,  
40 we can find the best trade-off between cluster size and computation time. Putting the requests in  
41 clusters can also provide the opportunity for parallel computations. So the problem is divided into



1 multiple sub-problems, and we favor a uniform distribution of requests among clusters to decrease  
2 computation times and facilitate parallel computations of each sub-problem.

### 3 *Hierarchical clustering*

4 Hierarchical clustering offers a flexible and no-parametric approach and is an algorithm that builds  
5 hierarchy of clusters (Murtagh, 29).

6 We use the agglomerative hierarchical method, which starts with taking singleton clusters  
7 (that contain only one request per cluster) at the bottom level and continue merging two clusters at  
8 a time to build a bottom-up hierarchy of the clusters (Reddy and Vinzamuri, 30).

## 9 **MATCHING ALGORITHM FOR DYNAMIC RIDE-SHARING**

10 The matching algorithm aims to identify the travelers who can share their trips and assign them  
11 to a vehicle. In the assignment, it is important to consider both passengers' and service provider's  
12 objectives. According to the state-of-the-art, the most important operation objective for the service  
13 provider is to minimize the total travel time, and the total travel distance of vehicles (Ota et al., Qian  
14 et al., 31, 32). The passengers also need to get to the destination on time and have the minimum  
15 waiting time (Hyland and Mahmassani, 33). So we define the objective function for the matching  
16 algorithm as below where  $i$  is the index of passengers, and  $m$  is the index of vehicles,  $W_i$  is the  
17 waiting time for passenger  $i$ ,  $T p_i$  is the travel time for passenger  $i$ ,  $T v_m$  is the travel time for  
18 vehicle  $m$ ,  $D_m$  is the travel distance for vehicle  $m$  and  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are the weights of each  
19 objective after normalization.

$$\min \sum_{i \in P} (\alpha \cdot W s_i + \beta \cdot T p_i) + \sum_{m \in M} (\gamma \cdot T v_m + \delta \cdot D_m) \quad (5)$$

20 The main constraints for the matching process are capacity constraints, time constraints,  
21 and assignment constraints. In addition, we have considered limitations on the number of sharing.  
22 The number of sharing for each passenger defines the allowed number of other passengers that  
23 can share their trip with this current passenger. So when the number of sharing is zero, it means  
24 that the service is not able to share and we just consider the in-sequence trips in the clustering  
25 function. But when we increase the number of sharing to one, it means that the service can serve  
26 2 passengers at the same time. Also, we made sure that there is a sufficient number of vehicles in  
27 the fleet to serve all the requests.

28 The vehicle can be in two different situations. It is either circulating in the network to serve  
29 the on-board passengers (en-route vehicle) or waiting at a stop location for the new passengers  
30 (idle vehicle). Our proposed algorithm considers both situations. When the system receives a new  
31 request, it first checks the en-route vehicles that have not yet reached their maximal occupancy.  
32 So, the first part of the algorithm assigns the new requests in priority to en-route vehicles. The  
33 algorithm works to minimize the total travel time for both vehicles and passengers while respecting  
34 the capacity of the vehicle's constraint and passengers' time window constraint. For the remaining  
35 requests, the second part of the algorithm checks the idle vehicles waiting in depots to assign the  
36 passengers. For this step, we introduce an exact method based on the branch-and-cut concept.  
37 The method creates branches of routes and then chooses the optimal solution among the feasible  
38 routes. Fig. 2 shows this part of the algorithm. Then we introduce a rolling horizon method to  
39 solve the problem dynamically. The requests that are assumed to be known are the only ones over  
40 the next rolling horizon (20 minutes). We also use an "assignment time horizon" of 10 minutes.

1 That means that every 10 minutes, we execute our optimization algorithm while considering the  
2 requests of the next 20 minutes.

3 We present a small example with four requests to show the execution of the algorithm. Ta-  
4 ble 1 presents the requests. Each request has an associated number of requested seats, a maximum  
5 number of accepted sharing, an earliest pick-up time and a latest arrival time.

**TABLE 1 : Example with 4 requests (configuration)**

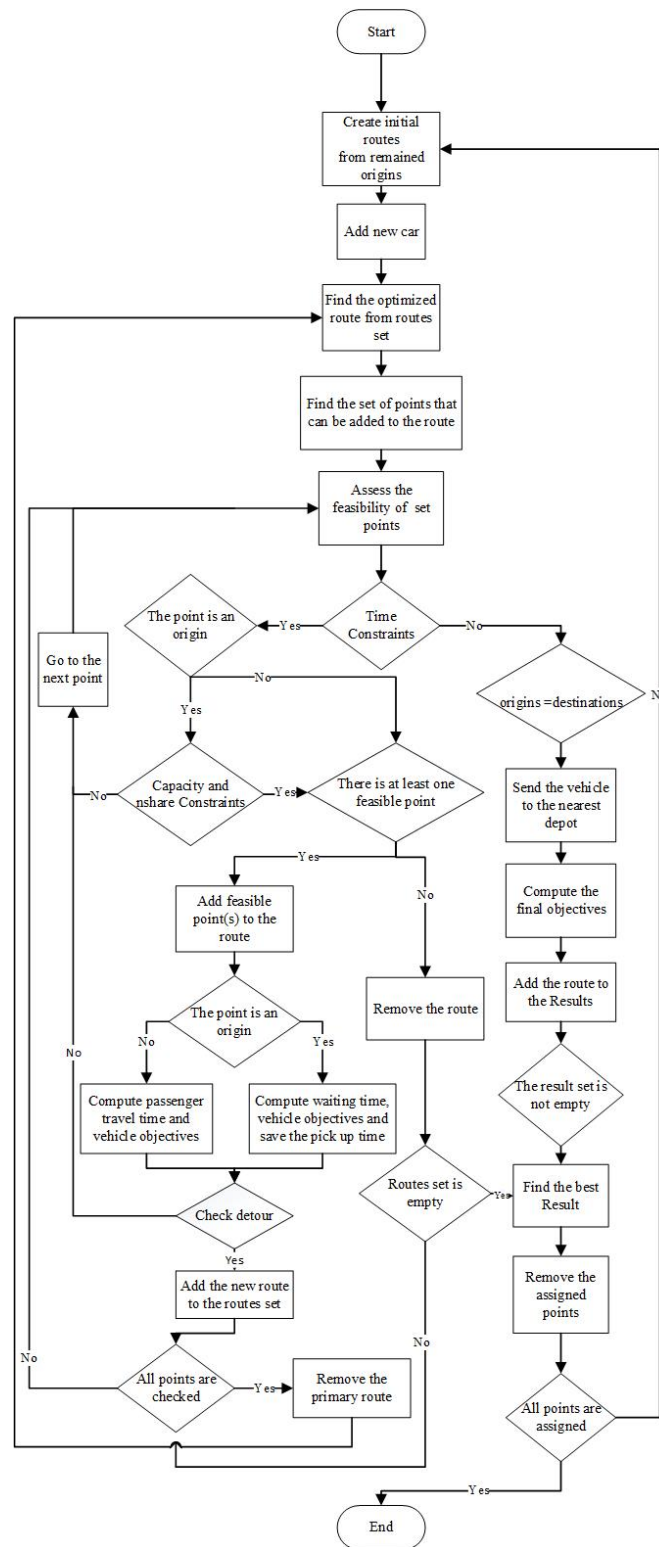
Request	Travel distance	demand	nshare	EPT	LDT
1	19	1	3	8:00	8:45
2	11	2	2	8:00	8:25
3	24	2	1	8:15	9:00
4	18	2	3	8:30	9:20

6 The algorithm starts by creating branches of routes to serve the requests. It sends a car  
7 from the closest stop location to pick the passengers up at the origin point, and then it continues  
8 by adding the feasible points to the branches. First, the algorithm finds feasible branches. Figure  
9 3 shows the final feasible solutions for the problem. The algorithm can find four feasible solutions  
10 for the problem.

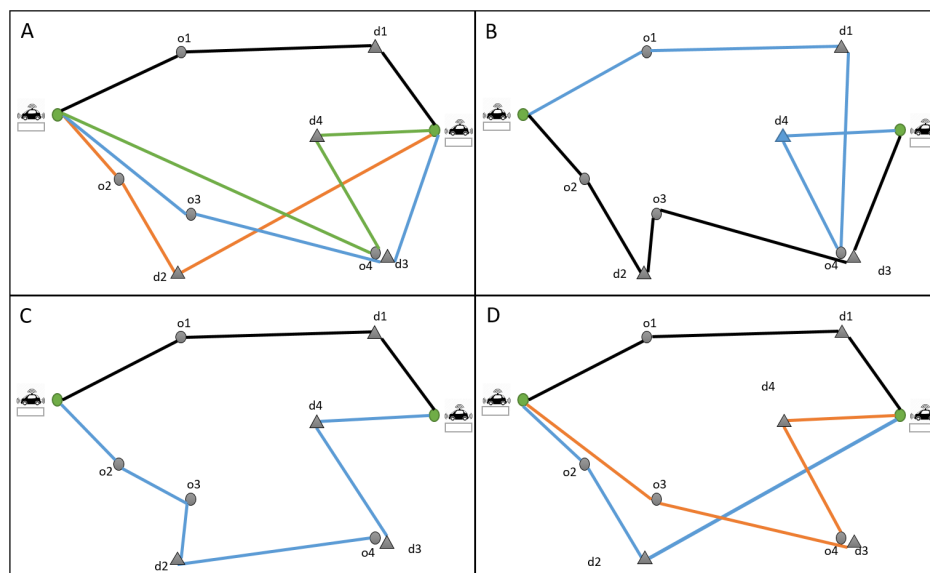
**TABLE 2 : Example with 4 requests (solutions)**

Solution	Total waiting time (s)	Total travel distance (m)	Number of cars
A	28	3030	4
B	51	2085	2
C	53	1785	2
D	29	2265	3

11 Table 2 shows the total passengers' waiting time, total vehicles' travel distance, and the  
12 number of vehicles for these solutions (note that the waiting time for passengers is the difference  
13 between the passengers' pick up time and the passengers' desired departure time). Solution *A* is  
14 when a car serves each passenger separately without sharing. In this solution, the waiting time is  
15 minimum, and the passengers wait for 28 seconds to be picked up. Solution *B* serves passengers  
16 1 and 4 with one car and passengers 2 and 3 with another car. It should be mentioned that in  
17 this solution, the trips are not shared. The first car serves passenger 1, and after dropping off this  
18 passenger, it goes to pick up passenger 4 at his destination. This solution increases the waiting time  
19 to 51 seconds, but it can reduce the travel distance from 3,030 meters to 2,085. Solution *C* serves  
20 the four requests with two cars like solution *B*. However, it shares the trip for passengers 2 and  
21 3, and it can make more significant progress in reducing the travel distance to 1,785 meters. The  
22 waiting time with this solution is 53 seconds. Solution *D* increases the waiting time just 1 second  
23 compared to solution *A*. It serves requests 3 and 4 with a car in sequence (without sharing) and  
24 requests 1 and 2 with two cars separately. If the weights for all the objective functions are equal,  
25 after normalizing the objective functions, the optimal solution will be solution *C*.



**FIGURE 2 : Final assignment algorithm**



**FIGURE 3 : Feasible routes for the example**

### 1 SIMULATION MODELS FOR DYNAMIC RIDE-SHARING

2 The ride-sharing service's optimization system uses estimates for the predicted travel time obtained  
 3 from a so-called "prediction model". When the fleet management plan is executed, a gap usually  
 4 exists between the estimation and the real traffic condition. The so-called "plant model" requires  
 5 dynamic adjustment of the initial assignment to fit with the conditions observed, and we use it to  
 6 represent the real traffic condition.

7 To provide a realistic service, in the proposed simulation component of the dynamic ride-  
 8 sharing system, we accurately distinguish the prediction and the plant models.

9 The trip-based MFD is used as the plant model (Ameli et al., 34) to consider individual  
 10 trips while keeping a very simple description of traffic dynamics (Lamotte and Geroliminis, Ameli,  
 11 35, 36). The general principle is to derive the inflow and outflow curves. When  $n(t)$  is the number  
 12 of en-route vehicles at time  $t$  and the mean speed of travelers is  $V(n(t))$  at every time  $t$ , the travel  
 13 distance  $L_i$  by a car  $i$  entering at time  $t - T(t)$  must satisfy the following equation:

$$L_i = \int_{t-T(t)}^t V(n(s)) ds \quad (6)$$

14 The function  $V(n(t))$  is the speed macroscopic fundamental diagram and can be derived  
 15 from common observations for a transportation network (Leclercq et al., 37). For more details on  
 16 the functioning of trip-based MFD, readers can refer to (Mariotte and Leclercq, 38).

17 The prediction model estimates the traffic situation for the next assignment time horizon  
 18 (every 10 minutes) to carry out travel time prediction during optimization. In our method, we  
 19 assign the passengers to the cars based on this prediction. The prediction model is based on the  
 20 direct travel time from each point  $i$  to  $j$  based on the *current mean speed* and the associated shortest  
 21 path between the two points for the next 10 minutes. Then the optimization algorithm assigns all  
 22 the requests for the next 10 minutes to the en-route cars or empty waiting cars.

23 In the rolling horizon method, the assignment procedure rolls over a specific horizon for  
 24 the requests announced of a particular optimization step. As stated earlier, the rolling horizon in

1 this paper is 20 minutes, and the optimization time step is 10 minutes. So, the requests of the next  
2 20 minutes that have not yet been assigned are optimized every 10 minutes. Some requests are  
3 re-optimized every 10 minutes. If a trip has been assigned to a vehicle which has left the depot, the  
4 algorithm does not assign it again, but if a trip is in the schedule of a waiting vehicle, the algorithm  
5 places it in the set of optimized trips in the particular horizon and re-optimizes it.

6 In this method, every  $\frac{TH}{2}$  time step, we stop the simulations and solve a new assignment  
7 problem for the requests over a new full rolling horizon ( $TH$ ). Some requests may arrive just after  
8 the end of a simulation period, and this method prevents the system from being myopic to the new  
9 demand.

## 10 EXPERIMENTS

### 11 Case study

12 In this paper, we use data from the network of Lyon city (the second-largest urban area of France)  
13 to evaluate the impact of proposed clustering method. The network area is more than  $80 \text{ km}^2$   
14 and the origins/destinations (ODs) set contains 11,314 points (Ameli et al., 39). Fig. 4 shows the  
15 network of the city.



**FIGURE 4 : Network of Lyon**

16 The network is loaded with travelers of all ODs with a given departure time to represent the  
17 morning peak hour from 6 AM to 10 AM. The number of trips during this period is 484,690. We  
18 have 279,382 personal trips in the network and 205,308 demand for the service cars in the system.

19 The service provider has a fleet of vehicles in a ride-sharing system to serve the service re-  
20 quests. Participating service vehicles start up from a number of known locations or depots and after  
21 serving the assigned requests, they stop at this allowed locations to wait for the next passengers.

### 1 Determining the proper clustering method

2 As we said in section 4.2, we can use either the k-means clustering method or the hierarchical  
 3 method to cluster the requests based on the presented shareability matrix. In our method, both  
 4 the quality of the clustering method and the computation time are very important. The time com-  
 5 plexity of k-means is linear, while that of hierarchical clustering is quadratic. Besides, k-means  
 6 clustering requires prior knowledge of number  $k$  of clusters and also needs to use mutidimensional  
 7 scaling to convert the similarity matrix into the distance matrix. On the other side, we can stop  
 8 at whatever number of clusters we find appropriate in hierarchical clustering by interpreting the  
 9 dendrogram. As we use the agglomerative hierarchical method, we can have larger clusters faster  
 10 with the hierarchical method.

**TABLE 3 : Clustering methods comparison**

Method	Number of requests	Objective function (normalized value)	Computation time (s)
Exact			
	112	29.99	12966.00
	1092	1924.00	288000.00
K-means clustering			
	112	30.96	100.10
	1092	1994.18	1134.00
	4482	7761.03	5340.00
	11160	20773.70	20981.10
Hierarchical clustering			
	112	30.96	99.03
	1092	1994.59	1131.10
	4482	7791.90	5187.50
	11160	20905.87	19950.00

11 To choose the best clustering method, we have compared both methods considering the  
 12 quality of objective function and the computation time for different sizes of problems.

13 Table 3 shows the objective function and computation time with k-means clustering and  
 14 hierarchical clustering method for different sizes of problems.

15 To have a baseline for the comparisons, we have computed the optimal objective function  
 16 (solution method without clustering) for 112 and 1,092 requests. In both clustering methods, we  
 17 try to have clusters with 50 requests to ensure potential trips to be shared inside the clusters for  
 18 both test cases with 50 requests and 1092 requests.

19 K-means clustering and hierarchical clustering methods increase the objective function by  
 20 3.22% for 112 requests and 3.64% and 3.66% for 1,092 requests, compared to the optimal solu-  
 21 tion. Both clustering methods can decrease the computation time from 216 minutes to less than 2  
 22 minutes. This shows that both clustering methods are very effective in terms of reducing the com-  
 23 putation time while keeping the quality of the solution acceptable. Then, by increasing the number  
 24 of requests, the computation time for both methods exponentially increases (the major part of the  
 25 k-means method computation time is dedicated to the multidimensional scaling method, which  
 26 exponentially increases by increasing the size of the problem).

27 K-means can give smaller objective function while hierarchical computation can result in

1 a lower time. For 11,160 requests, the objective function is 0.64% lower for the k-means method,  
2 while the computation time is 5% more.

3 In our Lyon6 + Villeurbanne test case, the maximum number of requests is 11,235. So  
4 we can use k-means clustering for this test case to have better solutions. However, for the Lyon  
5 network, we have more than 200,000 service requests. So the hierarchical method is a better  
6 solution for this network, since it is faster as it works directly with the similarity matrix, and it can  
7 provide high-quality solutions.

### 8 **The size of clusters**

9 We try to have the same size clusters (to avoid too big or too small clusters) to keep the computation  
10 time low and have the opportunity for sharing in all the clusters.

11 There are different methods in the literature to choose the optimal size of clusters. Here,  
12 the quality of the clusters (how similar are points within a cluster) is very important. Furthermore,  
13 we have to be sure that the clusters are separated from each others, and the possibility of sharing  
14 two trips from two different clusters is minimum. Thus the best way to find the optimal size of  
15 clusters is to use the Sum of Squares method (SS) (Krzanowski and Lai, 40). It is a clustering  
16 validation method that chooses the optimal size of clusters by minimizing the Within-cluster Sum  
17 of Squares (WSS) (a measure of how tight each cluster is) and maximizing the Between-cluster  
18 Sum of Squares (BSS) (a measure of how separated each cluster is from the others). We compute  
19 the WSS and BSS for all the clusters in different periods to evaluate the optimal size of clusters in  
20 different demand situations.

### 21 *K-means clustering*

22 We compute the sum of squares for k-means clustering with cluster sizes from 10 to 50. Fig. 5  
23 shows that when the size of clusters is 30, we can find the best trade-off between WSS and BSS.  
24 So, we choose 30 as the size of clusters when we want to cluster the service requests with k-means  
25 clustering.

**TABLE 4 : Sum of Squares method for k-means clustering**

Number	Size of clusters
1	10
2	20
3	30
4	40
5	50

### 26 *Hierarchical clustering*

27 In the hierarchical method, we can keep the first cluster of the desired size at the bottom of the  
28 dendrogram to have the same size clusters (Hippocampus, 41). The crucial point here is to find  
29 an approximation for the size of clusters in the hierarchical method. As we have explained earlier,  
30 we use the hierarchical clustering when the number of demand is huge. So, we analyze the size  
31 of clusters from 75 to 300. Also, in large-scale networks with high levels of traffic congestion, the  
32 demand density is different in different times of a day (during the on-set and off-set of congestion).



**FIGURE 5 : Sum of squares method for finding the optimal size of clusters in k-means clustering**

- 1 So we perform the analysis for the hierarchical clustering in four different times of the morning
- 2 peak to be able to decide about the clusters size considering different conditions.

**TABLE 5 : Sum of Squares method for hierarchical clustering**

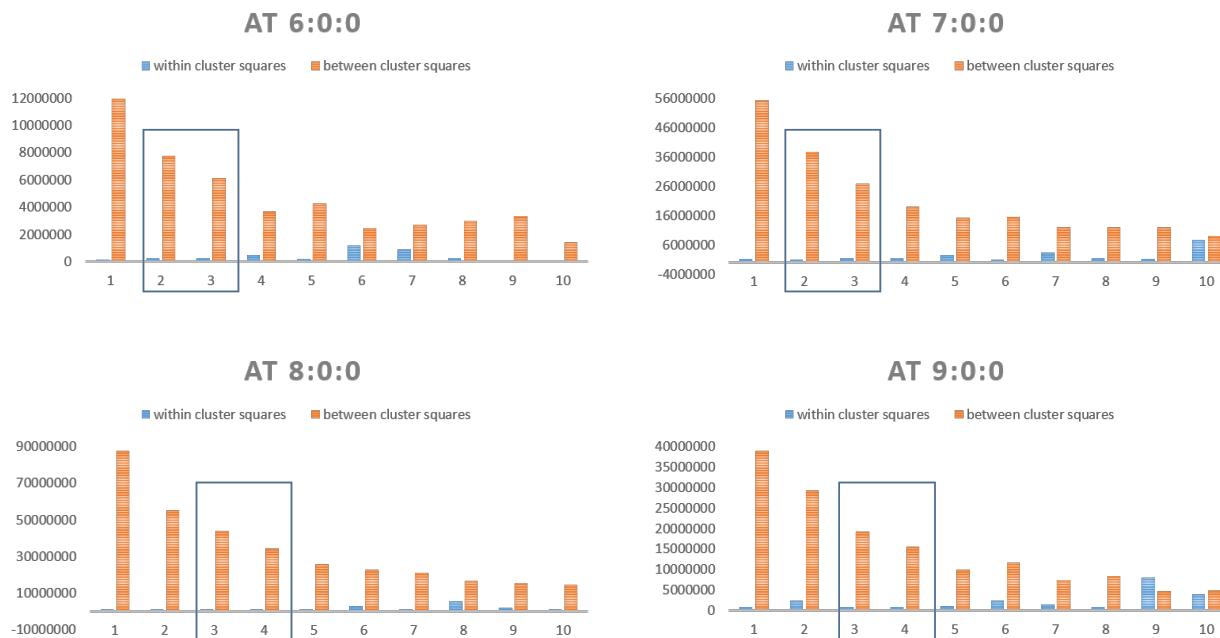
Number	Size of clusters
1	75
2	100
3	125
4	150
5	175
6	200
7	225
8	250
9	275
10	300

3 Fig. 6 shows the SS method at different times of the simulations. Increasing the size of  
 4 clusters decreases the BSS. It means that more number of clusters can ensure that the clusters  
 5 are separate from each others. We have determined the cluster sizes that minimize the WSS and  
 6 maximize the BSS. At 6 and 7 AM, the cluster sizes 100 and 125 can make this trade-off between  
 7 WSS and BSS. At 8 and 9 AM, the best cluster sizes are 125 and 150. Therefore, the best cluster  
 8 size to do the simulations for this scale is 125.

9 *Comparing the shareability clustering method with spatial and temporal methods*

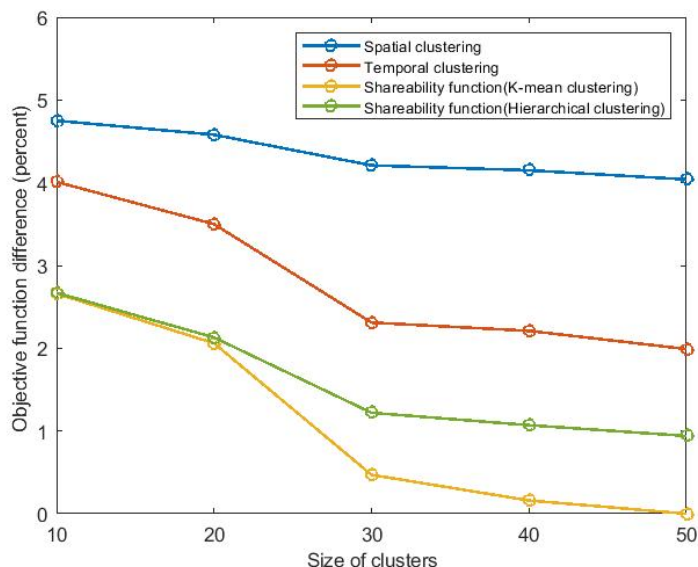
10 Some researches on ride-sharing use clustering methods to scale-down the matching problem and  
 11 compute the assignment for the service vehicles faster. They usually divide the space geographi-  
 12 cally and use a spatial clustering to downsize the problem (Chen et al., 15). The important factor  
 13 in spatial clustering is the distance between the trips’ origins. So two corresponding trips can be  
 14 in the same cluster if the distance between their origins is small. Another approach is to cluster  
 15 the trips based on the time in a temporal clustering method. In the temporal clustering, we put two





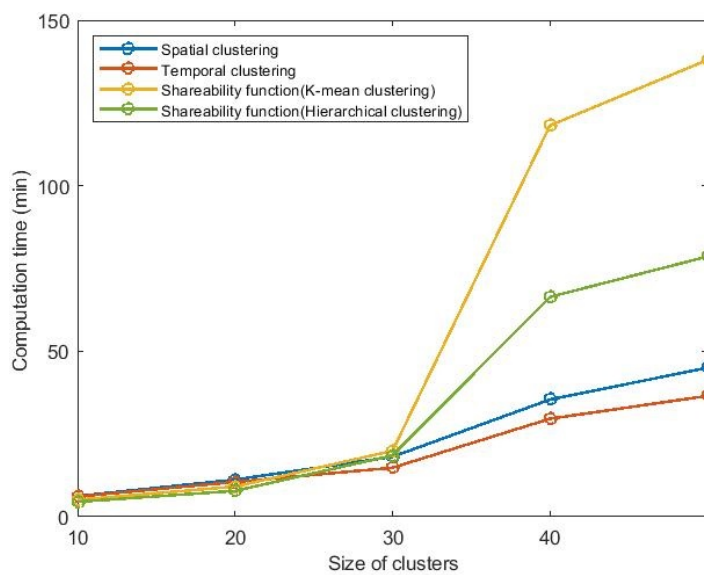
**FIGURE 6 : Sum of squares method for finding the optimal size of clusters in hierarchical clustering**

- 1 trips is the same cluster based on their departure time and their position. Here, we compare the
- 2 proposed clustering based on shareability function with spatial clustering and temporal clustering
- 3 methods to show the quality of our proposal.



**FIGURE 7 : Comparing clustering methods' objective function (market-share=50%)**

- 4 We compare the objective function and the computation time for the existing methods in the
- 5 literature and our proposed shareability clustering method. Figures 7 and 8 show the comparison



**FIGURE 8 : Comparing clustering methods' computation time (market-share=50%)**

1 for five different cluster sizes when the market-share is 50% (market-share is the percentage of the  
 2 total network demand that can be served by the ride-sharing service). Our proposed shareability  
 3 function with k-means clustering method can provide the best objective function (when the size  
 4 of clusters is 50 for this market-share). So the objective function for the shareability function and  
 5 k-mean clustering is considered as a base, and the percentage of difference for other methods is  
 6 computed considering this basic scenario in the first figure. The performance of spatial clustering  
 7 is poor compared to the other methods. Even in the best situation, the spatial clustering's objec-  
 8 tive function is 4.04% more than the shareability method with k-means clustering. The temporal  
 9 clustering can perform better than spatial clustering, but it can not outperform our (space-time)  
 10 shareability clustering method. The objective function for temporal clustering is 1.99% more than  
 11 k-means clustering when the size of clusters is 50. As we have to convert the shareability matrix  
 12 into a distance matrix using multidimensional scaling method to be able to use the k-means clus-  
 13 tering, for big clusters (like cluster size of 50 here) the computation time increases exponentially.  
 14 However, with the cluster size of 30, the algorithm can give a high-quality solution in a short time.  
 15 As we can apply the hierarchical clustering directly on the shareability matrix to put the shareable  
 16 trips in different clusters, the computation time for shareability function with hierarchical cluster-  
 17 ing is small. But the k-mean clustering can perform better in terms of quality of the clusters with  
 18 our proposed shareability function. Choosing the proper clustering method for the shareability  
 19 function highly depends on the scale of the problem and the demand density, as multidimensional  
 20 scaling is time-consuming and significantly reduces the possibility to use k-means with a large  
 21 number of vehicles.

## 22 CONCLUSION

23 In this paper, we proposed a new clustering method based on a space-time shareability function  
 24 for large-scale real-time ridesharing. The method puts the most shareable trips together inside the  
 25 same clusters. We implemented both partitioned clustering (k-means clustering) and hierarchical

1 clustering methods on the shareability matrix to cluster the requests. Then, we described a method  
2 to solve the matching problem inside each cluster. To evaluate the method, we employed the  
3 proposed method on the network of Lyon city in France with half-million requests in the morning  
4 peak from 6 to 10 AM. The results showed that the proposed clustering method produces high-  
5 quality solutions close to the optimal situation while reducing the computation time significantly.  
6 The clustering method proposed in this study can be used for other similar systems, such as taxis  
7 and dial-a-ride services.

8 We have used earlier versions of this method to assess the impact of real-time ridesharing  
9 on road transportation networks (Alisoltani et al., 42). In our future works, we plan to assess its  
10 impact on large-scale multimodal networks, i.e. considering all available modes of transportation,  
11 and not only private cars using GeoTwin multi-agent simulation platform (Bauguion et al., 43).

## 12 AUTHOR CONTRIBUTION STATEMENT

13 All the authors have contributed to all aspects of this study, ranging from the conception and design  
14 of the methodology, analysis and interpretation of the results and discussion, and the manuscript  
15 preparation.

## 16 CONFLICT OF INTEREST

17 The authors declare no potential conflict of interests.

## 18 References

- 19 1. Furuhashi, M., M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, Rideshar-  
20 ing: The state-of-the-art and future directions. *Transportation Research Part B: Method-*  
21 *ological*, Vol. 57, 2013, pp. 28–46.
- 22 2. Zargayouna, M., A. Othman, G. Scemama, and B. Zeddini, Multiagent simulation of real-  
23 time passenger information on transit networks. *IEEE Intelligent Transportation Systems*  
24 *Magazine*, Vol. 12, No. 2, 2018, pp. 50–63.
- 25 3. Agatz, N. A., A. L. Erera, M. W. Savelsbergh, and X. Wang, Dynamic ride-sharing: A sim-  
26 ulation study in metro Atlanta. *Transportation Research Part B: Methodological*, Vol. 45,  
27 No. 9, 2011, pp. 1450–1464.
- 28 4. Agatz, N., A. Erera, M. Savelsbergh, and X. Wang, Optimization for dynamic ride-sharing:  
29 A review. *European Journal of Operational Research*, Vol. 223, No. 2, 2012, pp. 295–303.
- 30 5. Alisoltani, N., M. Zargayouna, and L. Leclercq, A Sequential Clustering Method for the  
31 Taxi-Dispatching Problem Considering Traffic Dynamics. *IEEE Intelligent Transportation*  
32 *Systems Magazine*, Vol. 12, No. 4, 2020, pp. 169–181.
- 33 6. Alisoltani, N., L. Leclercq, M. Zargayouna, and J. Krug, Optimal fleet management for  
34 real-time ride-sharing service considering network congestion. In *TRB 2019, Transporta-*  
35 *tion Research Board 98th Annual Meeting*, 2019, p. 22p.
- 36 7. Liu, M., Z. Luo, and A. Lim, A branch-and-cut algorithm for a realistic dial-a-ride prob-  
37 lem. *Transportation Research Part B: Methodological*, Vol. 81, 2015, pp. 267–288.
- 38 8. Mourad, A., J. Puchinger, and C. Chu, A survey of models and algorithms for optimizing  
39 shared mobility. *Transportation Research Part B: Methodological*, 2019.
- 40 9. Wang, S., L. Li, W. Ma, and X. Chen, Trajectory analysis for on-demand services: A  
41 survey focusing on spatial-temporal demand and supply patterns. *Transportation Research*  
42 *Part C: Emerging Technologies*, Vol. 108, 2019, pp. 74–99.

- 1 10. Davis, N., G. Raina, and K. Jagannathan, Taxi demand forecasting: A HEDGE-based tes-  
2 sellation strategy for improved accuracy. *IEEE Transactions on Intelligent Transportation*  
3 *Systems*, Vol. 19, No. 11, 2018, pp. 3686–3697.
- 4 11. Qi, H. and P. Liu, Mining Taxi Pick-Up Hotspots Based on Spatial Clus-  
5 tering. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Ad-*  
6 *vanced & Trusted Computing, Scalable Computing & Communications, Cloud*  
7 *& Big Data Computing, Internet of People and Smart City Innovation (Smart-*  
8 *World/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, IEEE, 2018, pp. 1711–1717.
- 9 12. Pelzer, D., J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt, A partition-based match  
10 making algorithm for dynamic ridesharing. *IEEE Transactions on Intelligent Transporta-*  
11 *tion Systems*, Vol. 16, No. 5, 2015, pp. 2587–2598.
- 12 13. Zuo, H., B. Cao, Y. Zhao, B. Shen, W. Zheng, and Y. Huang, High-capacity ride-sharing  
13 via shortest path clustering on large road networks. *The Journal of Supercomputing*, 2020,  
14 pp. 1–26.
- 15 14. Qiang, X. and Y. Shuang-Shuang, Clustering algorithm for urban taxi carpooling vehicle  
16 based on data field energy. *Journal of Advanced Transportation*, Vol. 2018, 2018.
- 17 15. Chen, S., H. Wang, and Q. Meng, Solving the first-mile ridesharing problem using au-  
18 tonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 35, No. 1,  
19 2020, pp. 45–60.
- 20 16. Bard, J. F. and A. I. Jarrah, Large-scale constrained clustering for rationalizing pickup  
21 and delivery operations. *Transportation Research Part B: Methodological*, Vol. 43, No. 5,  
22 2009, pp. 542–561.
- 23 17. Özdamar, L. and O. Demir, A hierarchical clustering and routing procedure for large scale  
24 disaster relief logistics planning. *Transportation Research Part E: Logistics and Trans-*  
25 *portation Review*, Vol. 48, No. 3, 2012, pp. 591–602.
- 26 18. Sáez, D., C. E. Cortés, and A. Núñez, Hybrid adaptive predictive control for the multi-  
27 vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy  
28 clustering. *Computers & Operations Research*, Vol. 35, No. 11, 2008, pp. 3412–3438.
- 29 19. Li, Y. and S. H. Chung, Ride-sharing under travel time uncertainty: Robust optimiza-  
30 tion and clustering approaches. *Computers & Industrial Engineering*, Vol. 149, 2020, p.  
31 106601.
- 32 20. Santi, P., G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, Quantifying  
33 the benefits of vehicle pooling with shareability networks. *Proceedings of the National*  
34 *Academy of Sciences*, Vol. 111, No. 37, 2014, pp. 13290–13294.
- 35 21. Vazifeh, M. M., P. Santi, G. Resta, S. Strogatz, and C. Ratti, Addressing the minimum fleet  
36 problem in on-demand urban mobility. *Nature*, Vol. 557, No. 7706, 2018, p. 534.
- 37 22. Alonso-Mora, J., S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, On-demand high-  
38 capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National*  
39 *Academy of Sciences*, Vol. 114, No. 3, 2017, pp. 462–467.
- 40 23. Yu, X., H. Miao, A. Bayram, M. Yu, and X. Chen, Optimal routing of multimodal mobility  
41 systems with ride-sharing. *International Transactions in Operational Research*, Vol. 28,  
42 No. 3, 2021, pp. 1164–1189.
- 43 24. Celebi, M. E., *Partitional clustering algorithms*. Springer, 2014.
- 44 25. Vora, P., B. Oza, et al., A survey on k-mean clustering and particle swarm optimization.  
45 *International Journal of Science and Modern Engineering*, Vol. 1, No. 3, 2013, pp. 24–26.

- 1 26. Paea, S. and R. Baird, Information Architecture (IA): Using Multidimensional Scaling  
2 (MDS) and K-Means Clustering Algorithm for Analysis of Card Sorting Data. *Journal of*  
3 *Usability Studies*, Vol. 13, No. 3, 2018.
- 4 27. Wang, Q. and K. L. Boyer, Feature learning by multidimensional scaling and its applica-  
5 tions in object recognition. In *2013 XXVI Conference on Graphics, Patterns and Images*,  
6 IEEE, 2013, pp. 8–15.
- 7 28. Ganganath, N., C.-T. Cheng, and K. T. Chi, Data clustering with cluster size constraints  
8 using a modified k-means algorithm. In *2014 International Conference on Cyber-Enabled*  
9 *Distributed Computing and Knowledge Discovery*, IEEE, 2014, pp. 158–161.
- 10 29. Murtagh, F., A survey of recent advances in hierarchical clustering algorithms. *The com-*  
11 *puter journal*, Vol. 26, No. 4, 1983, pp. 354–359.
- 12 30. Reddy, C. K. and B. Vinzamuri, A Survey of Partitional and Hierarchical Clustering Algo-  
13 rithms. *Data clustering: Algorithms and applications*, Vol. 87, 2013.
- 14 31. Ota, M., H. Vo, C. Silva, and J. Freire, STaRS: Simulating taxi ride sharing at scale. *IEEE*  
15 *Transactions on Big Data*, Vol. 3, No. 3, 2017, pp. 349–361.
- 16 32. Qian, X., W. Zhang, S. V. Ukkusuri, and C. Yang, Optimal assignment and incentive design  
17 in the taxi group ride problem. *Transportation Research Part B: Methodological*, Vol. 103,  
18 2017, pp. 208–226.
- 19 33. Hyland, M. F. and H. S. Mahmassani, *Sharing Is Caring: Dynamic Autonomous Vehicle*  
20 *Fleet Operations Under Demand Surges*, 2018.
- 21 34. Ameli, M., M. S. S. Faradonbeh, J.-P. Lebacque, H. Abouee-Mehrizi, and L. Leclercq, De-  
22 parture Time Choice Models in Congested Transportation Systems Based on Mean Field  
23 Games. *arXiv preprint arXiv:2101.02238*, 2021.
- 24 35. Lamotte, R. and N. Geroliminis, *The morning commute in urban areas: Insights from*  
25 *theory and simulation*, 2016.
- 26 36. Ameli, M., Heuristic Methods for Calculating Dynamic Traffic Assignment. IFSTTAR and  
27 Université de Lyon, 2019.
- 28 37. Leclercq, L., N. Chiabaut, and B. Trinquier, Macroscopic fundamental diagrams: A  
29 cross-comparison of estimation methods. *Transportation Research Part B: Methodolog-*  
30 *ical*, Vol. 62, 2014, pp. 1–12.
- 31 38. Mariotte, G. and L. Leclercq, Flow exchanges in multi-reservoir systems with spillbacks.  
32 *Transportation Research Part B: Methodological*, Vol. 122, 2019, pp. 327–349.
- 33 39. Ameli, M., N. Alisoltani, and L. Leclercq, *Lyon North realistic trip data set during the*  
34 *morning peak*, 2021.
- 35 40. Krzanowski, W. J. and Y. Lai, A criterion for determining the number of groups in a data  
36 set using sum-of-squares clustering. *Biometrics*, 1988, pp. 23–34.
- 37 41. Hippocampus, Clustering into same size clusters, 2018.
- 38 42. Alisoltani, N., L. Leclercq, and M. Zargayouna, Can dynamic ride-sharing reduce traffic  
39 congestion? *Transportation research part B: methodological*, Vol. 145, 2021, pp. 212–  
40 246.
- 41 43. Banguion, P., G. Sirin, and A. Malard, GeoTwin: a platform to experiment new mobility  
42 services. *DSC 2021 EuropeVR*, 2021.