Multi-rider Ridesharing Stable Matching Optimization

Hua Ke
Tongji University School of Economics and Management
Haoyang Li (2131369@tongji.edu.cn)
Tongji University Urban Mobility Institute https://orcid.org/0009-0001-0654-0003

Research Article

Keywords: Ridesharing, Stablity, Subsidy, Single-driver Multiple-rider

Posted Date: August 30th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3019025/v1

Multi-rider Ridesharing Stable Matching Optimization

Hua Ke[1,2] and Haoyang Li[1*]

[1]Urban Mobility Institute, Tongji university, Shanghai, 200082,China.
[2]School of Economics and Management, Tongji university, Shanghai, 200082,China.

*Corresponding author(s). E-mail(s): 2131369@tongji.edu.cn;
Contributing authors: hke@tongji.edu.cn;

Abstract

The rapid growth of private car ownership has led to significant issues such as traffic congestion and environmental pollution. Ridesharing has emerged as a promising solution to alleviate the negative impacts associated with private car usage. This paper focuses on the stability of ridesharing systems and establishes a single-driver multiple-rider ridesharing matching model. To solve this model, a filtering algorithm for the pre-matching set and a fast-solving algorithm for stable matching scheme are proposed. Furthermore, we introduce the concept of a subsidy distance upper limit into the ridesharing system. Remarkably, our findings indicate that with a limit of 0.1km, the distance saved generated by the subsidy amounts to 560.5% of the total subsidy. To validate our approach, we simulate ridesharing demand data using real taxi data, and design computational experiments to prove the computational efficiency of the filtering algorithm and fast-solving algorithm. The impact of various parameters on ridesharing systems is also explored.

Keywords: Ridesharing, Stablity, Subsidy, Single-driver Multiple-rider

1 Introduction

With the continuous growth of the world population and the rapid expansion of major cities, the choice of travel modes and travel frequency of urban residents have undergone profound changes, especially the increasing demand for mobility and privacy of car travel (Papageorgiou et al., 2003). Car ownership in major cities around the world is increasing by years, and the resulting urban traffic congestion and environmental pollution problems have become a serious obstacle to the development of major cities around the world. Although a car has four or more seats, many private car owners tend to drive alone. According to a survey, 80% of private car owners in Beijing drive alone during rush hours (Zhou et al., 2014). In the United States, the average number of riders per car during rush hours is 1.4 (Sivak, 2013). There is no doubt that vacant seats are a waste of increasingly scarce transportation resources, and sharing a car with multiple people is an important solution for effectively utilizing transportation resources.

Transportation network companies, such as Uber, Lyft, Dida, and Didi, offer various shared transportation services, including online ridesplitting and private car ridesharing. Online ridesplitting operates on the traditional taxi model, where drivers are the employees of the transportation network companies and seek to make a profit. On the other hand, private car ridesharing, also known as peer-to-peer ridesharing, involves both drivers and riders being users of the transportation network companies. In private car ridesharing, drivers are not motivated by profit but rather aim to save on travel costs. This distinction results in private car ridesharing being more efficient in terms of reducing vehicle travel distance compared to online ridesplitting since drivers in private car ridesharing do not need to sacrifice their travel time to pick up riders.

Matching is a critical area of research in the field of ridesharing, and matching solutions aim at maximizing system-wide benefits, such as the total travel distance saving, which has been thoroughly studied (Rodier et al., 2016). However, it is crucial to acknowledge that the optimal matching solution aimed at maximizing system-wide benefits may not align with the individual preferences of participants. Consequently, participants may often reject proposed matches and instead opt to find their own matches, pursuing choices that are more advantageous to them. This paper assumes that all participants are rational individuals and establishes a ridesharing matching model that ensures satisfaction for all participants while preventing any incentives for them to deviate from their current matches and seek new ones. We refer to this type of matching as "stable matching", and its specific definition will be discussed in Section 3.

The fundamental problem of stable matching was initially proposed by Gale and Shapley in their seminal paper in 1962 (Gale and Shapley, 1962). They investigated the concept of stable marriage matching, which represents the most basic form of matching problems. Although marriage matching problems possess key characteristics applicable to most matching problems, many-to-one matching is better suited for real-world scenarios, such as company hiring processes and college admissions. One of the most well-known examples of many-to-one matching is Roth's hospital intern matching problem introduced in 1986 (Roth, 1986). Recognizing the computational complexity associated with an increasing number of matches, Roth transformed the many-to-one matching problem into a one-to-one matching problem by leveraging participants' substitutable preferences.

However, in the context of ridesharing problems, the preferences of participants are non-substitutable due to constraints related to routes and time. There is a possibility that when a driver shares rides individually with multiple riders, they can benefit from the shared riders. However, when the driver shares rides with these multiple rides together, they do not derive any benefit from the shared rides. Consider a simple example in Fig. 1. d represents a driver and r represents a rider. When driver d_1 shares a ride with riders r_1 and r_2 respectively, the route is shown as the bold solid line, and the total distance saving is 4 units of distance. However, when the driver shares a ride with both riders r_1 and r_2 at the same time, the route is shown as the dotted line, and the total distance saved by all the three participants is -4 units of distance. From the above example, it can be observed that driver d_1 is willing to individually ride with the two passengers but is not willing to ride with both the passengers simultaneously. This is counterintuitive because in traditional many-to-one matching problems when a hospital interviews two medical students and is satisfied with both of them, it is willing to hire both students at the same time if there are enough positions available. In large-scale ridesharing matching problems,

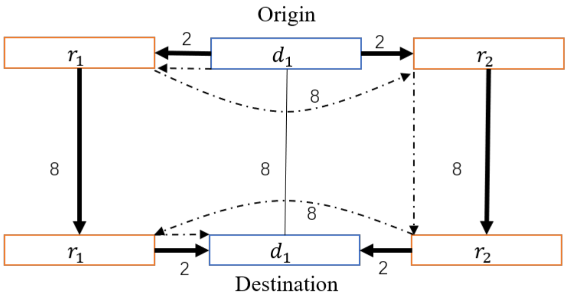


Fig. 1 Preference characteristics in ridesharing: An example

since many-to-one matching in ridesharing cannot be transformed into one-to-one matching, finding feasible solutions needs a large amount of calculation.

To fill the research gap, in this paper, we assume that participants make reservations and choose whether to participate as a driver or rider in the ridesharing, with provided information including origin location, destination, estimated departure time, and latest arrival time. We first propose a feature-based filtering algorithm for ridesharing problem to identify all potential matches, referred to as the pre-matching set in this paper, then construct a mathematical integer programming model that considers the stable matching solution of single-driver multiple-rider, and then propose a fast-solving algorithm to the integer programming model. We also introduce the concept of subsidy into the model, and discuss the impact of subsidy on the system performance. The contributions of this study can be summarized as follows:
(i) A single-driver multiple-rider ridesharing matching model is constructed, and based on the characteristics of the ridesharing problem, a filtering algorithm is proposed that greatly improves the efficiency of finding the pre-matching set.
(ii) Under the condition of average distribution, a fast-solving algorithm to the integer programming model is proposed, which can effectively obtain the exact solution

of stable matching.
(iii) The concept of subsidy is introduced into the single-driver multiple-rider ridesharing matching model, and the impact of subsidy on system performance is discussed. The results indicate that, for transportation network companies, a lower subsidy level can lead to higher return on investment.

The remainder of this paper is organized as follows: In Section 2, we discuss the literature related to our research. In Section 3, we define stable matching in the multi-rider ridesharing scenario and introduce optimization models and algorithms for finding stable matching solutions. We also present an optimization model that incorporates subsidy. In Section 4, we design computational experiments using real taxi data from New York City to explore the impact of subsidy and various parameters on system performance. Finally, in Section 5, we provide a conclusion of this paper.

2 Literature review

In recent years, numerous studies have considered the matching problem in ridesharing. Furuhata et al. (2013) gave a detailed classification and analysis of the matching in ridesharing, pointing out that the main challenges of ridesharing matching include dynamic ridesharing matching, pricing research, high-dimensional matching, trust and reputation, and mechanism design.

In the research on ridesharing matching mechanism design, Agussurja et al. (2019) investigated the application of ridesharing in meeting the last-mile transportation demand by constructing a two-stage Markov decision process framework to solve vehicle dispatch decisions. Kumar and Khani (2021) developed a ridesharing matching algorithm based on public transit scheduling for addressing the first-mile and last-mile travel problem, generating feasible matches by considering the shortest path to the public transit schedule. Fielbaum et al. (2021) improved overall matching efficiency by requiring users to walk to fixed pickup points, leading to a reduction of over 80% in ride rejections with users walking an average of approximately one minute. Khemiri et al. (2023) proposed an integrated approach in which a fleet of self-driving electric vehicles is coordinated to provide on-demand travel services and enable ridesharing among multiple passengers.

In the study of matching efficiency, Agatz et al. (2011) proposed a dynamic matching mechanism based on a fixed time period, which decomposes the dynamic matching problem into static matching within a fixed period. Agatz et al. (2012) provided a comprehensive review of optimization research in dynamic ridesharing problems. Due to the potential influx of a large number of matching requests within a short time, Najmi et al. (2017) clustered participants into smaller subsets based on the fixed rolling period decision framework proposed by Agatz (Agatz et al., 2011), and developed heuristic algorithms with the ability to solve large-scale instances. Hong et al. (2017) focused on ridesharing behavior during commuting and proposed a commuting ride matching method based on clustering trajectories using data mining techniques. Simonetto et al. (2019) transformed the ridesharing problem into a linear assignment problem between fleet vehicles and user travel requests in a federated architecture, significantly improving algorithm speed. Tafreshian and Masoud (2020) modeled the one-to-one dynamic matching problem as a bipartite graph problem, and reduced the complexity of the problem by decomposing the bipartite graph into multiple subgraphs. In solving the matching problem, heuristic algorithms such as greedy algorithms (Calvo et al., 2004), bee algorithms (Teodorovic and Dell'Orco, 2005) and genetic algorithms (Herbawi and Weber, 2011), have been widely applied. It can be observed that ridesharing matching research often decomposes large-scale optimization problems to enhance the efficiency of matching algorithms.

In the research on cost allocation in matching, due to the reliance of ridesharing on platforms, many papers have focused on pricing rather than cost allocation in the context of ridesharing (Santos and Xavier, 2015; Sun et al., 2019; Bian and Liu, 2019; Turan et al., 2020; Li et al., 2021). In cost allocation research, Bistaffa et al. (2015) developed a cost allocation algorithm based on the concept of the core in cooperative games to meet the demand for large-scale solving. Gopalakrishnan et al. (2016) introduced inconvenience costs resulting from collaboration among multiple individuals in ridesharing costs and studied the cost and benefit of dynamic ridesharing based on concepts of individual preference rationality and sequential fairness. Wang et al. (2018b) investigated the impact of various cost allocation strategies on the overall equilibrium of the transportation system by establishing a mode choice model for heterogeneous travelers in ridesharing. Chen et al. (2019) designed a double auction-based discounted transaction mechanism for dynamic cost allocation, which satisfies both individual rationality and incentive compatibility. Lu and Quadrifoglio (2019) introduced the concept of the core in cooperative games to achieve fair cost allocation among participants in the matching process. In this study, we propose two different cost allocation schemes: average allocation and proportional allocation based on the distance of participants in ridesharing. We compare and analyze the impact of these two allocation schemes on the matching process.

As the ridesharing model is further studied, researchers have recognized that matching solutions aimed at maximizing system-wide benefits may be unstable, leading to participants rejecting proposed matches and seeking their own matches. Additionally, when a participant acts as a driver in ridesharing, he may be willing to accommodate only one rider, but there is also a possibility that he would be willing to accommodate multiple riders in order to save on travel costs. Wang et al. (2018a) solved the matching problem in ridesharing based on stable matching theory, and used mathematical programming methods to obtain stable and nearly stable solutions. In their many-to-one ridesharing matching discussion, the efficiency of obtaining stable matching solutions was relatively low, and the saving costs were evenly distributed among the members, which is unfair to long-distance riders. In this paper, we propose an efficient solving algorithm based on the characteristics of the multi-rider ridesharing matching problem. Additionally, we introduce a more reasonable cost-saving allocation scheme and incorporate the concept of subsidy to improve the system performance. Ma et al. (2019) discussed many-to-one ridesharing matching considering match stability and proposed a heuristic algorithm to establish stable matches. However, in their process of finding the pre-matching set, they only judged the feasibility of many-to-one matches based on the feasibility of one-to-one matching, which is proved unreasonable in this paper. In this paper, we propose reasonable assumptions to simplify the solution of the pre-matching set and validate the efficiency of the pre-matching set solving algorithm using real-world data.

3 Methodology

In this study, we consider the situation where participants need to make a reservation on the platform and choose to participate as a driver or rider in the ridesharing system. We assume that the participants are allowed to choose only one role during the ride, and the participants are divided into two independent sets of drivers and riders. $p \in P$ denotes participant p, $i \in D$ denotes driver i, $j \in R$ denotes rider j, and $P = D \cup R$. Participant p needs to provide coordinates of their origin $o_p = (olon_p, olat_p)$, destination $d_p = (dlon_p, dlat_p)$, estimated departure time ot_p and latest arrival time dt_p. When driver i chooses to travel with multiple riders J, a coalition $c_i = (i, J)$ centered on the private car of driver i is formed. The pre-matching set, denoted as A, comprises a collection of ridesharing coalitions that satisfy both route and time constraints, ensuring benefits for all participating members. Within the pre-matching set, the coalitions are referred to as meeting internal stability. To obtain the final matching scheme, the optimization objective guides the selection of mutually non-conflicting coalitions from the pre-matching set. These mutually non-conflicting coalitions are characterized as meeting stability between coalitions.

In the following subsections, we will discuss the meanings of internal stability and stability between coalitions respectively, and provide solutions for pre-matching set and stable matching scheme.

3.1 Internal Stability of Coalition

For driver i, the result of the match is a route planned by the platform. For the ridesharing coalition $c_i = (i, J)$ formed by driver i and multiple riders J, we define a route $TR = (tr_1, tr_2, ...tr_{nu})$ composed of nu nodes, where nu is the number of nodes which is twice the members in the coalition. Driver i needs to pass through these nodes to pick up and drop off riders for completing their journey. In this route, tr_1 represents the origin point of driver i, also denoted as o_i, and tr_{nu} represents the destination of driver i, also denoted as d_i. For rider $j \in J$, his origin o_j and destination d_j must be in the route TR, and o_j must appear before d_j, which ensures that the driver picks up the rider at the rider's origin point and drops off the rider at the rider's destination. In addition to maintaining a specific sequence of nodes, the driver is also obligated to reach each node within a specific time window. Driver i should pick up rider j within the waiting time threshold of rider j, or rider j can reach the appointed place (assumed as the origin of rider j) within the waiting time threshold of driver i. For simplicity, we suppose the maximum tolerated waiting time for participants is a fixed value WT. Participants have their own latest arrival times, and the driver should arrive the node earlier than the corresponding latest arrival time on the route.

The aforementioned ridesharing routing generation can be considered equivalent to solving a variant of the traveling salesman problem with time windows and precedence constraints (TSP-TWPC). This can be formulated as a mixed-integer programming model, expressed as follows.

$$\min \sum_{m \in cn_i; n \in cn_i; m \neq n} x_{mn} d_{mn} \tag{1a}$$

$$\text{subject to:} \quad \sum_{m \in cn_i; m \neq n} x_{mn} = 1, \quad \forall n \in cn_i, \tag{1b}$$

$$\sum_{n \in cn_i; m \neq n} x_{mn} = 1, \quad \forall m \in cn_i, \tag{1c}$$

$$y_{o_j} < y_{d_j}, \quad j \in J, \tag{1d}$$

$$y_{o_i} = 1, y_{d_i} = nu, \tag{1e}$$

$$t_n = \sum_{m \in cn_i} x_{mn}(t_m + t_{mn} + wt_m) \tag{1f}$$

$$wt_m = \begin{cases} ot_j - t_{o_j}, & j \in J, ot_j > t_{o_j} \\ 0, & otherwise \end{cases} \tag{1g}$$

$$ot_j - WT < t_{o_j} < ot_j + WT, \quad j \in J, \tag{1h}$$

$$dt_j > t_{d_j}, \quad j \in J, \tag{1i}$$

$$dt_i > t_{d_i}, \tag{1j}$$

$$y_m \in \{0, 1, 2, ..., nu\}, \tag{1k}$$

$$x_{mn} \in \{0, 1\}, \quad \forall m \in cn_i; n \in cn_i; m \neq n. \tag{1l}$$

cn_i: All the nodes of all members in coalition c_i.
y_m: Integer variable with a value range of $\{1, 2, 3, ...nu\}$, indicating the position of node m in the route;
d_{mn}: The distance between node m and n;
t_{mn}: The required time from node m to n;
t_m: The moment of driver arrival at node m;
wt_m: The waiting time of driver at node m;
x_{mn}: Binary variable that takes the value 1 when node n is the next adjacent node to node m (i.e., when $y_n = y_m + 1$), and 0 otherwise.

In the mixed integer programming model (1), constraints (1b) and (1c) denote that the origin and destination points of each member in coalition c_i are visited and visited only once. Constraints (1d) and (1e) denote that the origin point of participants is visited before the destination point. Constraints (1f)-(1h) denote that the driver and riders can converge at the origin points within their maximum tolerated waiting time. Constraints (1i) and (1j) denote that participants can arrive at their destination before their latest arrival time. Constraints (1k) and (1l) denote the range of parameter values.

The TSP-TWPC problem has been proven to be an NP-hard problem. Besides, in a ridesharing scenario, there is an upper limit on the number of riders that each driver is willing to ride, which can be set as a fixed value Q, and $Q > 0$. It is generally believed that drivers can accept up to four stops during a journey, meaning they can ride up to two riders (Wang et al., 2018a). When driver i shares a ride with rider j, there is only one possible route $TR_i = (o_i, o_j, d_j, d_i)$. When driver i shares a ride with rider j and rider k, there are six possible routes, as shown in Fig. 2. The feasibility of an coalition in terms of route and time can be solved by traversal. However, since each coalition has many constraints, in large-scale ridesharing problems, it takes a lot of time to traverse all coalitions.

Within an internal stable coalition, it is not enough to meet the feasibility of the

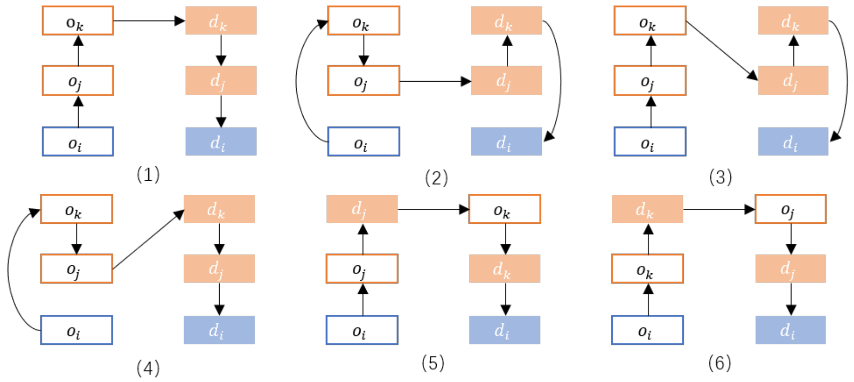


Fig. 2 The possible routes.

participant's travel time and precedence. The purpose of participating in ridesharing is to save travel costs. To ensure stability, we need to prevent the coalition from splitting. We assume the saved cost is evenly distributed among coalition members. Consider a ridesharing coalition $c_i = (i, J)$. When driver i shares a ride with rider $j \in J$, the average saved cost of the two may be higher than the average saved cost of the ridesharing coalition $c_i = (i, J)$. In this case, driver i and rider j will detach from the ridesharing coalition $c_i = (i, J)$ and form a new coalition. In this paper, due to the direct correlation between saved travel costs and saved travel distance, we assume that the participant's utility is equivalent to the saved distance by each participant in ridesharing. The utility value calculation of each participant is expressed as follows.

$$sd_{(i,J)} = d_i + \sum_{j \in J} d_j - d_{(i,J)}, \tag{2a}$$

$$\mu_i(i, J) = \omega_i sd_{(i,J)}, \tag{2b}$$

$$\mu_j(i, J) = \omega_j sd_{(i,J)}, \quad \forall j \in J, \tag{2c}$$

$$\omega_i + \sum_{j \in J} \omega_j = 1. \tag{2d}$$

d_p: Travel distance of participant p when not participating in ridesharing;
$d_{(i,J)}$: Travel distance of coalition (i, J) when participating in ridesharing;
$sd_{(i,J)}$: Saved travel distance of coalition (i, J) in ridesharing;
$\mu_p(i, J)$: Utility value of participant p in coalition (i, J);
ω_p: The distribution proportion of participant p in the coalition's saved travel distance.

Equation (2a) is the calculation of coalition travel distance savings. Equations (2b) and (2c) are the calculation of the participant's utility value. The calculation of ω_p is related to the allocation method. In this paper, two different allocation methods will be considered, and the specific calculation process will be discussed in subsequent sections. Equation (2d) indicates that the saved travel distance by the coalition is fully shared among all participants.

To ensure the internal stability of the coalition (i, J), the following constraints need to be met.

$$s_{(i,J)} > 0, \tag{3a}$$
$$\mu_i(i, J) > \mu_i(i, J'), \quad \forall J' \subset J, \tag{3b}$$
$$\mu_j(i, J) > \mu_j(i, J'), \quad \forall J' \subset J. \tag{3c}$$

Equation (3a) constrains the feasibility of saving distance in the coalition (i, J). Only when the coalition saving distance is positive, will the participants form a ridesharing coalition. Equations (3b) and (3c) ensure that the utility of part of one coalition is less than that of the coalition (i, J), which is a necessary condition for the coalition not to be split into small coalitions.

In summary, internal stability of coalition refers to the condition where the members simultaneously satisfy the feasibility of matching in terms of routes and time (satisfying constraints 1b-1l). Additionally, forming the coalition should be profitable, and the members should not have the incentive to split from the coalition (satisfying constraints 3a-3c).

3.2 Stability between Coalitions

Describing the stability between coalitions requires the introduction of the concept of blocking groups. For a more intuitive description of blocking groups, we can describe the matching scheme as a mapping α from riders to drivers.

$$\alpha : R \to D \tag{4}$$

In the mapping α, if there are $i \in D$ and $j, k \in R$ such that $\alpha(j) = \alpha(k) = i$, $\alpha^{-1}(i) = j$ or k, it means that driver i is matched with riders j and k. $i' \succeq_j \alpha(j)$ represents that for rider j, the utility of matching with driver i' is higher than that of matching with driver $\alpha(j)$ in matching scheme α.

Definition 1: A blocking group of a matching scheme α is defined as follows: If there exist $i' \in D$ and $J' \subset R$ that satisfy the basic matching conditions and $J' \succeq_i \alpha^{-1}(i)$, $i' \succeq_j \alpha(j)$, (i', J') is called a blocking group of matching scheme α.

Definition 2: We define such a matching scheme as a stable matching scheme, that there is no blocking group of the matching scheme.

3.3 Finding Pre-Matching Set

Finding the pre-matching set necessitates traversing all possible combinations of participants to determine whether they meet the internal stability constraints or not. However, in large-scale ridesharing problems, this approach can be inefficient as most infeasible combinations do not require traversing all internal stability constraints, resulting in a significant number of unnecessary calculations. In our study, we propose a new approach to mitigate these redundant computations and improve efficiency.

When the upper limit of riders that drivers are willing to ride is 2, the solution of finding the pre-matching set can be decomposed into two sub-problems: finding the pre-matching set of one-to-one and finding the pre-matching set of one-to-two. The main difficulty in finding the pre-matching set is concentrated in the solution of the latter sub-problem, while the solution of the former sub-problem is relatively simple and can be used as a filtering condition to simplify the solution of the latter sub-problem. Based on this, three rules are proposed to mitigate these redundant computations.

Rule 1: If the travel time of any driver i or rider j in the one-to-one matching exceeds the latest arrival time of the driver, then all coalitions containing i and j are infeasible.

Rule 2: When the one-to-one matching between driver i and rider j is feasible, the coalition carrying a third person only needs to consider three of all possible routes in Fig. 2.

Rule 3: The judgment of all constraints is implemented in the nested structure of judgment statements in the filtering algorithm.

It can be determined that when a rider is introduced into a one-to-one coalition, the total time used by the many-to-one coalition must be greater than that of the one-to-one coalition. If the reason for the instability of the one-to-one coalition is that driver i exceeds its latest arrival time due to taking rider j, then no matter how many more riders are taken, driver i will exceed its latest arrival time. This is also the basis for setting **Rule 1**. For **Rule 2**, when the one-to-one matching between driver i and rider j is feasible, it is only necessary to judge whether routes (1) (3) (5) in Fig. 2 satisfiy the stability constraint when k joins the coalition. For **Rule 3**, taking advantage of the characteristic that constraints in ridesharing problems need to be satisfied simultaneously, once a constraint does not meet the condition in the nested structure of judgment statements, it will jump out of judgment and will not continue to judge subsequent constraints. This can greatly reduce the number of constraints that need to be traversed. Based on the above rules, algorithm 1 and algorithm 2 for finding the pre-matching set is constructed.

3.4 Finding Stable Matching Scheme

The final stable matching scheme consists of coalitions that satisfy both the internal stability of the coalition and stability between coalitions. In subsection 3.3, we have

Algorithm 1: Search for one-to-one feasible matching coalitions

Input: $S = D \cup R$, O_s, WT, D_s, OT_s and DT_s
Output: A_1
for i *in* D **do**
    for j *in* R **do**
        Compute routed(TR_s); routet(TR_s);
        $wt_{i,j} = routet(o_i, o_j) + ot_i - ot_j$;
        if $routed(o_i, o_j, d_j, d_i) \leq routed(o_i, d_i) + routed(o_j, d_j)$ *and* $-WT \leq wt_{i,j} \leq WT$ **then**
            if $wt \leq 0$ *and* $o_i + routet(o_i, o_j, d_j, d_i) \leq d_i$ **then**
                $dwt_i = 1$;
                if $o_j + routet(o_j, d_j) - wt \leq d_j$ **then**
                    $A_1 = A_1 \cup \{(i, j)\}$;
            if $wt > 0$ *and* $o_i + routet(o_i, o_j, d_j, d_i) + wt \leq d_i$ **then**
                $A_1 = A_1 \cup \{(i, j)\}$;
                $dwt_i = 1$;

obtained pre-matching set A that meets the internal stability of the coalition. The process of finding a stable matching scheme is to select possible coalitions that do not conflict with **Definition 2** from the pre-matching set A. We set the total saved travel distance as the maximization objective, which is considered to be positively significant for participants and social welfare. For participants, their utilities are assumed to be equivalent to saved travel distance. For society, maximizing the total saved travel distance means reducing the total vehicle travel distance, which is beneficial to alleviate traffic congestion and reduce environmental pollution caused by traffic. The above solution process can be described by the following integer programming model.

$$\max \sum_{(i,J)\in A} c_{iJ} x_{iJ} \tag{5a}$$

$$\text{subject to: } \sum_{i\in D} x_{iJ} \leq 1, \quad \forall i \in D, \tag{5b}$$

$$\sum_{J\subseteq R} x_{iJ} \leq 1, \quad \forall J \subseteq R, \tag{5c}$$

$$\sum_{J'\succeq_i J} x_{iJ'} + \sum_{j\in J} \sum_{I'\succeq_j I} x_{I'j} + x_{iJ} \geq 1, \quad \forall (i,J) \in A, \forall (I,j) \in A, \tag{5d}$$

$$x_{iJ} \in \{0,1\}, \quad \forall (i,J) \in A. \tag{5e}$$

In this integer programming model, constraints (5b) and (5c) are used to constrain drivers and riders to participate in at most one match. Constraint (5d) means

Algorithm 2: Search for one-to-two feasible matching coalitions

Input: $S = D \cup R$, O_s, WT, D_s, OT_s and DT_s

Output: A

for i *in* D **do**

  for j *in* R **do**

    if $wt_{ij} \leq WT$ *and* dwt_i*==1* **then**

      if $wt_{ij} \leq 0$ **then**

        $dwt_{ij} = -wt_{ij}$; $rwt_{ij} = 0$

      else

        $rwt_{ij} = wt_{ij}$;$dwt_{ij} = 0$

      for k *in range(j,len(R))* **do**

        if $wt_{ij} > -WT$ *and* $wt_{ik} \leq WT$ **then**

          if $-WT \leq wt_k = ot_j + routet(o_j, o_k) + rwt_{i,j} - ot_k \leq WT$ **then**

            if $wt_k \leq 0$ **then**

              $rwt_{kj} = -wt_k$; $rwt_{jk} = 0$

            else

              $rwt_{jk} = wt_k$; $rwt_{kj} = 0$

            if $ot_i + routet(o_i, o_j, o_k, d_k, d_j, d_i) + dwt_{ij} + rwt_{kj} \leq dt_i$ **then**

              if $ot_j + routet(o_j, o_k, d_k, d_j) + rwt_{kj} \leq dt_j$ **then**

                $A_2 = A_2 \cup \{(i, j, k)\}$

            if $ot_i + routet(o_i, o_j, o_k, d_j, d_k, d_i) + dwt_{ij} + rwt_{kj} \leq dt_i$ **then**

              if $ot_j + routet(o_j, o_k, d_j) + rwt_{kj} + rwt_{i,j} \leq dt_j$ *and* $ot_k + routet(o_k, d_j, d_k) + rwt_{kj} + rwt_{j,k} \leq dt_j$ **then**

                $A_2 = A_2 \cup \{(i, j, k)\}$

          if $-WT \leq wt_k = ot_j + routet(o_j, d_j) + rwt_{i,j} - ot_k \leq WT$ **then**

            if $wt_k \leq 0$ **then**

              $rwt_{kj} = -wt_k$; $rwt_{jk} = 0$

            else

              $rwt_{jk} = wt_k$; $rwt_{kj} = 0$

            if $ot_i + routet(o_i, o_j, d_j, o_k, d_k, d_i) + dwt_{ij} + rwt_{kj} \leq dt_i$ *and* $ot_k + routet(o_k, d_k) + rwt_{jk} \leq dt_k$ **then**

              $A_2 = A_2 \cup \{(i, j, k)\}$

$A = A_1 \cup A_2$

that matching can only be carried out under the premise that participants in the matching coalition have no better matching, i.e., to avoid the occurrence of blocking groups to ensure the stability between coalitions.

Assume that the total saved distance of the coalition is evenly distributed among the members of the coalition, i.e. $\omega_i = \omega_j = \frac{1}{n}$, where n is the number of members in the coalition. The utility value of the participants is equal to the average value of the total saved distance of the coalition.

Inference 1: In the context of an average distribution, within the pre-matching set, the coalition exhibiting the highest average saved distance can not be a blocking group. Moreover, it is imperative that the coalition with the highest average saved distance is included in the final stable matching scheme.

Based on **Inference 1**, we can build a fast-solving algorithm as illustrated in Algorithm 3 for stable matching problem.

Under the condition of average distribution, the fast-solving algorithm 3 achieves

Algorithm 3: Search for stable matching

Input: A

Output: SM

```
sort_A = A.sort(by = averagesavingdistance);
while sort_A do
    SM.append(sort_A(rowindex = 0));
    sort_A.drop(driverindex = sort_A(0, driverindex) or
      riderindex = sort_A(0, riderindex));
```

the same results as the integer programming model (5). By observing constraint (5d), it can be known that when (i, J) is the coalition with the highest average saved distance, (i, J') and $(I', j), j \in J$ are empty set. According to constraints (5b) and (5c), all x_{iJ} of coalitions containing i or rider j equal 0, and all constraints corresponding to these coalitions can be deleted. This means deleting all coalitions involving i or $j \in J$ in set A does not affect on the solution of integer programming.

3.5 Ridesharing Matching Model with Subsidy

When disregarding the stability between coalitions, the integer programming model for determining the matching scheme with the objective of maximizing the total saved travel distance is formulated as follows. The solution to the integer programming model (6) can be considered as the optimal solution for ridesharing.

$$\max \sum_{(i,J)\in A} s_{iJ} x_{iJ} \tag{6a}$$

$$\text{subject to: } \sum_{i \in D} x_{iJ} \leq 1, \quad \forall i \in D, \tag{6b}$$

$$\sum_{J \subseteq R} x_{iJ} \leq 1, \quad \forall J \subseteq R, \tag{6c}$$

$$x_{iJ} \in \{0,1\}, \quad \forall (i,J) \in A. \tag{6d}$$

In the unstable matching schemes, part of blocking groups is formed because some participants choose to match with other participants who can only help save a little more travel distance. However, those blocking groups may cause a significant decrease in the total saved travel distance by the entire system. In this paper, we consider introducing a subsidy distance upper limit, which means that the platform is willing to subsidize a certain distance upper limit for participants in blocking groups to help the solution approach the optimal solution. The ridesharing matching model after introducing subsidy can be described by the following integer programming model.

$$\max \sum_{(i,J) \in A} c_{iJ} x_{iJ} \tag{7a}$$

$$\text{subject to: } \sum_{i \in D} x_{iJ} \leq 1, \quad \forall i \in D, \tag{7b}$$

$$\sum_{J \subseteq R} x_{iJ} \leq 1, \quad \forall J \subseteq R, \tag{7c}$$

$$\begin{aligned} &\sum_{J' \succeq_i J} x_{iJ'} + \sum_{j \in J} \sum_{I' \succeq_j I} x_{I'j} + x_{iJ} + \sum_{J'' \succeq_i S(i,s)} x_{iJ''} + \\ &\sum_{j \in J} \sum_{I'' \succeq_j S(j,s)} x_{I''j} \geq 1, \quad \forall (i,J) \in A, \forall (I,j) \in A \end{aligned} \tag{7d}$$

$$x_{iJ} \in \{0,1\}, \quad \forall (i,J) \in A. \tag{7e}$$

$J'' \succeq_i S(i,s)$ represents that under the condition that the subsidy distance upper limit is s, driver i accepts to share a ride with the set of riders J''. Constraint (7d) indicates that when the utility difference of a participant between different ridesharing coalitions is less than the subsidy distance upper limit s, it is permissible for that participant to be in a blocking pair. Constraint (7d) relaxes constraint (5d).

4 Computational Experiments

In the computational experiments, actual taxi data is used to simulate the demand data for ridesharing. We explore the effects of different allocation schemes, subsidy distance upper limits, waiting time threshold, participation rates, and the proportion of drivers among participants in the ridesharing system. Python is used as the programming language and Gurobi 10.0.1 is used as the integer programming solver. All experiments are conducted on a Windows 10 machine with 8GB RAM and an

Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz-1.80 GHz.

4.1 Simulation Setup

The data used in this paper is sourced from Trip Record Data (TLC), which comprises various fields capturing information about taxi trips, including pick-up and drop-off dates/times, locations, trip distances, itemized fares, rate types, payment types, and driver-reported rider counts. Specifically, we select the Green Taxi data during the peak hours of 8:00-10:00 on January 15th, 2016, a Tuesday, in the New York area. Following an initial data cleaning process, we filter the dataset to include samples where the driver reported a single rider, resulting in a total of 5182 ridesharing demand instances.

For the purpose of this study, we assume that a portion of these Green Taxi riders choose to participate in ridesharing and are assigned identities through random sampling. The latitude and longitude coordinates of the pick-up and drop-off locations are considered as the origin and destination coordinates for both the driver and the rider. The pick-up time represents the estimated departure time, while the drop-off time indicates the latest allowable arrival time. The trip distance is defined as the driving distance between the two locations. To ensure the accuracy of the experiment, we obtain the trip distance and travel time after the matching process by utilizing a map's route planning API.

To provide a more intuitive assessment of the parameter impact, we establish the following indicators.
Successful matching rate: The ratio of the number of successfully matched participants and the total number of participants. Drivers and riders can be independently calculated for matching successful rates to evaluate the matching effect of each group.
Distance saving rate: The ratio of the total saved travel distance of successfully matched participants and the total travel distance of all participants.
Average participants individual distance saving rate: The ratio of the average saved travel distance of successful participants and the average travel distance of all participants.
Cost of stable matching: The difference between the total saved travel distance of the optimal matching and the total saved travel distance of the stable matching as a proportion of the total saved travel distance of the optimal matching.

In subsequent experiments, unless otherwise specified, we assume that the participation rate in ridesharing is 75%; the rate of drivers to riders is 1:2; the waiting time threshold for participants is 5 minutes; the distribution method for distance savings in the matching coalition is average distribution; and a driver can ride at most two riders on a trip.

4.2 Solution Efficiency

Under the aforementioned experimental conditions, the pre-matching set is solved using the filtering algorithm and the traditional traversal algorithm respectively. The

CPU time required for each algorithm to solve the pre-matching set is recorded and is presented in Table 1. It can be observed that the filtering algorithm proposed in this paper achieves a solution in just 5.3 minutes, demonstrating significantly higher efficiency compared to the traditional traversal algorithm.

A comparison of the CPU time required to find the stable matching scheme using the fast-solving algorithm and the Gurobi solver is presented in Table 2. It is evident that the fast-solving algorithm yields results within 0.4 seconds, whereas the Gurobi solver takes nearly 132 seconds to complete the computation. This indicates that the fast-solving algorithm exhibits significantly improved efficiency compared to the Gurobi solver.

Table 1 Comparison of CPU time required to solve pre-matching set

Project	Algorithm proposed in this paper			Traversal algorithm
	Algorithm 1	Algorithm 2	Total	
CPU Time(min)	2.0	3.3	5.3	1673.7
Number of pre-matching set	1687	11100	12787	12787

Note: The solution results of the pre-matching set are same, which indirectly verifies that the algorithm proposed in this paper is equivalent to the traditional traversal algorithm.

Table 2 Comparison of CPU time required to find stable matching

Indicator	fast-solving algorithm	Gurobi solver
CPU time(s)	0.4	132.3
Total successful matching rate(%)	50.7	50.7
Total distance savings rate(%)	32.2	32.2
Cost of stable matching rate(%)	5.8	5.8

Note: The results obtained by the two algorithms is completely the same, which indirectly verifies that the fast-solving algorithm proposed in this paper is equivalent to the integer programming model.

4.3 Impact of Distribution Method

In the basic model, we assume that the saved distance within a coalition is evenly distributed among all its members. However, this approach may be unfair to certain members, particularly drivers, who have to travel longer distances within the coalition. In this section, we introduce a new allocation method that considers the individual distance traveled by each participant in ridesharing. In the new allocation method, the distance traveled by participant p in ridesharing is denoted as sd_p. The calculation formula for determining the distribution proportion of participant p in the saved distance of coalition (i, J) is as follows.

$$\omega_i = \frac{sd_i}{sd_i + \sum_{j \in J} sd_j}, \tag{8a}$$

$$\omega_j = \frac{sd_j}{sd_i + \sum_{j \in J} sd_j}, \quad \forall j \in J \tag{8b}$$

Keeping other basic assumptions unchanged, the results of comparing different allocation methods are shown in Table 3.

The results indicate that the shared distance allocation is better than the average allocation, which can be explained as some coalitions for long-distance and short-distance sharing are no longer feasible, leading to the formation of more coalitions with similar shared distance, which improves the overall matching quality.

Table 3 Comparison of different allocation methods on matching results

Indicator	Shared distance allocation	Average allocation
Total successful matching rate(%)	52.8	50.7
Total distance savings rate(%)	36.5	32.2
Cost of stable matching rate(%)	5.2	5.8

4.4 Impact of Subsidy

To evaluate the benefits that subsidy brings to the platform, the indicator of yields rate of the platform is introduced. The yields rate of the platform is equal to the ratio of the increase in total save distance by subsidy and the total distance that the platform needs to subsidize. Fig. 3 shows that over 50% of participants save distance between 0 to 1km. In this paper, we set the subsidy distance upper limit as 0.1 to 1km with an interval of 0.1km. Fig. 4 shows that the total successful matching rate has an upward trend, but the driver successful matching rate and rider successful matching rate remain relatively stable, indicating that subsidy has an equally positive effect on the rider and driver participation rates. In Fig. 5, as the subsidy limit increases, the proportion of total saved distance has a significant increase, with the most significant increase being from a subsidy limit as 0 to 0.1km. In Fig. 6, as the subsidy distance limit increases, the yields rate of the platform rapidly decreases. When the subsidy distance upper limit is 0.1km, the platform's profit rate reaches as high as 565.5%, but when the subsidy distance upper limit is 1km, the yields rate of the platform becomes negative, and at this point, the platform will lose its motivation for subsidy. The cost of stability rate gradually decreases, indicating that as the subsidy increases, the stable constraints between coalitions are relaxed and the system gradually approaches the optimal solution.

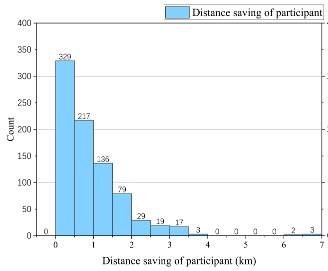


Fig. 3 Participants frequency distribution of saved distance

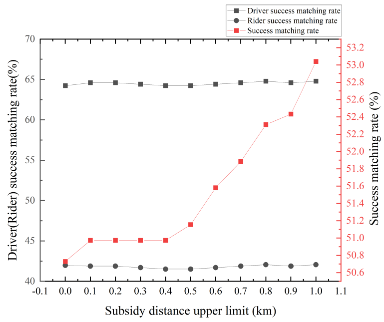


Fig. 4 The impact of subsidy distance upper limit on successful matching rate

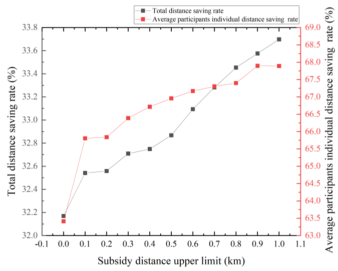


Fig. 5 The impact of subsidy distance upper limit on distance savings rate

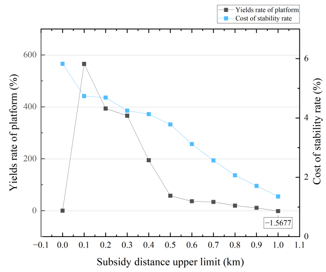
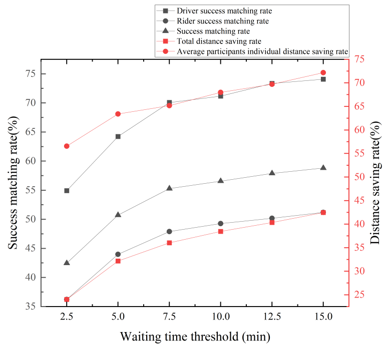


Fig. 6 The impact of subsidy distance upper limit on the yields rate of the platform and cost of stability

4.5 Impact of Waiting Time Threshold

We set the waiting time threshold to 2.5, 5, 7.5, 10, 12.5 and 15min respectively, and explore the impact of different waiting time thresholds on the ridesharing system. In Fig. 7, similar to the trends of the distance saving rate (total, average participants individual), the successful matching rates (driver, rider, total) all steadily increase with the increase of the waiting time threshold and the improvement rate gradually slows down, which indicates that the increase of the waiting time threshold helps to improve system performance, but when the threshold is raised to a certain level, the possible matches that are excluded by the waiting time threshold gradually decrease and the system performance improvement is limited. Fig. 8 shows that the cost of stability has an upward trend but remains at a low level (below 8% overall), indicating that there is a low overall difference between the optimal match and the stable match, i.e. the damage of stable matching to system performance is not high. However, this difference will increase with the increase of the waiting time threshold.



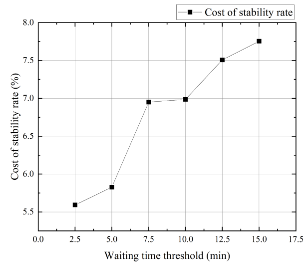
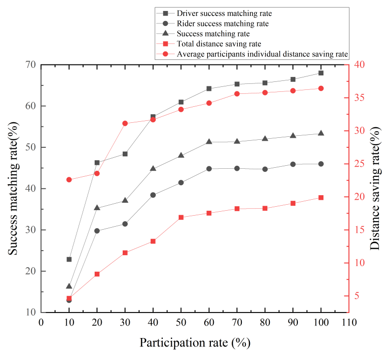


Fig. 7 The impact of waiting time threshold on successful matching rate and distance savings rate

Fig. 8 The impact of waiting time threshold on cost of stability

4.6 Impact of Participation Rate

We set the ridesharing participation rate from 10% to 100%, with an interval of 10%, and explore the impact of different participation rates on the ridesharing system. In Fig. 9, the successful matching rates (driver, rider, overall) all steadily increase with the increase of participation rate, and the distance saving rates (total, average participants individual) also show an upward trend, which indicates that as the number of participants in the ridesharing system increases, the overall performance and matching quality of the system are improved. In Fig. 10, the cost of stability has an upward trend, and this difference will increase with the increase of system scale due to the increase of feasible matching set size and more blocking groups.



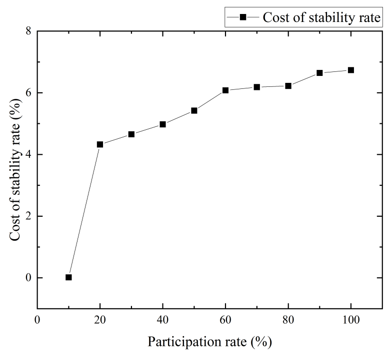


Fig. 9 The impact of participation rate on successful matching rate and distance savings rate

Fig. 10 The impact of participation rate on cost of stability

4.7 Impact of Driver Proportion

We set the driver proportion from 10% to 90%, with an interval of 10%, and explore the impact of different driver proportions on the ridesharing system. In Fig. 11, as the proportion of drivers among participants increases, the successful matching rate of drivers shows a downward trend, while the successful matching rate of riders shows an upward trend, and the overall matching successful rate first increases and then decreases. This can be explained as when there are fewer drivers and more riders, drivers have more choices and their successful matching rate will increase, while the successful matching rate of riders will decrease. In Fig. 12, the total saved distance rate first increases and then decreases, while the average participants individual distance savings rate shows an upward trend, indicating that in systems with a high proportion of drivers, the overall quality of matches is higher. In Fig. 13, similar to the trend of successful matching rate, the cost of stability rate first increases and then decreases, which can be explained as only when the proportion of drivers and riders is maintained at a certain ratio can a feasible matching set of a certain size be formed and blocking groups be generated.

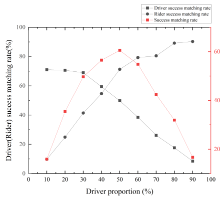


Fig. 11 The impact of participation rate on successful matching rate

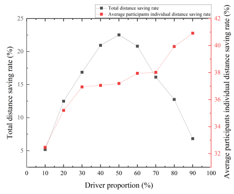


Fig. 12 The impact of participation rate on distance savings rate

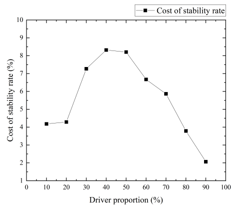


Fig. 13 The impact of participation rate on cost of stability

5 Conclusion

This paper presents a modeling approach for the many-to-one ridesharing matching problem, focusing on system optimality and stability. To address the pre-matching set, the problem is decomposed into two sub-problems, and the traversal object is selected based on ridesharing problem characteristics, leading to a significant reduction in the time required for model-solving. Moreover, an efficient algorithm is developed for solving the integer programming problem under average distribution, enabling solving large-scale ridesharing matching problems. The introduction of subsidies and their impact on stable matching solutions are also examined.

The study reveals that in ridesharing scenarios where the majority of saved travel distance is concentrated within short distances, a lower upper limit on subsidies can greatly enhance system performance, considering the platform yield rate as well. However, with an increase in the subsidy distance upper limit, the platform yield rate declines and may even become negative. Furthermore, the influencing factors of ridesharing matching problems are analyzed. It is observed that when the number of participants in the ridesharing system is fixed, increasing the waiting time threshold can contribute to improving system performance. Similarly, with a fixed proportion of drivers in the system, the successful matching rate rises as the participation rate increases. Additionally, when the participation rate of the ridesharing system is fixed, the average individual distance savings rate for participants tends to increase as the proportion of drivers rises.

In this research, there are several limitations that provide avenues for future investigation. This paper only discusses one allocation mechanism, excluding the average distribution. Further exploration could involve considering alternative allocation mechanisms within the ridesharing system to incentivize more participants to join and optimize resource allocation, thereby enhancing system efficiency and user satisfaction. Additionally, for the sake of model simplicity, this paper does not consider the negative impact of ridesharing on participants' utilities. In future research, it would be worthwhile to establish more realistic utility functions for participants and explore the influence of the negative effects of ridesharing on system stability.

Acknowledgments. This work is partly supported by the National Natural Science Foundation of China (NO. 41971252).

Declarations

Conflict of interest. On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical statement. This study does not violate and does not involve moral and ethical statement.

Appendix A

Proof of **Inference 1**: Proof by contradiction. Suppose that (i, J) is the feasible coalition with the largest average saved distance. In the final stable matching scheme α, i and J are not matched with each other. Since (i, J) is the coalition with the largest average saved distance, there must be $J \succeq_i \alpha(i)$, $i \succeq_j \alpha(j)$. (i, J) is a blocking group, which can be inferred that the matching scheme α is not a stable matching scheme. This contradicts our assumption and thus proves **Inference 1**.

References

Agatz N, Erera AL, Savelsbergh M, et al. (2011) Dynamic ride-sharing: A simulation study in metro atlanta. Procedia-Social and Behavioral Sciences 17:532–550

Agatz N, Erera A, Savelsbergh M, et al. (2012) Optimization for dynamic ride-sharing: A review. European Journal of Operational Research 223(2):295–303

Agussurja L, Cheng SF, Lau HC (2019) A state aggregation approach for stochastic multiperiod last-mile ride-sharing problems. Transportation Science 53(1):148–166

Bian Z, Liu X (2019) Mechanism design for first-mile ridesharing based on personalized requirements part i: Theoretical analysis in generalized scenarios. Transportation Research Part B-Methodological 120:147–171

Bistaffa F, Farinelli A, Chalkiadakis G, et al. (2015) Recommending fair payments for large-scale social ridesharing. In: Proceedings of the 9th ACM Conference on Recommender Systems, Proceedings of the 9th ACM Conference on Recommender Systems, pp 139–146

Calvo RW, de Luigi F, Haastrup P, et al. (2004) A distributed geographic information system for the daily car pooling problem. Computers & Operations Research 31(13):2263–2278

Chen W, Mes M, Schutten M, et al. (2019) A ride-sharing problem with meeting points and return restrictions. Transportation Science 53(2):401–426

Fielbaum A, Bai X, Alonso-Mora J (2021) On-demand ridesharing with optimized pick-up and drop-off walking locations. Transportation Research Part C-Emerging Technologies 126:103061

Furuhata M, Dessouky M, Ordóñez F, et al. (2013) Ridesharing: The state-of-the-art and future directions. Transportation Research Part B-Methodological 57:28–46

Gale D, Shapley LS (1962) College admissions and the stability of marriage. The American Mathematical Monthly 69(1):9–15

Gopalakrishnan R, Mukherjee K, Tulabandhula T (2016) The costs and benefits of sharing: Sequential individual rationality and sequential fairness. arXiv preprint arXiv:160707306

Herbawi W, Weber M (2011) Evolutionary multiobjective route planning in dynamic multi-hop ridesharing. In: European conference on evolutionary computation in combinatorial optimization, Springer, pp 84–95

Hong Z, Chen Y, Mahmassani HS, et al. (2017) Commuter ride-sharing using topology-based vehicle trajectory clustering: Methodology, application and impact evaluation. Transportation Research Part C-Emerging Technologies 85:573–590

Khemiri R, Naija M, Exposito E (2023) Dispatching and rebalancing for ride-sharing autonomous mobility-on-demand systems based on a fuzzy multi-criteria approach. Soft Computing 27(4):2041–2069

Kumar P, Khani A (2021) An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access. Transportation Research Part C-Emerging Technologies 122:102891

Li Y, Li X, Zhang S (2021) Optimal pricing of customized bus services and ride-sharing based on a competitive game model. Omega 103:102413

Lu W, Quadrifoglio L (2019) Fair cost allocation for ridesharing services – modeling, mathematical programming and an algorithm to find the nucleolus. Transportation Research Part B-Methodological 121:41–55

Ma R, Yao L, Song L, et al. (2019) A novel algorithm for peer-to-peer ridesharing match problem. Neural Computing and Applications 31(S1):247–258

Najmi A, Rey D, Rashidi TH (2017) Novel dynamic formulations for real-time ridesharing systems. Transportation Research Part E-Logistics and Transportation Review 108:122–140

Papageorgiou M, Diakaki C, Dinopoulou V, et al. (2003) Review of road traffic control strategies. Proceedings of The IEEE 91(12):2043–2067

Rodier C, Alemi F, Smith D (2016) Dynamic ridesharing: Exploration of potential for reduction in vehicle miles traveled. Transportation Research Record Journal of the Transportation Research Board 2542(1):120–126

Roth AE (1986) On the allocation of residents to rural hospitals: a general property of two-sided matching markets. Econometrica: Journal of the Econometric Society 54(2):425–427

Santos DO, Xavier EC (2015) Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. Expert Systems with Applications 42(19):6728–6737

Simonetto A, Monteil J, Gambella C (2019) Real-time city-scale ridesharing via linear assignment problems. Transportation Research Part C-Emerging Technologies 101:208–232

Sivak M (2013) Effects of vehicle fuel economy, distance travelled, and vehicle load on the amount of fuel used for personal transportation: 1970-2010. Tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute

Sun L, Teunter RH, Babai MZ, et al. (2019) Optimal pricing for ride-sourcing platforms. European Journal of Operational Research 278(3):783–795

Tafreshian A, Masoud N (2020) Trip-based graph partitioning in dynamic ridesharing. Transportation Research Part C-Emerging Technologies 114:532–553

Teodorovic D, Dell'Orco M (2005) Bee colony optimization–a cooperative learning approach to complex transportation problems. Advanced OR and AI Methods in Transportation 51:51–60

Turan B, Pedarsani R, Alizadeh M (2020) Dynamic pricing and fleet management for electric autonomous mobility on demand systems. Transportation Research Part

C-Emerging Technologies 121:102829

Wang X, Agatz N, Erera A (2018a) Stable matching for dynamic ride-sharing systems. Transportation Science 52(4):850–867

Wang X, Yang H, Zhu D (2018b) Driver-rider cost-sharing strategies and equilibria in a ridesharing program. Transportation Science 52(4):868–881

Zhou G, Huang K, Mao L (2014) Design of commute carpooling based on fixed time and routes. International Journal of Vehicular Technology 2014:1–8