A Fast Response Multi-Objective Matching Algorithm for Ridesharing

Jiaze Sun[1,2], Nan Han[1], Jianbin Huang[3], Jiahui Deng[4], Yang Geng[1]

[1] School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, China
[2] Shaanxi Key Laboratory of Network Data Intelligent Processing, Xi'an University of Posts and Telecommunications, China
[3] School of Computer Science and Technology, Xidian University, China
[4] School of Computer Science and Technology, Northwest University, China
sunjiaze@xupt.edu.cn, 1826995691@stu.xupt.edu.cn, jbhuang@xidian.edu.cn, dengjiahui@stumail.nwu.edu.cn, gywanliu@163.com

Abstract

In many metropolitans, especially during rush hours on holidays, thousands of riders will initiate travel orders at the same time, and the existing carpool matching model cannot handle largescale travel orders quickly enough. For handling this problem, a fast and efficient multi-objective carpool matching algorithm (MOCMA) is put forward, which generates a set of different matching schemes suitable for different practical scenarios. First, the idea of partition is adopted to gather riders and drivers with similar journeys, and the relationship matrix construction algorithm (RMCA) is proposed; then from the perspective of riders and drivers, the maximum service quality and the maximum shared mileage are two objectives, and a set of non-dominated solution sets are generated using MOCMA; finally, the simulation experiment results show that MOCMA proposed is suitable for different practical scenarios, the matching success rate is as high as 99.7%, and it has significant advantages over MOEA/D, SPEA2, and FastPGA.

Keywords: Ridesharing, Ride-Matching, Multi-objective optimization, Genetic algorithm

1 Introduction

In recent years, the problem of urban traffic congestion has become more and more serious. Traffic congestion will cause many problems such as energy waste, environmental pollution, and difficulty in taking taxis. The large demand for automobile transportation at rush hours together with low occupancies leads to traffic congestion in many urban [1]. Particularly when the taxi is empty load, it will cause unnecessary waste of resources and increase the burden on the environment. The number of vehicles that can travel every day is limited to avoid traffic congestion. Research shows that private car occupancy rates are relatively low. The average car occupancies used for leisure travel in Europe range from 1.8 for leisure trips to 1.1 for commuters [2], the same is true in the United States [3]. Ridesharing can share taxi services and occupy a large number of taxi seats, so that the remaining resources can be used rationally, which will help reduce the pressure on the economy, the environment, and traffic. Nowadays, the rapid development of ridesharing has gradually changed the way people travel. Especially during rush hours on holidays, many riders choose to accept ridesharing to avoid waiting too long. Out of environmental awareness or economic benefits, more and more private car drivers are willing to share their journey with riders. The requirements for eCommerce, such as safety and timeliness, are becoming increasingly strict [4]. To increase the occupancy rate of service taxis, and to make full use of a large number of human and environmental resources, scholars in various fields continue to explore and research. An efficient and reliable data routing decision scheme is critical for passengers [5].

Ridesharing can be divided into static ridesharing and dynamic ridesharing. Static ridesharing refers to the comprehensive consideration of the driver and riders' boarding location, alighting location, departure time, and maximum waiting time before the vehicle departs, the matching relationship is determined in advance and the driving path is planned. Dynamic ridesharing refers to the real-time matching of drivers and riders while the vehicle is driving. Ridesharing is beneficial in many aspects, such as increasing the income of drivers, reducing the expense of riders, alleviating traffic pressure, and reducing environmental pollution. Driven by huge economic and environmental benefits, the issue of ridesharing has gradually gained wide attention from academia [6-8] and industry [9-10]

*Corresponding Author: Nan Han; E-mail: 1826995691@stu.xupt.edu.cn
DOI: 10.53106/160792642021092205014

in recent years.

Carpool matching problem is usually modeled as an optimization problem, and its processing methods are divided into two categories: mathematical methods and heuristic algorithms. The main idea of the mathematical method is to solve it with integer programming or dynamic programming. Although simple and easy to implement, it has the disadvantage of high complexity. The complexity of the heuristic is low but unstable, and the performance depends on the specific problem. The running time and efficiency will be affected when the scale of the problem is too large. How to quantify the matching relationship between riders and drivers, and how to propose fast and efficient algorithms to match a large number of travel orders promptly have become the current research focus.

This paper proposes a fast response multi-objective matching algorithm for ridesharing, which is suitable for large-scale carpool matching problems, and the running time and efficiency will not be affected by the expansion of the problem scale.

In summary, the main contributions of our work are: (a) proposing a multi-objective carpool matching algorithm (MOCMA), (b) proposing the approximate shortest distance (ASD), (c) introducing the relationship matrix construction algorithm (RMCA), and (d) proposing the matrix compression.

2 Related Work

In recent research on carpool matching problems, literature [11] used mathematical programming to establish a stable matching model of the dynamic matching system. Literature [12] used mixed integer programming to solve different types of ridesharing problems. Literature [13] introduced the problem of large-scale real-time ridesharing with a service guarantee on road networks, which match the schedule request to the vehicle dynamically while trip waiting and service time constraints are satisfied and propose kinetic tree algorithms which are better suited to efficient scheduling of dynamic requests and adjust routes on-the-fly. Literature [14] used two methods to solve the carpool matching problem, one is to model the problem as a maximum-weighted bipartite graph matching problem for the situation of multiple riders and multiple drivers, an approximate method with error bound guarantee is proposed to compute the driver-rider pairs to avoid the expensive calculation of the shared route percentage, the other is to consider the situation of multiple drivers and a single rider, and develop a best-first method to compute the top-k drivers for a rider. Some literature adds user preferences into consideration [15-16], e.g., female riders are more willing to match female drivers due to gender considerations [17-18]. Optimal task allocation strategy is widely used to deal with combinatorial optimization problems [19].

Heuristic algorithms have also been widely used in the research of ridesharing problems. Literature [20] proposed a hybrid algorithm based on GIS and ant colony. First, the spatio-temporal clustering of passengers is carried out; second spatio-temporal matching of passengers' clusters and drivers are carried out by combining Voronoi continuous range query, a region connected calculus, and Allen's temporal interval algebra; third, the optimum shared-route is found by the ant colony optimization algorithm. Literature [21] proposed a heuristic algorithm to divide the graph based on the bipartite graph generated by the dynamic carpool matching problem.

Many papers also consider multi-objective algorithms to deal with ridesharing matching problems. Literature [22] proposed an exact and a heuristic method for ridesharing problem, which aims to encourage employees to pick up colleagues to and from get off work, reduce the number of private cars to and from the company site, minimize vehicle mileage and maximize the number of participants. Literature [23] applied the hybrid simulated annealing algorithm to allocate rider demand by proposing shared-taxi algorithms, to minimize the total travel time of riders and maximize the system benefits. Literature [24] proposed an incentive-compatible dynamic ridesharing solution based on auction, which adapts to the individual preferences of the participants, minimizes the total travel distance and maximizes the number of participants.

In the papers mentioned above, there are characteristics of high computational complexity and slow operation speed when using exact mathematical methods to deal with large-scale carpool matching problems. Most of the literature that used heuristic algorithms to deal with it converts multi-objective into single-objective considerations through weighting or divide multi-objectives into primary and secondary objectives, and does not fully consider the dominance relationship between multiple objectives. The Kuhn-Munkras (KM) algorithm that is the mathematical method for solving the maximum weight matching problem takes $O(n^3)$ time, which is too expensive when the bipartite graph is quite large. The proposed MOCMA not only takes into account the interests of drivers and passengers simultaneously, but also has a lower time complexity, and the algorithm performance does not decrease with the increase of the problem scale.

3 Problem Formulation

DEFINITION 1 (ROAD NETWORK) A road network is denoted by a weighted undirected graph $G=(V,E)$, where $V=(node\_id, latitude, longitude)$ is a set of nodes. $E=(u,v,W(u,v))$, $(u,v \in V)$ is a set of edges, W(u,v) is the Manhattan Distance between node u and node v.

DEFINITION 2 (RIDER) $rider_i = (r_i^s, r_i^d, r_i^{d\_time}, r_i^{MW\_time})$, where r_i^s is the starting point of the ith rider, r_i^d is the destination, $r_i^{d\_time}$ is the departure time, $d_i^{MV\_time}$ is the maximum waiting time.

DEFINITION 3 (DRIVER) $diver_i = (d_i^s, d_j^d, d_i^{d\_time})$, where d_i^s is the starting point of the ith driver, d_i^d is the destination, $d_i^{d\_time}$ is the departure time.

DEFINITION 4 (BIGRAPH) Bigraph is a special model in graph theory. Let $g = (V, E)$ be an undirected graph. If vertex V can be divided into two disjoint subsets (A, B), and the two vertices i and j associated with each edge (i, j) in the graph belong to these two different vertex sets $(i \in A, j \in B)$, then the graph g is called a bigraph.

We construct a bigraph $g = (g_R, g_D, g_E)$, where the vertex set g_R represents the set R composed of riders, the same for g_D, and g_E represents the set of associated edges between g_R and g_D. The vertex sets g_R and g_D do not intersect each other, and the two vertices associated with each edge in the graph belong to different vertex sets.

DEFINITION 5 (BIGRAPH MATCHING) Given a bigraph g, in a subgraph M of g, any two edges in the edge set of M are not attached to the same vertex, then M is a bigraph match.

DEFINITION 6 (PERFECT MATCHING) If each edge of the bigraph has a weight, a complete matching scheme M^* is required to maximize the weight of all matching edges, then M^* is called the perfect matching.

Example 1. As shown in Figure 1, the set of red edges is a matching M(0.3 + 0.2 + 0.1 = 0.6), and the matching M* (0.2 + 0.3 + 0.4 = 0.9) shown in Figure 2 is a perfect matching.

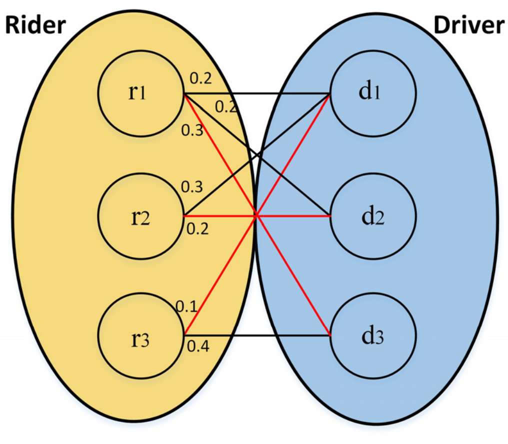

Figure 1. Unperfect matching

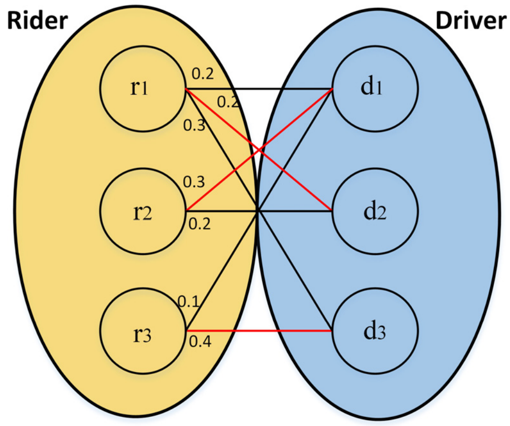

Figure 2. Perfect matching

DEFINITION 7 (α) It is used to describe the compactness of the relationship between the rider and the driver set, i.e., the number of drivers that can be matched of the rider divided by the driver set size, or the compactness of the relationship between the driver and the rider set, i.e., the number of riders that can be matched of the driver divided by the rider set size.

Through time constraints and road network constraints, the candidate set is constructed for drivers and riders, which greatly reduces the search space.

The Time Constraint: The delta-T between the departure time of the driver and that of the rider is less than or equal to the rider's maximum waiting time, which ensures that the driver has enough time to serve the rider within the acceptable range.

$$r^{d\_time} - d^{d\_time} \le r^{MV\_time} \quad \textbf{(1)}$$

The Road network Constraint: The distance between the starting point of the driver and the rider divided by the average road speed should be less than or equal to the maximum waiting time of the rider, so that drivers who are far away from the rider are excluded. $AD(d^s, r^s)$ represents the actual distance traveled from d^s to r^s.

$$\frac{AD(d^s, r^s)}{Speed} \le r^{MW\_time} \quad \textbf{(2)}$$

Quality of Service (QoS): The driver travels from his starting point to the riders', takes the shortest route to send the rider to the destination, and then goes to his destination. For riders, there is no extra detour distance, riders care about the time they need to wait, so the authors use the rider waiting time to measure the quality of service provided by the driver to the rider.

$$QoS_{ij} = 1 - \frac{r_i^{AV\_time_j}}{r_i^{MV\_time}} \quad \textbf{(3)}$$

Equation(3) is the quality of the ith rider served by the jth driver. The actual waiting time of the rider is less than the maximum waiting time, i.e.,

$r_i^{AW\_time_j} \le r_i^{MV\_time}$, so $0 \le QoS_{ij} \le 1$, the larger the QoS, the better the service quality.

Shared mileage Rate (SmR): For drivers, the more miles shared with riders, the more generous the rewards. Therefore, the authors use the shared mileage rate to describe the interests of drivers.

$$SmR_{ij} = \frac{AD(r_j^s, r_i^d)}{AD(d_j^s, r_i^s) + AD(r_i^s, r_i^d) + AD(r_i^d, d_j^d)} \quad (4)$$

Equation(4) is the shared mileage rate of the ith rider and the jth driver, $0 \le SmR_{ij} \le 1$, the large the SmR, the more profit the driver. Introduce 0-1 variable:

$$x_{ij} = \begin{cases} 1\,(\text{the ith rider is matched to the jth driver}) \\ 0\,(\text{the ith rider is not matched to the jth driver}) \end{cases} \quad \textbf{(5)}$$

The QoS_{ij} and SmR_{ij} represents the quality and shared mileage rate of the ith rider served by the jth driver respectively, n and m is the number of riders and drivers respectively. The mathematical model of multi-objective carpool matching (MOCM) problem is:

$$\begin{cases} \max \sum_{i=1}^{n} \sum_{j=1}^{m} QoS_{ij} * x_{ij} \\ \max \sum_{i=1}^{n} \sum_{j=1}^{m} SmR_{ij} * x_{ij} \\ s.t. \sum_{i=1}^{n} x_{ij} = 0/1 (j = 1, 2, \ldots, m) \\ \quad \sum_{j=1}^{m} x_{ij} = 0/1 (i = 1, 2, \ldots, n) \\ \quad x_{ij} = 0/1((i = 1, 2, \ldots, n), (j = 1, 2, \ldots, m)) \end{cases} \quad \textbf{(6)}$$

Equation (6) represents the mathematical model of the MOCM problem, where the objective function represents the maximum service quality and the maximum shared mileage rate, constraints, in turn, means the rider can be served by one driver at most, each driver can serve at most one rider, and the decision variable can only take 0 or 1.

4 Multi-Objective Carpool Matching Algorithm

4.1 Approximate Shortest Distance

The shortest path is used in the process of selecting candidate drivers for riders and calculating SmR. However, the shortest path takes $O((E+V)\log V)$ time [25], which is not suitable for processing large-scale complex road networks. Therefore, we hope to find an approximate distance to replace the Dijkstra shortest path distance (SPD).

(1). Euclidean Distance (ED): refers to the true distance between two points in n-dimensional space.

$$ED_{xy} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad \textbf{(7)}$$

$X = (x_1, x_2, \ldots, x_n)$. x_i is the ith dimensional coordinate of the point X, same for y_i. Since the straight line between two points is the shortest, ED must be less than SPD. Because the shortest path between two nodes in the road network is related to the distribution structure of the road network, the road network is complicated. Although the ED has the advantages of low time and space complexity, it is not appropriate to directly replace SPD. Taking these factors into account, we divide the road network into regions, use the K-means clustering algorithm [26] to divide the road network into several regions according to the geographic location of the nodes, and propose a cluster distance considering the comprehensive geographic location.

(2). Cluster Distance (CD): Considering the reality comprehensively, two locations in the same region have similar road traffic, and the sum of ED from these two locations to the regional center is used to describe CD. In different regions, road traffic is very different, so CD is described by the sum of ED between two locations and regional centers and the sum of ED between two regional centers. We define CD_{ij} as Equation (8). $C(v)$ represents the center of the cluster where site v is located.

$i \in V, j \in V$

$$CD_{ij} = \begin{cases} ED(i, C(i)) + ED(j, C(j)), C(i) = C(j) \\ ED(i, C(i)) + ED(j, C(j) + ED(C(i), C(j)), C(i) \ne C(j) \end{cases} \quad \textbf{(8)}$$

(3). Approximate shortest distance (ASD): As we know, the ED must less than ASD. The CD obtained by adding multiple distances is often greater than SPD when the network partition is specific enough. Therefore, we use the average of ED and CD to replace SPD. ASD is defined Equation (9).

$$ASD = \frac{ED + CD}{2} \quad \textbf{(9)}$$

4.2 Relationship Matrix Construction Algorithm

Given the large number of travel orders initiated by riders and drivers at a certain moment, the authors aim to quickly and efficiently match riders with drivers. So the problem is transformed into: Find a perfect matching in a weighted bigraph $g = (g_R, g_D, g_E)$ quickly and efficiently.

In this paper, the adjacency matrix is used to store the bigraph, but for a large-scale matrix with thousands of riders and drivers, it is somewhat difficult to deal with it as a whole. Therefore, riders and drivers with close relationships are separately classified for processing, and the relationship matrix construction algorithm (RMCA) is proposed in this section.

As shown in Algorithm 1, input an unmatched rider and the parameter α, add the rider to the rider set R and marked it (line 1). Enter the loop when there is an unmarked driver in the candidate set of R (line 3) and select a driver from that (line 5) to determine whether it can be added to D (line 6). Mark this driver and jump out of this layer of the loop (line 7) if the conditions are met, otherwise it is marked that the driver cannot be added and continues to search for unmarked drivers in the candidate set of R (line 8). When there are unmarked riders in the candidate set of D, it enters the loop (line 9) and selects a rider (line 11) in that to determine whether it can be added to R (line 12). Mark that the rider has been visited and jump out of the loop (line 13) if the conditions are met, otherwise mark that the rider cannot be added and continue to search for unmarked riders in the candidate set of D (line 14). Remove the rider or driver that has been visited before and cannot be added, so that these riders or drivers can be selected again after a cycle of R or D is expanded (line 15). The algorithm ends until no driver(rider) can join in the candidate set of R(D).

Algorithm 1. Pseudocode for EstablishSet

```
Algorithm 1. EstablishSet
Input: rider, α
Output: R[], D[]
1.  R[]←rider, Marked(rider, 1)
2.  While true
3.     while Exist_unmarked (R.candidateSet)
4.          dflag = 0
5.          driver = GetCandidate(R)
6.          if IfcanAdd(D, driver, α)
7.              dflag = 1, D[]←driver,
8.          Marked (driver, 1), break
9.          else Marked (driver, 0)
10.    while Exist_unmarked (D. candidateSet)
11.         rflag = 0
12.         rider = GetCandidate(R)
13.         if IfcanAdd(R, rider, α)
14.             rflag = 1, R[]←rider
15.         Marked (rider, 1), break
16.        else Marked (rider, 0)
17.    unMaked (D, R, 0)
18.    IF dflag = 0 & rflag = 0
19.        Break
```

4.3 Matrix Compression

From Section 3, the solution to the n*m scale carpool matching problem is a n*m matrix, where the element of the matrix values is 0 or 1, and each row and column have at most one non-zero element. We compress and store it as a 1*m matrix to reduce the storage space. There is a one-to-one correspondence between riders and drivers, and the one-dimensional matrix values are a sequence of all integers between 0 and n-1 when $n = m$; the number of drivers is more than that of riders when $n < m$, there are unmatched drivers (represented by -1), and the one-dimensional matrix values are a sequence consisting of all integers between 0 and n-1 and the number -1 of m; the number of riders is more than that of drivers when n>m, and there are unmatched riders, and the one-dimensional matrix values are a sequence of integers between 0 and n-1. As shown in Figure 3 to Figure 5, $x[j] = i$ indicates that the element in the ith row of the jth column is non-zero, and the meaning in the corresponding carpool matching problem is that the jth driver will match the $x[j]$ th rider. Therefore, the authors only need matrix x to represent the solution of carpool matching.

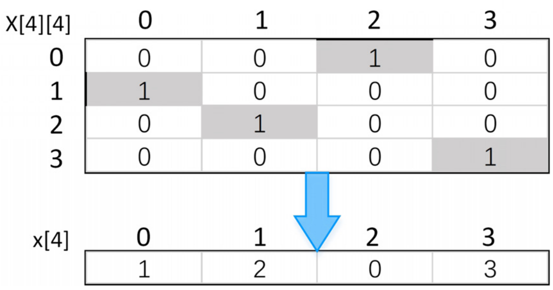

Figure 3. Matrix compression1

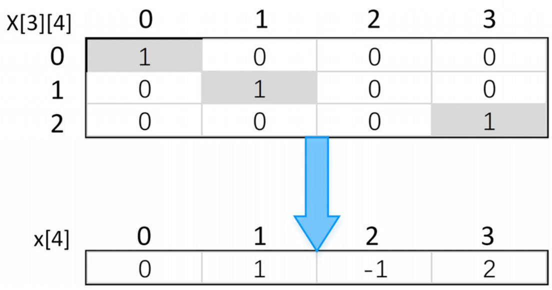

Figure 4. Matrix compression2

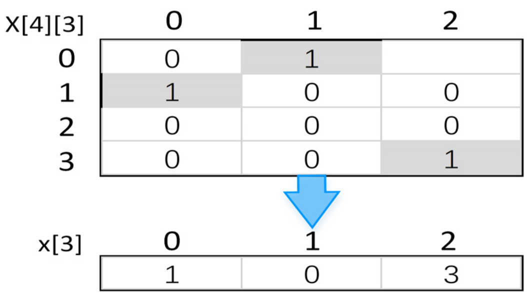

Figure 5. Matrix compression3

4.4 The Flow of Multi-Objective Carpool Matching Algorithm

According to the characteristics of the MOCM problem, the coding in the experiment uses permutation number coding, the multi-objective genetic algorithm uses the NSGA-II [27] algorithm framework. Figure 6 is the Flowchart of MOCMA. Algorithm 2 is the Pseudocode of MOCMA.

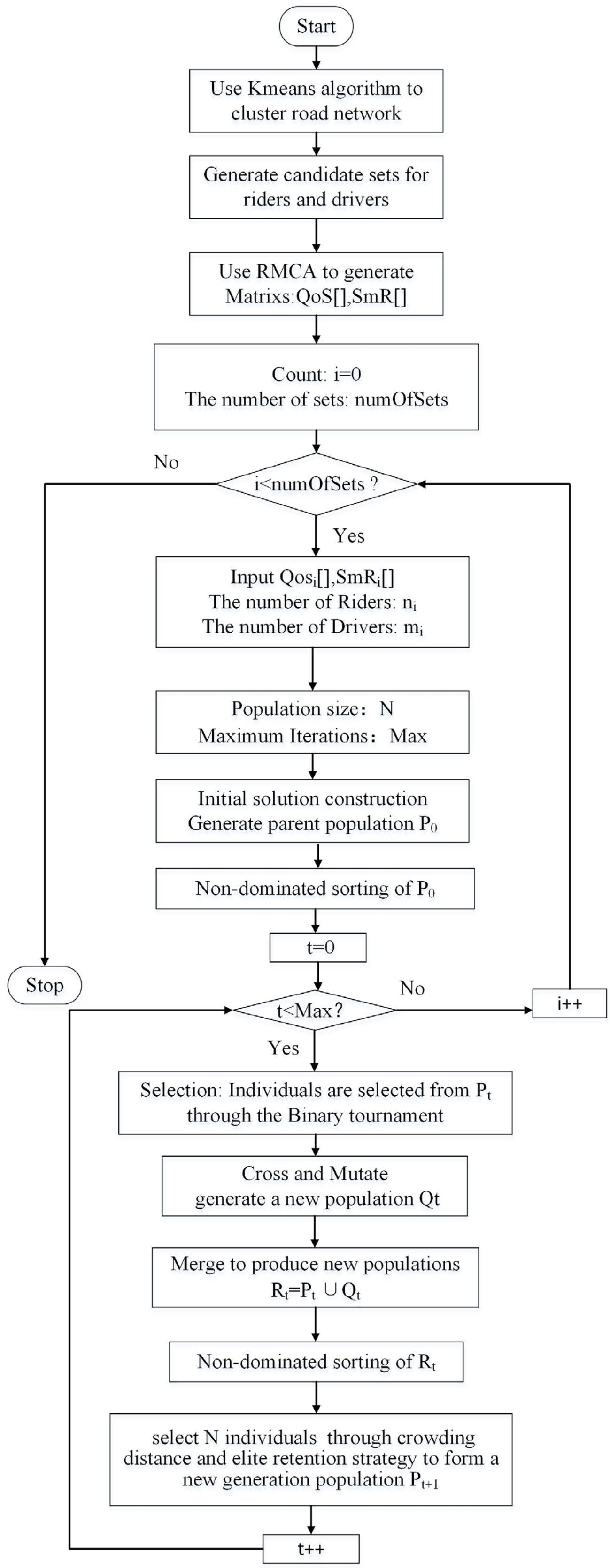

Figure 6. Flowchart of MOCMA

Algorithm 2. Pseudocode for MOCMA

Algorithm 2. MOCMA
Input: G, Drivers[], Riders[]
Output: M_Pairs[]
1. Kmeans(G)
2. numofsets = RMCA(Drivers, Riders, QoS, SmR)
3. $i = 0$
4. for i < numofsets do
5. $P_0 = \text{Pop\_Intialization}(QoS_i, SmR_i)$
6. NondominatedSort(P_0)
7. t = 0
8. for t < Max do
9. Selection(P_t)
10. Crossover(P_t), Q_t = Mutation(P_t)
11. R_t = Merge(P_t, Q_t)
12. NondominatedSort(R_t)
13. P_{t+1} = Select(R_t)
14. t++
15. M_Pairs[] = P_{t+1}
16. i++
17. Print(M_Pairs)

As shown in Algorithm 2, input road network, the order of drivers, and riders. Use the k-means clustering algorithm to cluster the road network (line 1). Call RMCA to divide the global riders and drivers into a series of closely independent small sets, then generate the corresponding QoS matrix and SmR (line 2). Enter the loop when there exists a set unprocessed. (line 4). Perform initial encoding according to the size relationship the number of riders n_i and the number of drivers m_i, generate the parent population P_0 (line 5), and perform non-dominated sorting on P_0 (line 6). Enter the loop when iteration is not terminated (line 8). Select individuals from P_t through the Binary tournament (line 9). Two points crossover and Swap mutation are used to cross and mutate selected individuals to generate a new generation of population Q_t (line 10). By merging P_t and Q_t to generate a new population $R_t = P_t \cup Q_t$ (line 11). Non-dominated sorting of R_t (line 12), and N individuals are selected through crowding distance and elite retention strategy to form a new generation population P_{t+1} (line 13). Print matching pairs M_Pairs (line 17).

Figure 7 shows an example of using MOCMA to obtain a set of non-dominated solution sets. Reference macroeconomics resource allocation model, the drivers' interests are first considered when the overall number of drivers is less than riders, the solution with the highest total SmR is selected as the final matching plan; the riders' interests are first considered when the overall number of riders is less than drivers, the solution with the highest total QoS is selected; when the total number of riders is equal to drivers, the best compromise solution (solution in the middle of the ordered non-dominated solution set) is taken.

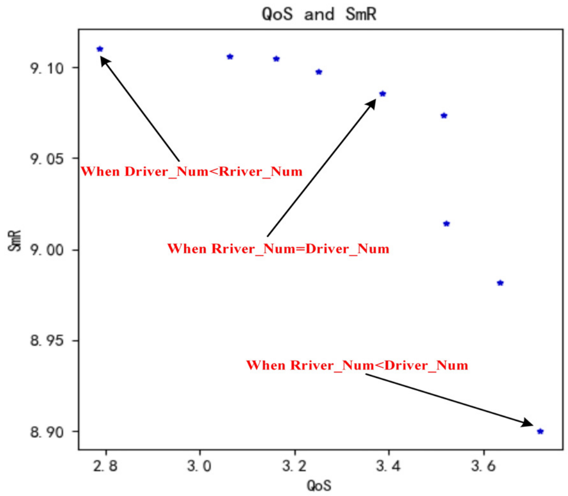

Figure 7. Selection of solution in different situations

The time complexity of ED is O(1), and that of CD is O(n), where n is the number of road network nodes, so the ASDA time complexity is O(n). The time complexity of RMCA is O(N), where $N = numofRider + numofDrier$, numofRider, and numofDriver is the number of rider and driver sets respectively. The time complexity of NSGA-II is O(M*N^2), M is the number of objective functions, and N is the population size, so the algorithm complexity of MOCMA is O(S*M*N^2), and S is the number of matrix decomposition.

5 Simulation Experiments

The experiment is divided into three sections, the first section introduces the dataset, the second section is to discusses the two important parameters of this article: the number of clusters *Core* and the parameter α, and the third section is algorithm contrast. In the following experiments, the initial population size is 80, the number of iterations is 500, and the crossover and mutation probabilities are both 0.9. The experiment was repeated 50 times independently and the average value was taken as the final result.

5.1 Dataset

In this paper, the New York city road network dataset is obtained through Open Street Map [28], which contains 61,298 nodes and 141,373 edges. The order dataset uses the New York Taxi Data Set [29]. The information available for each trip includes the longitude and latitude of the trip origin and destination and the departure time of the trip. We choose the green taxi records of February 2016 as the order data set.

5.2 Parameters Discuss

To reduce the time complexity, ASD is proposed to replace SPD. ASD is related to the number of clustering clusters *Core*. To determine the best *Core* and minimize the difference between ASD and SPD, we use the average *Closeness Closeness* of SPD and ASD of 10,000 randomly selected test orders as the final *Closeness* the equation of *Closeness* is shown in (10):

$$Closeness = 1 - \frac{|ASD - SPD|}{SPD} \quad \textbf{(10)}$$

In Figure 8, Core selects 10 to 100. The overall proximity is in an upward trend as the number of clusters increases. The compactness is minimal when $Core = 10$ and $Closeness = 0.802$. The degree of compactness is maximum when $Core = 90$ and $Closeness = 0.842$. It depends on the structure of the road network, the size of the area, and the distribution of the location. The partition effect of the road network is closest to the real situation when $Core = 90$. Therefore, the *Core* of the following experiments takes 90.

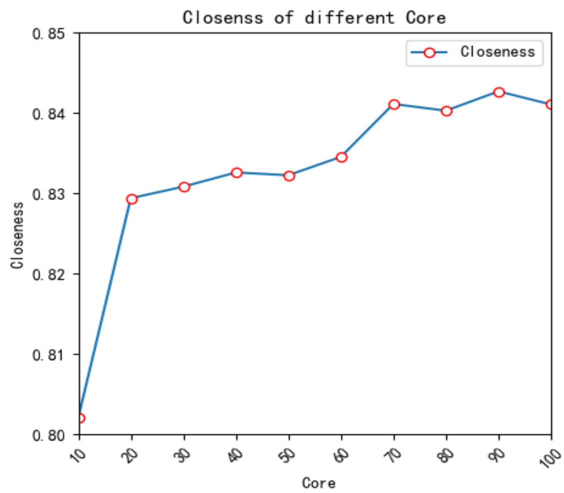

Figure 8. Closeness of different Core

The size of α directly affects the results of RMCA and then affects the quality and efficiency of MOCMA. To discuss the impact of the size of α on the experimental results, we use a randomly selected 10,000 test order set, in which the number of driver orders and rider orders are both 5000, and the running time in milliseconds and MFR are used as the evaluation criteria of the experimental results. The experiment results are shown in Table 1.

Table 1. Comparison of the number of different α matrices, solving time, and matching failure rate

1/α	Number Of matrices	Build Matrix Time	Solving Matrix Time	Total Time	MFR
2	1407	24776	8809	33585	0.20%
3	1048	33764	5918	39682	0.22%
4	849	42871	5046	47917	0.18%
5	754	49811	4806	54617	0.16%
6	699	60309	4108	64417	0.16%
7	644	65559	4270	69829	0.14%

In Table 1, the larger α means that the relationship between riders and drivers in the same matrix is closer, the more matrices are generated, the longer the solving time, and the smaller the matrix construction time. When α tends to be infinitely small, the number of matrices is infinitely close to 1, which means that without any processing, the running time increases infinitely. With the increase of α, MFR is not much different, but the running time is greatly reduced. Therefore, it is proved that RMCA can improve the matching speed without affecting the matching quality. The following experiment α takes 1/2 for convenience.

5.3 Algorithm Contrast

To verify the superiority of MOCMA, we designed two groups of comparative experiments. One is compared with the mathematical KM algorithm and the other with the three classical multi-objective algorithms.

MOCMA has a high quality to deal with the carpool matching problem as a very small MFR shown in Table 1. The authors use the KM algorithm as the reference object and the running time as the evaluation criteria to evaluate the efficiency of MOCMA. Since the KM algorithm cannot solve the multi-objective problem, we use a weighted method to assign weights to the two objectives of QoS and SmR, which are 0.5 and 0.5 respectively. Convert MOCMA to GA and compare the running time with KM. The experiment results are shown in Figure 9.

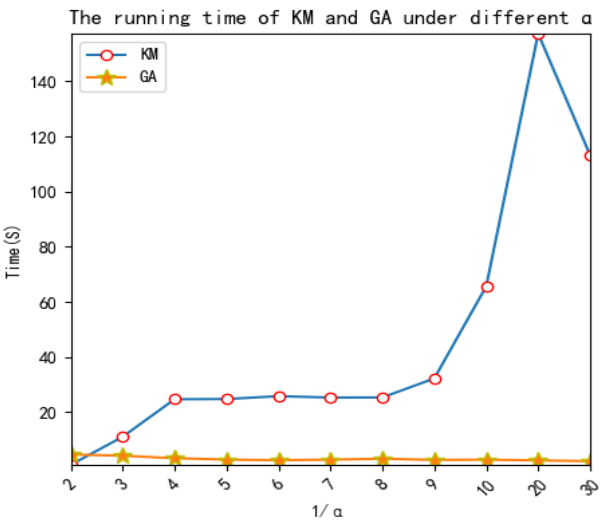

Figure 9. Running time of KM and GA

In Figure 9, the number of matrices increases as α becomes smaller. The running time of the KM algorithm is much longer than GA. The running time of the KM algorithm no longer increases when α takes 1/20 and 1/30, this's because when α is reduced to a certain extent, the number of orders limits the number of matrices and indirectly limits the size of matrices. The running time of GA is relatively small and stable in the range of 2200ms-4700ms.

To show the advantages of MOCMA, we compared it with MOEA/D, SPEA2, and FastPGA based on running time and MFR as evaluation criteria. In Table 2, the unit of time in milliseconds, and the size of the test data set are 100, 200, 400, 1000, and 2000. The size of the passenger set and the driver set are the same for convenience.

Table 2. The comparison of different algorithm

Size	MOEA/D		SPEA2		FastPGA		MOCMA	
	time	MFR	time	MFR	time	MFR	time	MFR
100	359	0.960	469	0.940	1828	0.920	**89**	**0.83**
200	652	0.970	975	0.970	1718	0.960	**259**	**0.83**
400	825	0.985	1791	0.955	1849	0.965	**417**	**0.64**
1000	2752	0.986	3897	0.978	3849	0.972	**910**	**0.44**
2000	8120	0.987	10535	0.984	10824	0.980	**1512**	**0.42**

As shown in Table 2, the number of matching orders increases as the order set size increases, so the MFR of MOCMA decreases. The experimental results show that compared to other algorithms, MOCMA has a significant decrease in running time and MFR, which is more suitable for handling large-scale and timely response order matching problems.

To verify the statistical results of the algorithms, the Wilcoxon signed-rank test is used for the performance difference of pair-wise comparison algorithms. The following hypothesis is proposed: H_0: MOCMA has a significant improvement over the other algorithms.

Table 3 shows the statistical results, p-values is considered to reject H_0 or not. p-value=0.043 < 0.05. means the result of the Wilcoxon test is less than the significance level α. we can accept that MOCMA has a significant improvement over the other three algorithms.

Table 3. Wilcoxon signed-rank test results with significance level α = 0.05

Methods	MOCMA vs MOEA/D	MOCMA vs SPEA2	MOCMA vs FastPGA
p-value	0.043	0.043	0.043

6 Conclusion

This paper uses the k-means clustering algorithm to cluster the road network. It narrows the geographic region by judging whether they are in the same cluster and improves the search speed when constructing the candidate set of riders or drivers. Using the boundary approximation idea, we propose the ASD to replace the complicated and time-consuming SPD. To reduce the scale of the problem and improve the matching efficiency, the riders and drivers with a high degree of compactness are gathered to form multiple smaller sets. Comparison experiments with multiple algorithms show that MOCMA has good performance. Besides, MOCMA not only solves the carpool matching problem, but its variants can be used in the fields of operation research and economics to solve assignment, optimize resource configuration, and task allocation problems. However, the uniform driving speed is used to calculate the travel time in MOCMA. In future research work, dynamic real-time driving speed can be considered. It is also possible to consider the many-to-many matching mode.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grant No. 61876138), the Key R & D Project of Shaanxi Province (2020GY-010) the Industrial Research Project of Xi'an (2019218114GXRC017CG018-GXYD17.10), and the Special Fund for Key Discipline Construction of General Institutions of Higher Learning from Shaanxi Province.

References

[1] N. Agatz, A. Erera, M. Savelsbergh, X. Wang, Optimization for dynamic ride-sharing: A review, *European Journal of Operational Research*, Vol. 223, No. 2, pp. 295-303, December, 2012.

[2] European Environmental Agency, *occupancy rates of passenger vehicles*, TERM 029, July, 2010.

[3] A. Santos, N. Mcguckin, H. Y. Nakamoto, D. Gray, S. Liss, *Summary of travel trends: 2009 national household travel survey*, FHWA-PL-ll-022, June, 2011.

[4] H. H. Gao, W. Q. Huang, Y. C. Duan, The Cloud-edge-based Dynamic Reconfiguration to Service Workflow for Mobile Ecommerce Environments: A QoS Prediction Perspective, *ACM Transactions on Internet Technology*, Vol. 21, No. 1, pp. 1-23, February, 2021.

[5] H. H. Gao, C. Liu, Y. H. Z. Li, X. X. Yang, V2VR: Reliable Hybrid-Network-Oriented V2V Data Transmission and Routing Considering RSUs and Connectivity Probability, *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-14, April, 2020.

[6] M. H. Hasan, P. V. Hentenryck, A. Legrain, The commute trip-sharing problem, *Transportation Science*, Vol. 54, No. 6, pp. 1640-1675, November-December, 2020.

[7] X. Liu, H. Titheridge, X. Yan, R. Wang, W. Tan, D. Chen, J. Zhang, A passenger-to-driver matching model for commuter carpooling: Case study and sensitivity analysis, *Transportation Research Part C: Emerging Technologies*, Vol. 117, Article No. 102702, August, 2020.

[8] A. EL Azzaoui, J. H. Park, Post-Quantum Blockchain for a Scalable Smart City, *Journal of Internet Technology*, Vol. 21, No. 4, pp. 1171-1178, July, 2020.

[9] P. M. D'Orey, R. Fernandes, M. Ferreira, Empirical evaluation of a dynamic and distributed taxi-sharing system, *International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, USA, 2012, pp. 140-146.

[10] Y. Y. Yin, Q. Huang, H. H. Gao, Y. S. Xu, Personalized APIs Recommendation with Cognitive Knowledge Mining for Industrial Systems, *IEEE Transactions on Industrial Informatics*, pp. 1-1, November, 2020.

[11] X. Wang, N. Agatz, A. Erera, Stable Matching for Dynamic Ride-sharing Systems, *Transportation Science*, Vol. 52, No. 4, pp. 850-867, July-August, 2018.

[12] V. Armant, N. Mahbub, K. N. Brown, Maximising the Number of Participants in a Ride-Sharing Scheme: MIP Versus CP Formulations, *2015 IEEE 27th International Conference on Tools with Artificial Intelligence*, Vietri sul Mare, Italy, 2015, pp. 836-843.

[13] Y. Huang, F. Bastani, R. Jin, X. S. Wang, Large scale real-time ridesharing with service guarantee on road networks, *Proceedings of the VLDB Endowment*, Vol. 7, No. 14, pp. 2017-2028, October, 2014.

[14] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, Z. Gong, An efficient ride-sharing framework for maximizing shared route, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 30, No. 2, pp. 219-233, February, 2018.

[15] H. H. Gao, L. Kuang, Y. Y. Yin, B. Guo, K. Dou, Mining Consuming Behaviors with Temporal Evolution for Personalized Recommendation in Mobile Marketing Apps, *Mobile Networks and Applications*, Vol. 25, No. 4, pp. 1233-1248, August, 2020.

[16] X. X. Yang, S. J. Zhou, M. Cao, An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews, *Mobile Networks and Applications*, Vol. 25, No. 2, pp. 376-390, April, 2020.

[17] Z. Peng, W. Shan, P. Jia, B. Yu, Y. Jiang, B. Yao, Stable ride-sharing matching for the commuters with payment design, *Transportation*, Vol. 47, No. 1, pp. 1-21, February, 2020.

[18] K. K. Charles, P. Kline, Relational Costs and the Production of Social Capital: Evidence from Carpooling, *The Economic Journal*, Vol. 116, No. 511, pp. 581-604, April, 2006.

[19] L. Dong, W. Wu, Q. Guo, M. N. Satpute, T. Znati, D. Z. Du, Reliability-Aware Offloading and Allocation in Multilevel Edge Computing System, *IEEE Transactions on Reliability*, Vol. 70, No. 1, pp. 200-211, March, 2021.

[20] M. Jelokhani-Niaraki, N. N. Samany, M. Mohammadi, A. Toomanian, A hybrid ridesharing algorithm based on gis and ant colony optimization through geosocial networks, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, No. 2, pp. 2387-2407, February, 2021.

[21] A. Tafreshian, N. Masoud, Trip-based graph partitioning in dynamic ridesharing, *Transportation Research Part C: Emerging Technologies*, Vol. 114, pp. 532-553, May, 2020.

[22] R. Baldacci, V. Maniezzo, A. Mingozzi, An Exact Method for the Car Pooling Problem Based on Lagrangean Column Generation, *Operations Research*, Vol. 52, No. 3, pp. 422-439, May-June, 2004.

[23] J. Jung, R. Jayakrishnan, J. Y, Park, Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing, *Computer-Aided Civil and Infrastructure Engineering*, Vol.

31, No. 4, pp. 275-291, April, 2016.
[24] A. Kleiner, B. Nebel, V. A. Ziparo, A mechanism for dynamic ride sharing based on parallel auctions, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Catalonia, Spain, 2011, pp. 266-272.
[25] E. J. Dhulkefl, A. Durdu, H. Terziolu, A* and Dijkstra Algorithm, in: O. Gunday, E. Pehlivan, O. Minaz (Eds.), *Engineering Research Papers*, Gece Publishing, 2019.
[26] T. Roughgarden, J. R. Wang, The Complexity of the k-means Method, *24th Annual European Symposium on Algorithms*, Aarhus, Denmark, 2016, pp. 1-14.
[27] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, April, 2002.
[28] OpenStreetMap, www.openstreetmap.org.
[29] NYC, www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

Biographies

Jiaze Sun is a professor at Xi'an University of Posts & Telecommunications of China. He received a Ph.D. degree in computer science from Northwest University in 2015. His research interests include swarm intelligence algorithms and data mining.

Nan Han is an M.S candidate from Xi'an University of Posts & Telecommunications of China. Her research interests include data mining and multi-objective optimization.

Jianbin Huang is a professor at the School of Software, Xidian University of China. His research interests are in data mining and knowledge discovery.

Jiahui Deng is a Ph.D. candidate from Northwest University of China. Her research interest is multi-objective optimization.

Yang Geng is an M.S candidate from Xi'an University of Posts & Telecommunications of China. Her research interest is route planning.