Final Report

Developing Optimal Peer-to-Peer Ridesharing Strategies

Young-Jae Lee, Ph.D.
Professor, Dept. of Transportation and Urban Infrastructure Studies
Morgan State University, 1700 E. Cold Spring Ln, Baltimore, MD 21251, USA
Email: youngjae.lee@morgan.edu

Amirreza Nickkar, Ph.D.
Adjunct Faculty Lecturer, Dept. of Transportation and Urban Infrastructure Studies
Morgan State University, 1700 E Col Spring Lane, Baltimore, MD 21251
Email: amirreza.nickkar@morgan.edu

Date
August 1, 2023

Prepared for the Urban Mobility & Equity Center, Morgan State University, CBEIS 327, 1700 E Coldspring Ln, Baltimore, MD 212

ACKNOWLEDGMENT

This research was supported by the Urban Mobility & Equity Center at Morgan State University and the University Transportation Center(s) Program of the U.S. Department of Transportation.

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No. UMEC-022	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Developing Optimal Peer-to-Peer Ridesharing Strategies		**5. Report Date** August 2023 **6. Performing Organization Code**
7. Author(s) Young-Jae Lee, Ph.D. https://orcid.org/0000-0002-1422-7965 Amirezza Nickkar https://orcid.org/0000-0002-1242-3778		**8. Performing Organization Report No.**
9. Performing Organization Name and Address Morgan State University 1700 E. Cold Spring Lane. Baltimore, MD 21251		**10. Work Unit No.** **11. Contract or Grant No.** 69A43551747123
12. Sponsoring Agency Name and Address US Department of Transportation Office of the Secretary-Research UTC Program, RDT-30 1200 New Jersey Ave., SE Washington, DC 20590		**13. Type of Report and Period Covered** **14. Sponsoring Agency Code**

15. Supplementary Notes

16. Abstract

Thanks to recent developments in ride-hailing transit services, the Peer-to-Peer (P2P) ridematching problem has been actively considered in academia in recent years. P2P ride-matching not only reduces travel costs for riders but also benefits drivers by saving them money in exchange for their additional travel time and costs. However, assigning riders to drivers in an efficient way is a complex problem that requires a focus on maximizing the benefits for both riders and drivers. This study first aims to formulate a multi-driver multi-rider (MDMR) P2P ride-matching problem based on rational preferences and cost allocation for both driver and rider. This model also enables riders to transfer between multiple drivers to complete their journeys if needed. To solve the ride-matching problem, a Tabu Search (TS) for system optimum ride-matchings and Greedy Matching (GM) algorithm for the stable ride-matchings were created to produce stable ride-matchings.

17. Key Words:	**18. Distribution Statement** No restrictions.		
19. Security Classif. (of this report): Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages**	**22. Price**

Developing Optimal Peer-to-Peer Ridesharing Strategies

Young-Jae Lee, Ph.D.
Professor
Department of Transportation and Urban Infrastructure Studies
Morgan State University
1700 E. Cold Spring Lane, Baltimore, MD 21251
Tel: 443-885-1872; Fax: 443-885-8324
Email: YoungJae.Lee@morgan.edu
ORCiD: 0000-0002-1422-7965

Amirreza Nickkar, Ph.D.
Adjunct Faculty Lecturer
Department of Transportation and Urban Infrastructure Studies
Morgan State University, 1700 E Col Spring Lane, Baltimore, MD 21251
Email: amirreza.nickkar@morgan.edu
ORCiD: 0000-0002-1242-3778

ABSTRACT

Thanks to recent developments in ride-hailing transit services, the Peer-to-Peer (P2P) ride-matching problem has been actively considered in academia in recent years. P2P ride-matching not only reduces travel costs for riders but also benefits drivers by saving them money in exchange for their additional travel time and costs. However, assigning riders to drivers in an efficient way is a complex problem that requires a focus on maximizing the benefits for both riders and drivers.

This study first aims to formulate a multi-driver multi-rider (MDMR) P2P ride-matching problem based on rational preferences and cost allocation for both driver and rider. This model also enables riders to transfer between multiple drivers to complete their journeys if needed. To solve the ride-matching problem, a Tabu Search (TS) for system optimum ride-matchings and Greedy Matching (GM) algorithm for the stable ride-matchings were created to produce stable ride-matchings.

The results show that the developed algorithm could successfully solve the proposed P2P MDMR ride-matching problem. MDMR P2P ride-matching can be used in areas where not much demand for ridesharing demands is available or for a long-distance travel. It can also be applied to design for on a more efficient demand transit network design which can allow for transfers between routes. Moreover, the comparison of results between two implemented approaches shows that system optimum centralized ride-matching can bring more cost savings for all participants in the system, although it may not always be stable when riders and drivers can choose their ride-matching for their maximum benefit.

Using the algorithm that we developed for the P2P ride-matching problem, we also developed an algorithm for on-demand transit network which allows transfers. On-demand transit systems aim to improve mobility in the transportation network. Recent technological advancements in communications and intelligent transportation systems have provided many opportunities to make the conventional transit systems more effective and efficient. Although on-demand services have already shown better performance than traditional fixed-route transit services in reducing total travel costs in low demand areas, additional attributes may further increase the usability of on-demand systems.

This study aims to develop an algorithm for optimal on-demand network design by accommodating transfer points for users. The proposed optimization problem also considers time windows and simultaneous pickup/delivery as constraints. An example was developed and tested to demonstrate the algorithm in two different networks (with and without transfer points). The algorithm successfully assigned riders to the transit system while considering transfers points for users. A comparison between the two networks shows that considering transfer points can save up to 2.73% in total travel costs in the whole transit system.

I. DEVELOPING AN OPTIMAL PEER-TO-PEER RIDE-MATCHING PROBLEM ALGORITHM WITH RIDE TRANSFERS

I.1 INTRODUCTION

In recent years, ridesharing has become an important element of urban transportation. Thanks to emerging smart phones and mobile apps, it has become easy to connect travelers and drivers offering ridesharing through these on-demand transit services *[1]*. In the United States, major ride-hailing companies like Uber and Lyft have invested substantial funding toward improving shared services to travelers. For example, Uber, a ride-hailing company, has the ridesharing option called UberPool, which offers cheaper prices compared to ride-hailing services.

Although many studies have shown that shared ride-hailing services provide financial advantages and benefits to transportation networks over regular ride-hailing services, shared ride-hailing mobility services still have some issues that raise questions about their efficiency *[2]*. Shared ride-hailing mobility may also increase deadheading travels for the ride provider and consequently a longer travel time for ride requesters, which makes the system less efficient with higher vehicle-miles traveled (VMT) *[3]*. Recently, scholars have considered peer-to-peer ridesharing services to solve these issues.

A peer-to-peer ride-matching service (P2P RMS) is similar in many aspects to ride-hailing services; however, some fundamental differences make it more profitable in certain situations. Mainly, shared ride-hailing services are provided by transportation network companies (TNC); both the operator and ride provider should financially benefit from the service, and ride providers make trips just for the money without any other purpose. In P2P RMSs, ride providers have their own trips and need to be compensated for the additional travel to provide a service, so trip costs for the ride requester will be cheaper than the trip costs of the ride-hailing company *[4]*. Moreover, the stability of matching is an issue that is highlighted in P2P RMSs where drivers and riders can only be matched when there is a reasonable cost saving for them. The flexibility of the ridesharing system is a key factor that makes the system profitable for both drivers and riders, and in the current market, this flexibility is more easily achieved than at any time before, thanks to app-based platforms and developed algorithms *[5]*.

This study proposes a novel, stable P2P ride-matching system to optimize the multi-driver, multi-rider matching, and routing problem, allowing riders to transfer between drivers (vehicles) and share rides with other riders. It then develops optimal algorithms to solve the proposed problems based on Tabu Search (TS), and Greedy Matching (GM) approaches. This study also compares the performance of the model for stable and unstable matchings. The main motivation of this study is to have better insights toward practical comparison between stable and non-stable matching algorithms and also using transfer points to optimize matching drivers and riders (whether short-distance travel or long-distance) in the transportation network. Allowing riders to transfer between vehicles may increase the chance of having more possible feasible matches between participants, and also, it will reduce the cost by decreasing the total traveled distance in the system that will benefit both the rider and the driver. Furthermore, considering transfer points can be useful in areas with a low ridesharing demand and less access to public or paratransit systems. The results of this study could be utilized by TNCs and on-demand shared mobility service agencies. The organization of this study is as follows: The next section provides

a comprehensive review of background studies. The third section describes the proposed problem and two algorithms to solve the model. Section four discusses the developed algorithms' performance in a hypothetical example, and, finally, the last section concludes this research.

I.2 LITERATURE REVIEW

With the emergence of shared transit modes in urban areas, scholars have studied the ride-matching problem (RMP) widely in recent decades. The RMP is basically a combinatorial optimization problem that aims to find matches between drivers who are willing to provide rides and riders who need rides by considering feasibility and profitability constraints. RMPs are mainly associated with transit modes that can deliver and pick up passengers simultaneously by request; therefore, RMPs are often considered as demand responsive and dial-a-ride problem (DARP) studies.

In the past few decades, DARPs have been widely considered by scholars *[6-8]*. Generally, the homogeneity of the variables and the status of passengers' request orders are the main challenges that motivate scholars to optimize DARPs *[9]*. The homogeneity of the problem is related to adding some variables that make the model more realistic, like multi-hubs *[10]*, the various capacity of the vehicles, transferring of passengers *[11]*, and the degree of circuity (DOC) *[12, 13]*. Alternatively, the system's flexibility in serving passengers was introduced by transforming the problem from static to dynamic, so the transit system can accept a new request while operating *[14]*.

In DARPs, ride providers are considered as a separate entity as an operator, while in P2P RMP the ride operator can be considered as a user of the systems *[15]*. Furthermore, with DARPs, the availability of vehicles and ride requests are known in advance, while in a P2P RMP, this availability is not deterministic as riders would be matched to drivers by considering their spatial-temporal readiness and the feasibility of matching. This special attribute makes the P2P RMP more complicated than DARP, especially when some other constraints should be considered, such as allocating riders with special needs to specific ride providers *[16]* or considering the time window constraint for each rider *[17]*. Stiglic et al. *[18]* introduced the concept of meeting points in the ride-matching problem. An optimal algorithm has been developed to maximize the number of matched drivers and riders to save drivers' driving distance. The concept of the transfer station, applied by Masoud and Jayakrishnan *[16]*, can make the ride-matching system more efficient by increasing the chance of matching between more riders and drivers. The transfer stations work as given spots where riders would be switched to another driver to continue their trips. The transfer stations are not necessarily located at drivers' starting or end points; however, serving riders at transfer stations could be more acceptable to drivers as it makes for less complexity and confusion *[16]*.

Generally, in past related studies, the ride-matching problems in terms of complexity have been categorized as Single-Driver Single-Rider (SDSR), Single-Driver Multiple-Riders (SDMR), and Multiple-Driver Multiple-Riders (MDMR). Agatz et al. *[19]* investigated a single driver and single rider P2P ride-matching problem to optimize unmatched announcements aiming to maximize the saved traveled distance of the drivers. They believed the performance of driver-rider matching is highly dependent on the spatial-temporal status of the ride requesters. Boyacı et al. *[20]* solved a one-way vehicle sharing problem using a simulation approach aimed at maximizing the net revenue for the operator and net benefits for users.

Considering the ability of riders to transfer between drivers makes the problem more complicated. A few scholars have considered the MDMR ride-matching problem. Masoud and Jayakrishnan *[16]* developed a multi-hop P2P ridesharing system as a binary optimization problem in which a rider was able to transfer between multiple drivers. They used a decomposition algorithm to solve the model optimally. In another work by these authors *[15]*, they developed a flexible ridesharing system as a dynamic, real-time, and multi-hop system with the ability to find itineraries for a rider by means of optimally routing drivers. Following this study, Tafreshian and Masoud *[21]* proposed a one-to-many ride-matching problem based on a graph partitioning algorithm that grouped travels into several sub-problems.

Recently, some studies focused on rational aspects of ridesharing participants (both ride provider and ride requester). Preferences considering saving money for riders and drivers are the main focus of these studies. Silva et al. *[22]* considered a Quota Travelling Salesman Problem with Passengers (QTSP) to solve the ride-matching problem in order to increase the flexibility of the model for serving riders in meeting points (alighting and boarding). Although their model aimed to increase satisfaction for both riders and the driver, the proposed model neglected the match stability. Wang et al. *[23]* developed a stable ride-matching model to fairly distribute cost allocation between driver and rider; however, they considered an average cost-savings between driver and rider, which is unfair in some matches. Furthermore, their proposed model was an SDSR problem that is limited to realistic and substantial service. Ma et al. *[24]* modified the algorithm by introducing the theory of two-sided matching to solve the ridesharing stable matching problem by considering cases in which a driver may receive identical benefits for a match with a rider, which turned the problem into a stable matching problem with incomplete preferences. Also, their developed model was an SDMR problem which is more useful than SDSR.

Since the P2P ride-matching problem was recognized as an NP-hard problem *[25]*, most of the past studies used heuristics and metaheuristics to solve the problem; however, some other studies focused on the goal and scale of the problem using a simulation-based approach *[25-29]*. The objective functions of the previous related researches included maximizing the number of rider and driver matches *[17, 18, 27, 30-33]*, maximizing the total number of assignments *[34-36]*, optimizing system costs and benefits *[17, 20, 33, 37]*, travelers' mode choice *[38-40]*, and optimizing total distance savings *[19, 25, 27]*.

Although various types of the P2P ride-matching problem have received considerable attention in recent years, little knowledge is available about the stable MDMR problem allowing transferring riders between multiple vehicles. Only one study by Peng et al. *[36]* considered a stable MDMR problem to optimize matching assignment and pricing; however, their model was unable to permit riders to transfer between multiple drivers. Therefore, this study has two main contributions rather than the reviewed studies. First, it proposes a novel 2-stage integer programming model for the MDMR problem according to the SDMR matching problem's extension. Second, it develops an optimal algorithm to solve the stable MDMR problem considering rational preferences and cost allocation for drivers and riders while allowing a rider to transfer between multiple drivers. The other contribution of this study is to solve the proposed model for both stable and non-stable forms and compare them in terms of total cost savings.

I.3 METHODOLOGY

As discussed in the literature review section, two main approaches for the ride-matching strategies are stable matching and non-stable matching. The stable matching system is a decentralized system, while a non-stable matching system is a centralized system. Although the stable matching approach provides a more stable solution, it may not provide the most efficient peer-to-peer ride-matching for the entire system because of the smaller total number of matches due to the only rational matches for both the rider and the driver. Therefore, in general, non-stable matching provides more matches in the system, while this may not be individually optimal for both drivers and riders in some circumstances. Figure 1 shows the simple example network with two drivers (D1 and D2) and two riders (R1 and R2). Assume that the cost of trips is commensurate with the shortest distance traveled to reach the destinations. In Figure 1, the vertices represent the location related to origin (shown with "O") or destination (shown with "D") of the rider (R)/driver (D) and the arrows show the connection between locations, and the value on the arrows indicates the cost of a ride between the two vertices which is directly related to distance between the two locations. Without ridesharing, each rider/driver rides from its origin to its destination (with routs, D_1^O-D_1^D, D_2^O-D_2^D, R_1^O-R_1^D, R_2^O-R_2^D), consequently, each trip costs \$6 (total system cost is \$24). However, with ride-matching, if R1 and D1, as well as R2 and D2 are matched, for system optimization, the cost per match is \$10 (with route D^O-R^O-R^D-D^D, accordingly cost of \$2+\$6+\$2) and total system cost becomes \$20 (\$10+\$10), which lowers \$4 from the individual trips.

Although the system optimal ridesharing solution ensures the system cost minimization, those ride-matchings are not stable because D1 and R2 ride-matching lowers their travel costs from \$10 to \$9.2 (with route D_1^O-R_2^O-R_2^D-D_1^D, accordingly cost of \$1.6+\$6+\$1.6), and they will not accept the optimal system ride-matching. Considering the ride-matching of D1 and R2, it is optimal for R1 and D2 to ride individually with a total cost of \$12 rather than to have ride-matching. Thus, in this case, the total system cost increases to \$21.2 (\$6+\$6+\$9.2) from the system optimal solution of \$20. This ride-matching that makes the individual cost minimization is called stable ride-matching. It is a more realistic ride-matching solution when the individuals choose their ride-matching.

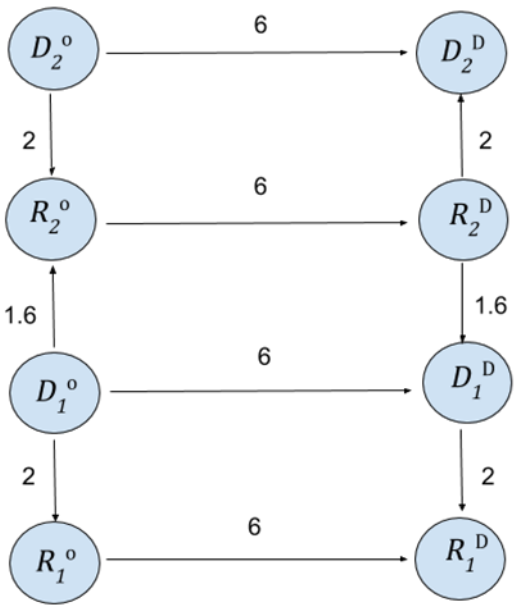

Figure 1 Example of Ridesharing Network

I.3.1 Problem

In this paper, a many-to-many ridesharing system is considered in which the centralized system matches groups of riders and drivers and offers itineraries to drivers. The system begins its work when a set of participants requests trips as drivers or riders. Each participant selects an origin and a destination within an allowable time window. All origins and destinations of drivers and riders are considered as potential transfer stations. A transfer station is a meeting location where a rider can be transferred to another driver in order to complete his/her trip. Furthermore, riders can choose their maximum allowed transfers between vehicles, and drivers can decide the maximum number of passengers on board at the same time during their entire trip. Therefore, each time a rider or a driver applies for a trip in the system, the information about these preferences is asked and registered.

The model should be able to consider a case in which a driver can provide a ride to a rider from their origin to their destination and from their origin to their transfer point. We assumed that the participants cooperated with the system for the financial efficiency of the service as well as for self-interest goals like the effect on the environment. Hence, the objective of the problem is considered to maximize the total travel distance savings of all participants, which also corresponds to minimizing total travel costs. In this problem, a rider can take all of his/her trip with one driver or can be switched to another driver in the middle of his/her trip at a transfer station. According to minimizing total travel costs, the mathematical model decides whether a rider should be switched to another driver in the middle of his/her trip. In this regard, the model considers all possible cases of matchings (a set with n drivers and m riders that has at most $2n \times 2m$ possible matchings) where the number of participants in the match is less than a predefined value and calculates the optimal costs of these matching; therefore, the model chooses the best case by comparing the total costs of each matching.

In many-to-many ridesharing systems, riders may be assigned to multiple drivers, transferring between multiple vehicles. Therefore, the system sets the location where each driver picks up and drops off the rider. When the system matches a set of riders with a set of drivers, it means that the trip timing of all participants becomes coordinated as itineraries. Matches and trips offered by the system should satisfy the participants, although they may choose not to accept the offer and instead decline using the system. Two important aspects of participant satisfaction are the timing and cost of their trip. We assume that matches and itineraries are feasible only if they fulfill the timing requirements of all participants as a constraint. There may be ridesharing matches that satisfy the participants' timing constraints but do not generate positive vehicle-mile savings. Indeed, to consider a match as feasible, the involved participants should also benefit with respect to the travel cost when accepting the offer. As mentioned earlier, when the system matches a set of riders with a set of drivers, the trip timing of all participants becomes coordinated as the itineraries. In cases of participant delay, the dependence of the time scheduling of participants for the match to each other may result in delays experienced by other participants grouped in a match. To overcome this issue, the maximum number of participants in each match is restricted. When a set of drivers is matched with a set of riders, an optimization problem is solved to determine each of the participants' itinerary and the start and end time of the trip. We assume that the vehicle-mile savings (cost savings or benefits) of sharing a ride are divided equally between the ride's participants. Another option is to divide the cost-savings of a ride among the participants in the match according to each participant's trip distance when driving alone.

A good ridesharing system should produce matches that are stable; we define stability as the concept noted in cooperative game theory. In fact, we call a solution stable when no groups of riders and drivers prefer to depart from their current match. This also includes the case in which a subset of riders and/or drivers prefer to reject some of their current partners. Riders and/or drivers may not use the system if they believe they can establish a better match with higher cost saving on their own. The following are the known parameters of the problem.

Parameters:

P: The set of participants in the system including riders and drivers

R: The set of riders

D: The set of drivers

TS: The set of meeting points/transfer stations, where participants can start and end their trips, and riders can change vehicles in these points

O_p: The transfer station where participant $p \in P$ starts his/her trip.

D_p: Destination of the participant $p \in P$

tt_{mn}: Travel time between each pair of locations $m, n \in TS$

dis_{mn}: Distance between each pair of locations $m, n \in TS$

ed_p: The earliest time participant $p \in P$ can start her/his trip in her origin O_p

la_p: The latest time when participant $p \in P$ must arrive at his/her destination D_p

f_r: The maximum number of transfers between vehicles acceptable for rider $r \in R$

Ca_d: Maximum number of passengers driver $d \in D$ can carry in his/her vehicle

Let A and B represent the set of all non-empty subsets of R and D, respectively. We introduce the set A_r as the set of members of A, which include $r \in R$ and B_d as the set of members of B, which includes $d \in D$. In this way, $A = \bigcup_{r \in R} A_r$ and $= \bigcup_{d \in D} B_d$.

To model the problem, the preference lists of all groups of participants should be known. The preference list of a group of riders/drivers consists of the set of feasible matches ranked based on the corresponding best ride-share cost savings. The best share-ride with the maximum cost-savings between any two groups of drivers and riders and the associated cost-savings can be determined through the model (11)-(30) in the next subsection. The solution of the model also determines whether a match is feasible or not.

We denote the preferences with the following notation: a $>_c$ b denotes that person c prefers person a to b, and a $\geq_c$ b denotes that either a $>_c$ b or person c is indifferent to person a or person b. If the cost-savings of two share-rides for a group of drivers/riders is the same, the riders/drivers are indifferent to the two matches. Thus, preference lists may contain ties. Moreover, some matches may be infeasible, so the preference list may be incomplete as well.

By defining sets A and B, the problem can be modeled as a generalized matching problem between members of sets A and B. Whenever a match is established between two groups of riders and drivers, the members of the two sets share rides together and the trip timing of all participants becomes correlated with their itineraries. To the best knowledge of the current authors, the MDMR ridesharing problem has not yet been formulated as a matching problem, unlike SDMR. Let x_{ab}, a binary decision variable, be equal to 1 if: a ride-share match between riders of the set $a \in A$ and drivers of the set $b \in B$ is established and 0. Otherwise, let g_{ab} represent the corresponding cost-savings. Moreover, Let $B(a)$ be the set of all members b of the set B where matching a with b is feasible and let $A(b)$ be the set of all members a of the set A where matching a with b is feasible. As mentioned earlier, the feasible matches and the parameter g_{ab} are determined by solving another model (equations 11 to 30), which is described later in the document.

$$Max \sum_{a \in A} \sum_{b \in B(a)} g_{ab} x_{ab} \quad (1)$$

s.t.

$$\sum_{a \in A_r} \sum_{b \in B(a)} x_{ab} \leq 1 \qquad \forall r \in R \quad (2)$$

$$\sum_{b \in B_d} \sum_{a \in A(b)} x_{ab} \leq 1 \qquad \forall d \in D \quad (3)$$

$$x_{ab} + \sum_{a' \neq a \subset a} \sum_{b' \in B: b' \succcurlyeq_{a'} b} x_{a'b'} + \sum_{b' \neq b \subset b} \sum_{a' \in A: a' \succcurlyeq_{b'} a} x_{a'b'} \geq 1; \forall a \in A; \forall b \in B(a) \quad (4)$$

$$x_{ab} \in \{0; 1\} \qquad \forall a \in A; \forall b \in B \quad (5)$$

The objective (1) maximizes the total cost-savings of the system, which coincides with minimizing the total travel cost. Constraint sets (2) and (3) force each rider (driver) to be matched with at most one group of drivers (riders). Constraint (4) forces the model to generate a stable solution. These constraints state that each pair of riders (set $a \in A$) and drivers (set $b \in B$), matched or a subset of the members of the set $a \cup b$, prefers their current match to this one.

In fact, at least one of the three outcomes below must happen for each pair of riders a ∈ A and drivers b ∈ B:

- "a" is matched with "b" and therefore $x_{ab} = 1$.
- Members of the set a are not united for pairing with set b. Therefore, a proper subset $a' \neq a \subset a$ exists where a' is paired with $b' \in B$ ie. $x_{a'b'} = 1$, and the members of set a' prefer their current match to pairing a with b, ie. $b' \succcurlyeq_{a'} b$ and $\frac{g_{a'b'}}{|a'|+|b'|} > \frac{g_{ab}}{|a|+|b|}$.
- Members of the set b are not united for pairing with set a. Therefore, a proper subset $b' \neq b \subset b$ exists where b' is paired with $a' \in A$ ie. $x_{a'b'} = 1$, and the members of set b' prefer their current match to pairing a with b, ie. $b' \succcurlyeq_{a'} b$ and $\frac{g_{a'b'}}{|a'|+|b'|} > \frac{g_{ab}}{|a|+|b|}$.

To be detailed, the stability constraints (4) can be replaced with the following constraints, where z_p is the cost-savingS of participant $p \in P$ in the current match and q_p^{ab} is a binary variable that if equal to one, the participant p saves less when a is matched with b than its cost-saving in the current match. Furthermore, the set a is matched with the set b if the binary variable q_0^{ab}equals to one.

$$z_r = \sum_{a \in A_r} \sum_{b \in B(a)} \frac{g_{ab} x_{ab}}{|a|+|b|} \qquad \forall r \in R \qquad (6)$$

$$z_d = \sum_{b \in B_d} \sum_{a \in A(b)} \frac{g_{ab} x_{ab}}{|a|+|b|} \qquad \forall d \in D \qquad (7)$$

$$x_{ab} \geq q_0^{ab} \qquad \forall a \in A, \forall b \in B(a) \qquad (8)$$

$$z_p \geq q_p^{ab} \times \frac{g_{ab}}{|a|+|b|} \qquad \forall a \in A, \forall b \in B(a); \forall p \in a \cup b \qquad (9)$$

$$q_0^{ab} + \sum_{\forall a' \subset a, b' \subset B(a')} \prod_{\forall p \in a' \cup b'} q_p^{ab} + \sum_{\forall b' \subset b, a' \subset A(b')} \prod_{\forall p \in a' \cup b'} q_p^{ab} \geq 1 \quad \forall a \in A, \forall b \in B(a) \qquad (10)$$

Constraints (6) and (7) define z_p as the cost-savings of participant p in the current match. Constraint set (8) forces the set of drivers a to be matched with the set of riders b if q_0^{ab} equals one. Constraint set (9) forces the variable q_p^{ab} to be equal to zero if the cost-savings of participant p is increased when the set of drivers a is matched to the set of riders b. Constraint (10) forces each set of riders a and set of drivers b to be matched or at least two sets $a'$and b' exist where $a' \subset a, b' \subset B(a')$ or $b' \subset b, a' \subset A(b')$, and for all members of the sets a' and b', the cost-saving is decreased if the match between sets a and b happens.

I.3.2 Feasibility and Profitability Conditions of Ride-matching

When a set of drivers *b* is matched with a set of riders *a*, the problem of finding the best itinerary for each driver is modeled by restricting the set of drivers and riders to the members included in sets *a* and *b* denoted by SD and SR and by using four sets of decision variables as defined below. Moreover, the set of stations is restricted to the ones included in the largest polyhedral generated by connecting the riders and drivers in the match. In fact, the model (11)-(30) inputs a subset of stations $S \subseteq TS$, a subset of riders $SR \subseteq R$, and a subset of drivers $SD \subseteq D$ as a matched group and finds the best itinerary for each driver in SD such that all riders and drivers start their trips from their corresponding origins and end in their destinations in the

allowed timing while the total cost-savings is maximized. Therefore, the model assigns drivers to each rider and determines the stations where each rider should get on or off.

In the proposed formulation, an equal number of stops is considered for each driver denoted by K_{max}, and we allow the model to repeat the destination of the driver at the end of their itinerary to consider all possible solutions to the problem. It is assumed that a driver, unlike a rider, may visit any transfer station more than once in their itinerary. Besides, although this is not a usual case, it was assumed a rider may change the vehicle but return to the same one later.

v_{mnk}^{d} Driver d travels from m to n and m is the k-th stop of the driver d, $n \neq m \in S$

y_{mnk}^{rd} Rider r travels with driver d from transfer station m to n when m is the k-th stop of the driver d, $n, m \in S$

z_{k}^{rd} Rider r starts a partial path with driver d in the driver's k-th stop

t_{k}^{d} Trip starting time of driver d at the k-th stop in the driver's itinerary, the service starting time at the first point is just the earliest departure time of driver d.

Then the formulation is as follows:

$$\text{Min} \sum_{d\in SD}\sum_{m\in S}\sum_{n\in S}\sum_{k} v_{mnk}^{d} \times dis_{mn} \quad (11)$$

s.t.

$$t_{k+1}^{d} \geq t_{k}^{d} + \sum_{m,n} tt_{mn} v_{mnk}^{d} \qquad \forall d \in SD; K_{max} \geq k \geq 1 \quad (12)$$

$$t_{k}^{d} + tt_{mD_r} - M(1 - y_{mD_r k}^{rd}) \leq la_r \qquad \forall d \in SD; \forall r \in SR; \forall m \in S; K_{max} \geq k \geq 1 \quad (13)$$

$$t_{k}^{d} \geq ed_r - M(1 - \sum_{m} y_{O_r mk}^{rd}) \qquad \forall d \in SD; \forall r \in SR; K_{max} \geq k \geq 1 \quad (14)$$

$$t_{K_{max}}^{d} \leq la_d \qquad \forall d \in SD \quad (15)$$

$$t_{1}^{d} \geq ed_d \qquad \forall d \in SD \quad (16)$$

$$\sum_{n} v_{mn,k+1}^{d} = \sum_{n} v_{nm,k}^{d} \qquad \forall d \in SD; \forall m \neq O_d, m \neq D_d \in S; K_{max} - 2 \geq k \geq 1 \quad (17)$$

$$\sum_{n} v_{O_d n,k+1}^{d} = \sum_{n} v_{nO_d,k}^{d} \qquad \forall d \in SD; K_{max} - 2 \geq k \geq 2 \quad (18)$$

$$\sum_{n} v_{D_d n,k+1}^{d} = \sum_{n} v_{nD_d,k}^{d} \qquad \forall d \in SD; K_{max} - 3 \geq k \geq 1 \quad (19)$$

$$\sum_{n} v_{nD_d,K_{max}}^{d} = 1 \qquad \forall d \in SD \quad (20)$$

$$\sum_{n} v_{O_d n,1}^{d} = 1 \qquad \forall d \in SD \quad (21)$$

$$\sum_{d\in SD}\sum_{n}\sum_{k} y_{mn,k}^{rd} = \sum_{d\in SD}\sum_{n}\sum_{k} y_{nm,k}^{rd} \leq 1 \qquad \forall r \in SR; \forall m \neq O_r, m \neq D_r \in S \quad (22)$$

$$\sum_{d\in SD}\sum_{k}\sum_{n} y_{O_r n,k}^{rd} = 1 \qquad \forall r \in SR \quad (23)$$

$$\sum_{d\in SD}\sum_{m}\sum_{k} y_{m,D_r,k}^{rd} = 1 \qquad \forall r \in SR \quad (24)$$

$$\sum_{r\in SR} y_{mn,k}^{rd} \leq Ca_d \times v_{mn,k}^{d} \qquad \forall d \in SD; \forall n, m \in S; K_{max} - 1 \geq k \geq 1 \quad (25)$$

$$z_{k}^{rd} \geq \sum_{m}\sum_{n} y_{mn,k+1}^{rd} - \sum_{m}\sum_{n} y_{mn,k}^{rd} \qquad \forall r \in SR, \forall d \in SD, K_{max} - 2 \geq k \geq 1 \quad (26)$$

$$\sum_{d\in SD}\sum_{k} z_k^{rd} - 1 \le f_r \qquad \forall r \in SR \qquad (27)$$

$$\sum_{d\in SD}\sum_{k}\sum_{m} t_k^d \times y_{smk}^{rd} \ge \sum_{d\in SD}\sum_{k}\sum_{m}\left(t_k^d + tt_{ms}\right) \times y_{msk}^{rd} \qquad \forall r \in SR, \forall s \ne O_r, s \ne D_r \in S \qquad (28)$$

$$v_{mnk}^d, y_{mnk}^{rd}, z_k^{rd} \in \{0,1\} \qquad \forall r \in SR; \forall d \in SD; \forall n,m \in S; K_{max} - 1 \ge k \ge 1 \qquad (29)$$

$$t_k^d \ge 0 \qquad \forall d \in SD; K_{max} \ge k \ge 1 \qquad (30)$$

Equation (11) presents the objective function of the problem, minimizing the total distance traveled by all drivers. Constraints (12)-(16) force the model to satisfy the rider's and driver's latest arrival and earliest departure times. Constraint sets (17) to (19) enforce that the number of times a driver enters a transfer station equals the number of times they leave the transfer station. Constraint set (20) directs the drivers out of their original transfer stations, and constraint (21) ensures that drivers end their trips at their destination transfer stations. Constraint sets (22)-(24) route riders in the network. Constraint set (25) ensures that vehicle capacities are not exceeded and ensures that riders are accompanied by drivers throughout their trips. Constraints (26) and (27) limit the total number of transfers between vehicles for each rider. Constraint (28) ensures that a rider leaves a transfer station after they have arrived at it.

Then, g_{ab} can be determined simply using equation (31) and the solution of the model (11) to (30) for the feasible match between sets a and b. Note, that,

$$g_{a.b} = g_{SD.SR} = \left(\sum_{d\in SD} dis_{O_dD_d} + \sum_r dis_{O_rD_r} - \sum_{d\in SD}\sum_{m\in S}\sum_{n\in S}\sum_k v_{mnk}^d \times dis_{mn}\right) \qquad (31)$$

The designed problem may have unmatched riders and drivers if the timing constraint is not satisfied. In this case, the match between the set of riders and the set of drivers is considered as infeasible. Moreover, as mentioned earlier, there may be ride-share matches that satisfy the participants' timing constraints but do not generate positive vehicle-mile savings, which are also considered infeasible matches. Thus, preference lists can be incomplete. Furthermore, when a set of riders is matched with a set of drivers, one does not expect the solution to generate disjoint itineraries for a subset of riders. However, in the optimal solution of presented formulations (11) to (31), disjointed itineraries can be generated and the output of these formulations is used as an input for formulations (1) to (4). For example, let D={d1,d2,d3}t and R={r1,r2,r3}. The solution may result with drivers d1 and d2 routing riders r1 and r2 to their destinations and driver 3 accompanying rider 3 throughout his/her trip. In this circumstance, the drivers routes are alike, r3 is matched with d3 and the set {r1,r2} is matched with the set {d1,d2}. Indeed, the disjointed itinerary is infeasible for formulations (1) to (4). Thus, this solution is unstable and will not be chosen in the matching problem. This is because the trip cost associated with one of the disjointed sets can be increased in comparison of matching r3 with d3 and the set {r1,r2} with the set {d1,d2}. Therefore, the match will be considered infeasible in the second phase of the solution process. In fact, the solution process consists of two stages. In the first stage, parameter g_{ab} should be determined by applying formulation (11)-(30) for any set of riders a and drivers b. The preference lists of all groups of participants are also obtained. In the second phase, the output formulations (1) to (5) defines the optimal solution of the stable ride-matching problem.

Model (11)-(30) is an NP-hard problem and solving the model within a reasonable time for even medium sized problems is difficult. However, as mentioned before in the document, the model (11)-(30) inputs a subset of riders SR and a subset of drivers SD as a matched group and finds the best itinerary for each driver in SD such that all riders and drivers start their trips from their corresponding origins and end in their destinations in the allowed timing while the total cost-savings of the trips are maximized. To overcome the issue caused by matched participants being dependent on one another's schedules, the maximum number of participants in each match is restricted. Thus, one expects the set SD and SR which are the inputs of the model to be small sets with cardinality generally less than three.

I.3.3 Optimization methods

Two algorithms of TS and GM were selected to solve the proposed problem. The GM algorithm is useful when matched samples with similar balanced characteristics are generated. The GM considers one-to-one and one-to-many matched pairs, while it does not allow for sampling with replacement *[41]*. This GM algorithm has been widely used in ridesharing studies to compute maximum matchings *[42-45]*. The TS metaheuristics is known for its speed and ability to escape from local minima; however, the main advantage of TS which makes this algorithm suitable for ride-matching rests in its ability to set moving distance to search for the optimal solution, while the algorithm can automatically adjust its parameters and change the direction of searches. The TS metaheuristics have been implemented to solve the ridesharing/ride-matching problem thanks to its ability to set moving distance as an integer while searching *[38, 46]*. In the following, a GM heuristic is presented to solve the equations (1)-(5).

I.3.3.1 Greedy Matching (GM) Algorithm

1. Select the match (a, b) with the largest cost-savings per participant. Note that the cost-savings per participant of a match between the set *a* with the set *b* is determined by $\frac{g_{ab}}{|a|+|b|}$.
2. Remove the members of the sets a and b from the set of riders and drivers, respectively. Moreover, for each $r \in a$ and $d \in b$, remove the members of the sets A_r and B_d from A and B, respectively.
3. Repeat step 1 and 2 until no feasible matches remain.

Theorem 1. The GM algorithm generates a stable matching solution.

Proof:

Let the algorithm solution be a Greedy Matching solution that is not stable. By definition, there must then exist a pair (a, b) for which $\frac{g_{ab}}{|a|+|b|} > z_p; p \in a \cup b$. This is a contradiction because if this was true the match (a, b) would have been selected before the other matches were considered.

Corollary 1 of theorem 1. If riders and drivers are never indifferent between any two possible matches, the GM algorithm generates the best stable solution with the maximum cost-savings for the system.

Pf. (by contradiction)

Let (a,b) be the first match selected in the GM solving procedure for which the set of riders *a* or drivers *b* are not united to match in the maximum cost saving solution, eg., the set a is not matched with set b because, $a' \subset a$ (or $b' \subset b$) exists that are already matched with other participants and members of a'(or $b' \subset b$), prefer their current match to matching a with b. Without loss of generality, consider that the set of riders a are not united. Let $(a'_1, b''_1), (a'_2, b''_2), \dots, (a'_k, b''_k)$, $k \geq 1$ be the match of members of a'. In this way, $\frac{g(a'_n,b''_n)}{|a'_n|+|b''_n|} > \frac{g(a,b)}{|a|+|b|}$; for $1 \leq n \leq K$ and each a'_n prefers b''_n to match (a,b).

Note, that the set of riders a are the first which are not united. Therefore, participants which are matched with set a' that are matched already, eg., b''_n prefer their current match a'_n, to their match in the maximum cost saving solution. Thus, both sides of the match (a'_n, b''_n), prefer the match to match (a,b) and (a,b) is unstable as a maximum cost savings solution. ▪

If riders and drivers are never indifferent between any two possible matches, the GM algorithm generates the best stable solution with the maximum cost-saving for the system. However, in the presence of indifferences, randomly selecting the pair with the maximum cost-saving per participant may lead to the loss of optimality. In this circumstance, all possible outcomes of the GM algorithm should be checked, and the one with the maximum objective value is the best solution. Although for most cases the total number of this outcome is not large, finding the best stable matching using the GM algorithm is possible.

I.3.3.2 Tabu Search (TS) Algorithm

To calculate the effect of the selfish behaviors of the participants, the equations (1)-(3) and (5) are solved by employing a TS algorithm. The TS algorithm starts with a potential solution to the problem generated by using two random permutations of riders and drivers, where a rider and a driver with the same index in the two permutations are matched. This solution is considered as the initial bestMatch. Next, in each iteration of the algorithm, neighbors of the bestMatch are created in the hope of finding an improved solution. The algorithm considers solutions as neighbors if one driver and one rider, at most, are separated from their previous matches. The disjointed rider and driver may be matched or may join independent itineraries. Furthermore, there is a possibility for both riders to not match with anyone in the neighbor solution and continue their trips individually. The disjointed rider and driver and their new positions should be selected so that the neighbor solution satisfies feasibility. For each new solution, the objective function is calculated, and the algorithm moves to the next iteration by selecting the neighbor with the most improvement in the objective value as the new bestMatch solution. The iteration continues until it reaches MaxIteration. In order to determine the new objective value, there is no need to calculate the cost-savings for all matches as the algorithm investigates the differences of the previous solution and its neighbor. Moreover, a match score is determined between any two pairs of the participants initialized to zero. In each iteration of the algorithm this score is updated as neighbors are created. The difference of the objective value of the neighbor solution and the current bestMatch is then calculated. If this value is greater

(smaller) than zero, one is added (reduced) to the match score of the new fellow travelers of the disjointed rider and driver and one is reduced (added) from their previous co-travelers. The match score of any two participants falls under a specific value called the MinimumMatchScore; the two are added to the tabu list and cannot be co-travelers for the next candidate solutions, in a particular number of iterations. Then the match score of the pair is set to zero again. Furthermore, the algorithm allows the tabu list to include at most a constant number of tabu pairs represented by TabuTenure. Each time a new tabu pair is added to the tabu list and makes the list's length greater than TabuTenure, the oldest pair in the tabu list gets removed. Finally, the whole TS algorithm, from generating a random initial solution to reaching the MaxIteration iterations, is repeated a specific number of times denoted by MaxAlgorithmRepeat and the best match among these runs of the algorithm is selected as the best solution. Figure 2 shows the developed TS algorithm to solve the proposed P2P ride-matching problem. The platform used to program the algorithms is MATLAB 2019a.

```
tabuList = []
currentSolution ← initialSolution
bestSolution ←s
while ObjectiveValue(bestSolution) < 0 do
    bestMatch ←null
    for candidate ∈currentSolution.getNeighbourhood do
        If (¬tabuList.contains(candidate)) then
            If (ObjectiveValue(candidate) > ObjectiveValue(bestMatch)) then
                bestMatch ←candidate
            End If
        Else If (ObjectiveValue(candidate) > ObjectiveValue(bestSolution)) then
            bestMatch ←candidate
        End If
    End for
    currentSolution ← bestMatch
    If ObjectiveValue(bestMatch) > ObjectiveValue(bestSolution) then
        bestSolution ←bestMatch
    End If
    tabuList.push(bestMatch)
    If tabuList.size > tabuTenure Then
        tabuList.removeFirst()
    End If
End while
Return bestSolution
```

Figure 2 Developed TS algorithm to solve the problem

I.4 EXAMPLE

The proposed model was applied to a numerical experiment in a hypothetical network to demonstrate its performance in solving the problem. The experiment considered 10 riders and 10 drivers in the problem instance. Transfer station locations and participant origins and destinations (which are selected from the transfer stations) were generated randomly. The algorithms were coded in MATLAB 2019a and all the models were performed on a computer with CPU Intel® Core(TM) i5-7400 3GHz 16 GB of RAM memory. The calculation times

required to perform the GM and the TS algorithms for the hypothetical example were 34.2 and 45.1 seconds respectively.

In Figure 3, origins and destinations of the participants are shown, respectively. To differentiate the drivers as well as the riders, each has a unique number shown in the Figure 3. For example, D2 represents the driver number 2.

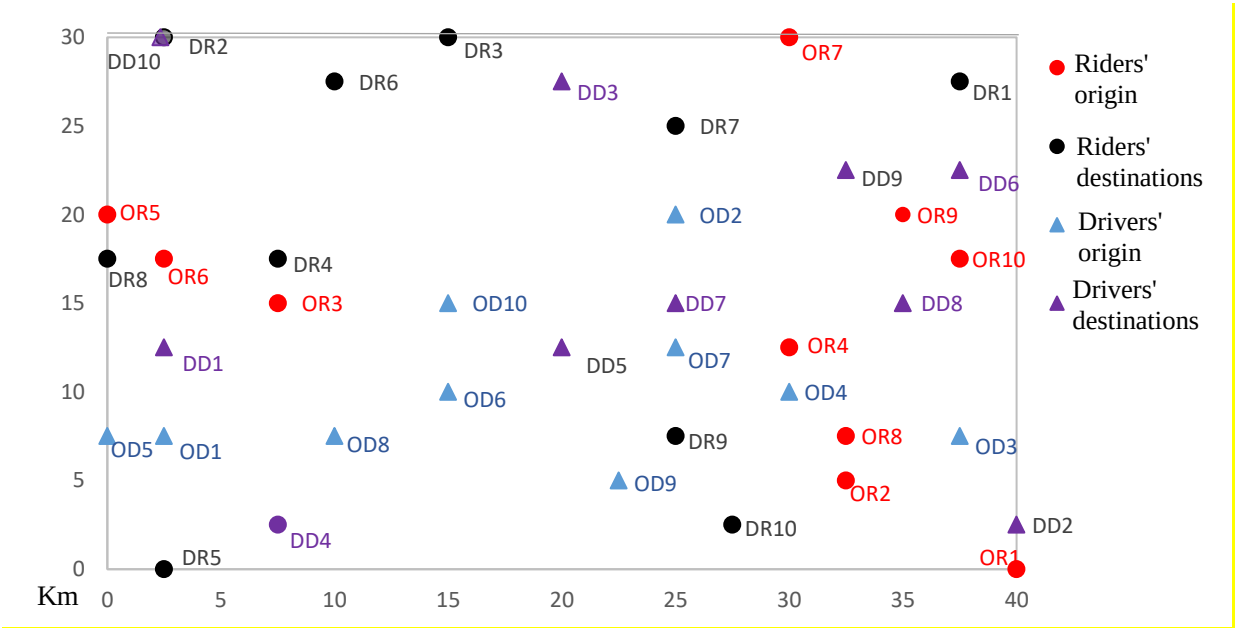

Figure 3 Destinations and origins of the participants in the generated problem instance

Next, preference lists were created for all groups of riders and drivers. To create the preference lists, the cost-savings of the matches were determined. Moreover, in order to make the computations faster, for each rider and a pairs of two drivers, the algorithm determines the best interface transfer station for the rider to move between the two drivers. In this problem, the total number of transfers between vehicles for each rider is limited to two transfers.

The TS and the GM heuristic algorithms explained earlier in the document were used to solve the generated instance. The parameters were set as following. $f_r = 2, K_{max} = 5$, MaxIteration=5000, MaxAlgorithmRepeat=10, MinimumMatchScore=-10, TabuTenure =30.

As mentioned before, the optimal stable solution can be obtained using the GM algorithm; however the TS algorithm produces an unstable solution. The difference of the objective value of the solution generated using the two methods represents the price of anarchy which shows the outcome of the selfish behavior of the participants in a practical ridesharing setting.

The conclusions of the GM and the TS algorithms for the total cost-saving of the ride sharing system is determined as 7969.3 (meters) and 7604.8 (meters), respectively. In this sense, the Price of Anarchy is 364.5 (meters) which is a significant value in contrast to the cost-savings. In fact, the total ridesharing system cost-saving can be increased by 4.8% if based on satisfaction level of the riders. Table 1 represents the results of algorithms for the proposed example. The total cost per kilometer has been considered 35 cents according to the last travel reimbursement fees issued by the IRS *[47]* and also the monetary saving calculation has been considered based on dollar per kilometer.

As shown in Table 1, the total saving of the TS algorithm is higher than the GM algorithm because the GM algorithm provides an exact solution through a stable model while the TS algorithm is a metaheuristic approach and finds near to system optimal solution. Therefore, the TS model cannot be considered stable, but it opens doors to compromise between riders and drivers most likely to maximize the number of matching and/or the shared travel distances, which will bring more cost savings in the system.

Table 1. Results of algorithms for the proposed example

Approach	Routes	Travel distance (m)	Travel distance if not matched (m)	Saving ($)	Total Saving ($)
Greedy Matching Algorithm	D3, D10, R2, R8	6870.6	11925.8	17.7	26.6
	D2, D4, D5, R9, R4, R3	9771.4	12321	8.9	
	D1	500	500	0.0	
	D6	2573.9	2573.9	0.0	
	D7	250	250	0.0	
	D8	2610.1	2610.1	0.0	
	D9	2015.6	2015.6	0.0	
	R1	2761.3	2761.3	0.0	
	R5	2015.6	2015.6	0.0	
	R6	1250	1250	0.0	
	R7	707.1	707.1	0.0	
	R10	1802.8	1802.8	0.0	
Tabu Search Algorithm	D8, D2, D4, R9, R10, R8	10582.4	14090.7	12.3	27.9
	D3, D10, R2, R4	6369.3	10830.3	15.6	
	D1	500	500	0.0	
	D5	2061.6	2061.6	0.0	
	D6	2573.9	2573.9	0.0	
	D7	250	250	0.0	
	D9	2015.6	2015.6	0.0	
	R1	2761.3	2761.3	0.0	
	R3	1677.1	1677.1	0.0	
	R5	2015.6	2015.6	0.0	
	R6	1250	1250	0.0	
	R7	707.1	707.1	0.0	

In Figure 4 and Figure 5, the participants who are grouped together as a match are connected with lines of the same color as their destinations and the transfer stations are shown in a red circle in these figures. As shown in Figure 4, itineraries as the solution generated by TS algorithm, the matched groups are drivers 2, 4, and 8 with riders 8, 9, and 10 and drivers 3 and 10 with riders 2 and 4. As intended, some drivers (e.g., D2) carried a single rider (e.g., R8), while other drivers (e.g., D10) carried multiple riders (e.g., R2 & R4). Some riders (e.g., R9) were carried by a single driver (e.g., D4), while other riders (e.g., R4) were carried by multiple drivers (e.g., D10 & D3). Other drivers and riders not in the figure were unmatched due to two

reasons: either there was no match with cost-saving for these riders with the respective drivers, or available matches were not possible due to the presence of at least one of the drivers in more cost-effective groups.

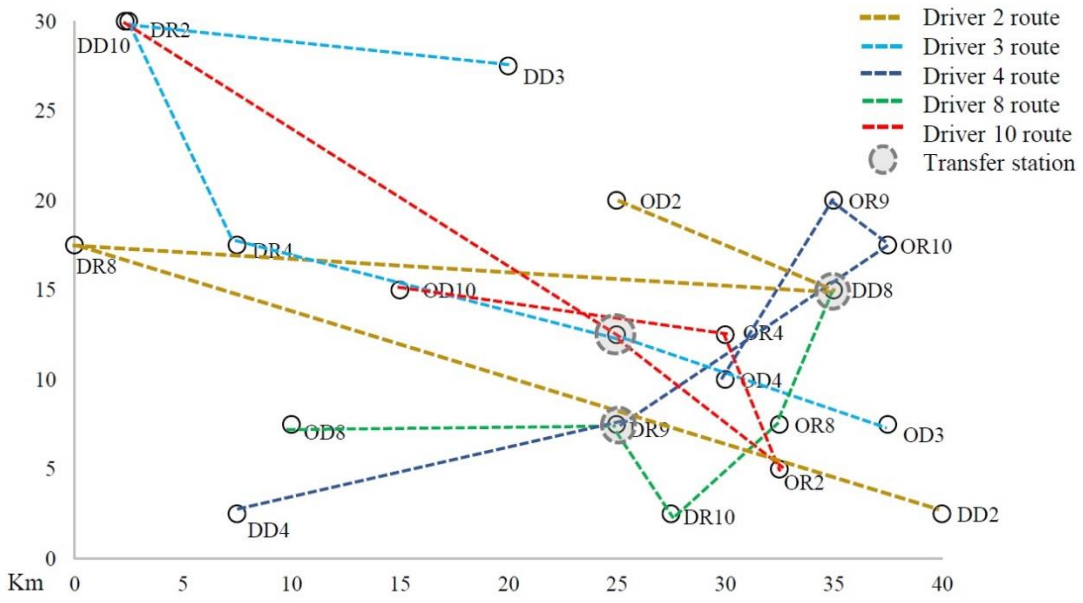

Figure 4 Itineraries of the matched drivers and riders in the solution generated running the TS algorithm

As shown in Figure 5, the GM algorithm consists of two groups of matched drivers and riders. In one group, drivers 2, 4, and 5 are matched with riders 3, 4, and 9. In the second group, drivers 3 and 10 are matched with riders 2 and 8. In the third group, drivers 9 and 7 are matched with riders 2 and 7. Like the results from the TS algorithm in Figure 5, some drivers (e.g., D10) carried a single rider (e.g., R2), while other drivers (e.g., D2) carried multiple riders (e.g., R9 & R4). Some riders (e.g., R4) were carried by a single driver (e.g., D2), while other riders (e.g., R2) were carried by multiple drivers (e.g., D3 & D10).

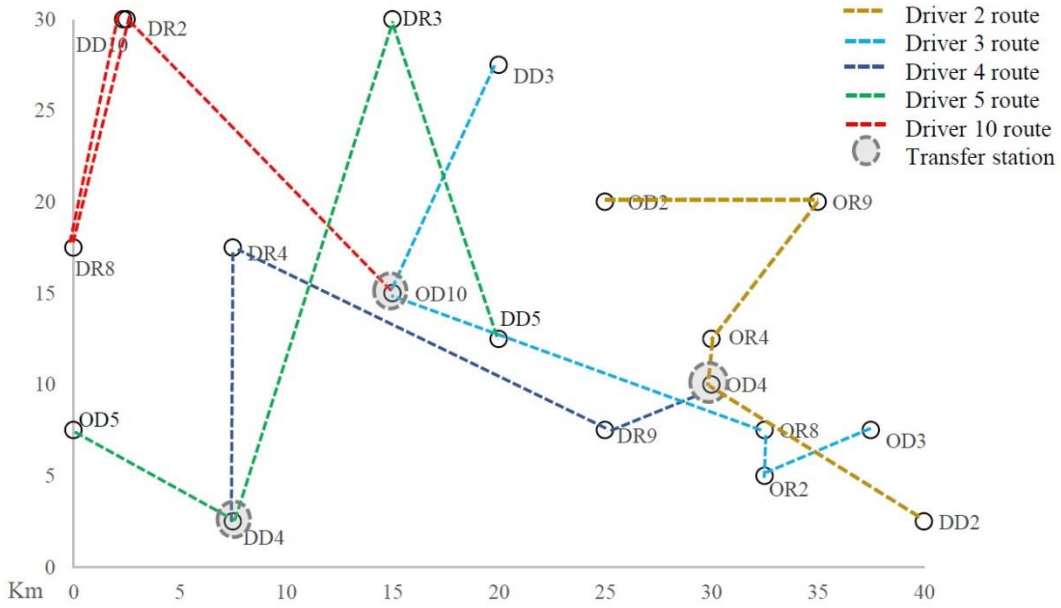

Figure 5 Itineraries of the drivers and single riders' paths in the solution generated running the GM algorithm

I.5 CONCLUSION

In recent years, shared mobility has become more available and attractive due to its economic efficiency, environmental advantages, and social equity aspects, especially where traditional transit service is not available. Among the shared mobility modes, peer-to-peer (P2P) ridesharing is considered the most economical and desirable because each individual involved in the ridesharing has a purpose for their travel, and drivers only need to be compensated for the additional travel distance and costs, which make the ridesharing more affordable for riders, compared to ride-hailing or shared ride-hailing services. P2P ridesharing not only requires trust between drivers and riders, but it also requires technological advancement to match drivers and riders efficiently. Recent studies have improved ride-matching algorithms; however, this problem still has room for improvement due to its complexity regarding consideration of spatial and temporal constraints with financial feasibility – to achieve minimization of total travel distance while maximizing the benefits for both drivers and riders.

In this research, a multi-driver, multi-rider (MDMR) P2P ride-matching problem based on rational preferences and cost allocation for both driver and rider was developed, and the algorithms using Tabu Search (TS) and Greedy Matching (GM) algorithms were formulated. Due to the multi-driver, multi-rider algorithm, not only can drivers transport multiple riders in their trips but also riders can transfer between multiple drivers for their travels if needed. A hypothetical example was developed to evaluate the performance and the computational efficiency of the proposed algorithms, and the developed algorithms could successfully solve the proposed P2P multi-driver, multi-rider ride-matching problem.

One important result of this study comes when we compare the results of the models in two forms of stable matches and non-stable matches. The TS algorithm provides a near-optimal solution in terms of system cost savings in which a better matching performance between riders and drivers is expected rather than the stable form of the model by the GM algorithm. Obviously, the metaheuristic approach using Tabu Search is seeking to minimize the total costs by relaxing stability constraints of the model; therefore, more cost savings and matchings are expected in this approach rather than a stable model. The maximization of the total cost saving of the system depends on how the rider and driver(s) can compromise on costs and trade-off. Therefore, ridesharing should consider incentive policies and reduce some constraints to boost the flexibility of the non-stable approach to maximize the number of matchings.

In most short-distance travel, ride transfer may not look practical, but for the long-distance travel, finding feasible ride-matching may be very difficult, and with a ride transfer option, it is possible to increase ride-matching and reduce total costs. Also, this ride-matching with ride transfer algorithm can be used for multiple on-demand flexible transit routings. Since the proposed algorithm has a main focus on long distance trips, this makes the algorithm more applicable in rural areas where fewer transportation options are available and travel distances are relatively longer. The results showed that a centralized ride-matching system may convince the driver to have profitable service in rural areas and it eventually makes the whole system more profitable (or less costly) in rural areas.

Future research could add more specific matching constraints between drivers and riders, such as same-gender matching and also use other metaheuristics and solving approaches. In addition, other heuristic methods could be tested so that the efficiency of the algorithm can be improved. Moreover, the proposed P2P ride-matching problem introduces a set of transfer stations randomly before matching the drivers and riders and then finds the optimal transfer station among the existing stations. Future studies should find the optimal locations of transfer stations before the matching and also consider more factors such as the waiting time for the vehicles and transfers in the model.

II. TRANSFER-ENABLED ON-DEMAND BUS NETWORK DESIGN WITH METAHEURISTIC APPROACH

II.1 INTRODUCTION

On-demand transit services have gained notoriety among transportation authorities and agencies in recent decades with the rapid growth of communication technologies. Currently, on-demand transportation services represent more than 30% of all daily commutes in North America, and this figure is expected to grow by 14% annually through 2026 [48]. New generations of transportations systems are trying to provide effective mobility services that are economically-competitive, compatible with emerging vehicle technology (like automated vehicles), and conscious of the needs of both the user and operator. Traditional fixed-route bus service is one of the transportation modes that has been most impacted by emerging on-demand services. A study by Rayle [49] discovered that ride-hailing services are responsible for 30% of the decline in public transit ridership in San Francisco. Carpooling services also decreased public transit ridership, as 75% of carpooling activity in the Bay Area shifted away from the public transportation system [50]. Compared with traditional bus transit service, on-demand bus service may offer several advantages that make it more efficient and attractive, route flexibility, lower travel costs for both the user and the operator, and increased network mobility. Figure 6 represents a comparison between possible advantages of on-demand services over fixed-route services.

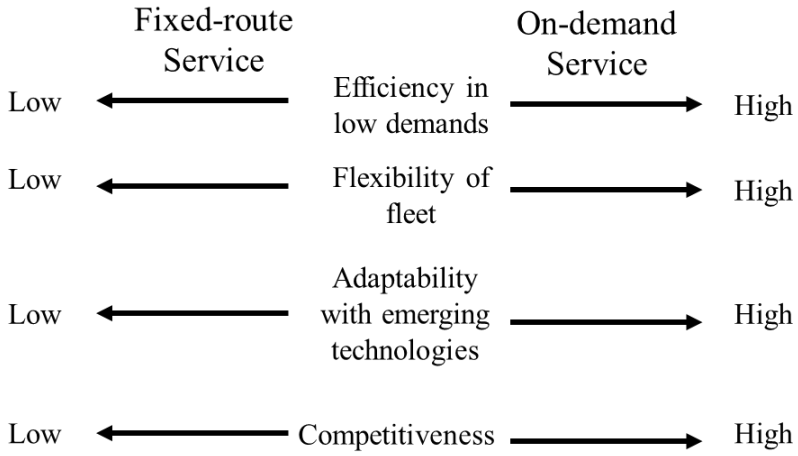

Figure 6. Potential benefits of on-demand services Compared with fixed-route services

Transfer points are a highly applicable concept in on-demand transit service that distinguish it from conventional ride-hailing services and make on-demand service more efficient and effective. Transfer points enable riders to be served by more than one vehicle from their origin to the destination, allowing for less travel time and shorter travel distances in the process. However, transfer points can only be effective when passengers can be served within operational time windows. Transfer points can be more efficient for riders with a longer traveling distance; therefore, considering time windows might help the rider be served better. This study aims to develop an optimal algorithm for an on-demand transit network that considers deploying transfer points and time windows for riders at the same time. This study first reviews the existing

literature and finds the potential area that needs to be covered. Then the study proposes optimization problem goals to minimize the total traveled distances. After that, an algorithm based on the metaheuristic method will be developed to solve the proposed problem. Finally, the performance of the proposed optimal algorithm will be evaluated on a hypothetical network with and without consideration of transfer points to show the performance of the approach. The results of this study could be utilized by transportation authorities, transport investment agencies, and collaborators in urban and suburban transportation systems.

II.2 LITERATURE REVIEW

The literature with respect to this topic can be categorized into two main parts: vehicle routing problems with simultaneous delivery and pickup and on-demand transit network design. The vehicle routing problem (VRP) is the basis of algorithms for ridesharing. The VRP algorithms are combinatorial optimization and integer programming problems that aim to find the optimal set of routes for vehicles to service a given number of customers. In past studies, the VRP algorithms have been divided into various classes to address specialized cases. The vehicle routing problem with pickup and delivery (VRPPD) is an extension of the VRP that has been used to address problems related to public transit and freight transportation. Researchers have further extended the VRPPD to the routing problem with pickup and delivery with time window (VRPPDTW), which is specialized for demand-responsive transit systems. The demand-responsive transit (DRT) or passenger-oriented transit system has the same structure as the VRPPDTWs. The main goal of the VRPPDTW is to propose a holistic routing optimization approach that minimizes total transportation network costs by considering the integration of vehicles and passengers' requests within space-time restrictions. Similar objective functions such as maximization of the total profits [51, 52] and service quality [53, 54] have been used in some studies. Generally, VRPPDTWs are designed to optimize vehicle routing and scheduling; however, this objective has often been expanded to assign passengers to vehicles more effectively (many-to-one or many-to-many) and determine optimal pricing.

Psaraftis [55, 56] introduced single vehicle VRPPDTW by developing a dynamic programming algorithm that aimed to minimize total waiting and riding times; however, this model was only able to run in small networks. Other studies tried to develop VRPPD as a linear program to minimize inconvenience to passengers. Their main shortcomings were that they only considered a single vehicle and one-sided time windows [57, 58].Considering multiple vehicles was a noticeable improvement that was introduced by Dumas et al. [59]. Their model aimed to minimize total vehicle costs using a column generation heuristic method. After developing this model, many scholars tried to develop a multiple vehicle VRPPDTW by adding more realistic constraints, modifying the objective function of the model by considering total vehicle fixed costs and route costs, and implementing different heuristic approaches such as Branch-and-cut-and-price and set-partitioning [60-68]. More recent studies have mainly focused on developing a VRPPDTW that overcomes the aforementioned shortcomings by using larger networks, state-space-time networks, and new heuristic and metaheuristic methods.

The on-demand transit network design considers the advantages that technology and real-time data may provide to the transportation system's route and scheduling flexibility. Addressing the on-demand transit problem is complicated by the dynamic nature of passengers' locations and time-windows. This complexity may vary depending on the scale of the network and the structure of communication between the user and the operator. In the existing literature, researchers have explored two primary approaches to address this complexity. In both cases, the

problem is treated as static. In the first approach, passenger demand is assumed to remain constant, requiring real-time data updates for each solution. On the other hand, the second approach employs metaheuristics and heuristic methods, enabling the solution to be continuously updated [14]. The insertion heuristic has been widely used to deal with the dynamic structure and randomness of passenger demand [69]. In terms of considering constraints, most of the studies considered time windows and the ability to pick up and deliver passengers at the same time. However, a few studies have considered transfer points for on-demand service.

The collaborative design of a transfer Customized Bus (CB) operational network, combined with passenger-route assignments encompassing transfer operations and a modular fleet, represents a relatively nascent area of study. Gong et al. [70] designed a transfer-based CB network utilizing a modular fleet while concurrently optimizing the assignment of passenger routes. They sought to ascertain the optimal network structure under this innovative design paradigm by employing a linearization approach in conjunction with a particle swarm optimization (PSO) algorithm. However, their approach did not take into account the passengers' travel time window. This study attempts to cover this gap. Therefore, the main contribution of this study is to consider transfer points for riders along with time windows and the ability to pick up and deliver passengers at the same time. Table 2 represents a summary of relevant studies on on-demand transit network design.

Table 2. Summary of selected, relevant, and recent studies on the design of on-demand transit networks

Study	Solving method	Objective function	Constraints			
			Vehicle capacity	Time windows	Pickup and delivery	Transfer point
Tong et al., 2017 [54]	Heuristics, lagrangian decomposition	Min. the number of unserved passengers	X	X	X	
Mahéo et al., 2017 [71]	Heuristics, benders decomposition	Min. the total operating cost	X	X	X	
Guo et al., 2018 [72]	Metaheuristics, Genetic Algorithm	Min. the total operating cost	X		X	
Wang, 2019 [73]	Metaheuristics, Tabu search	Min. the total in-vehicle travel time	X	X	X	
Lyu et al., 2019 [52]	Heuristics for a non-linear problem	Max. the total profit	X			
Huang et al., 2020 [51]	Heuristics, insertion	Min. the total cost, Max. profit	X	X	X	
Wu et al., 2021 [74]	Heuristics, A* algorithm	Min. the total number of passenger transfers	X		X	X
Chen et al., 2021 [75]	Heuristics, Modified sweep-based algorithm	Min. the total operating cost	X	X	X	
Dou et al., 2021 [76]	Heuristics, branch-and-price	Max. the total profit	X	X	X	

Wang et al., 2021 [77]	Heuristics, step-wise searching	Max. the total profit	X	X	X	
Ma et al., 2023 [78]	Metaheuristics, genetic and large neighborhood search algorithms	Max the profit minus the operational costs	X	X	X	
This study	Metaheuristics, Tabu search	Min. the total operating cost	X	X	X	X

II.3 PROBLEM DEFINITION

This study presents a Vehicle Routing Problem with Pickup and Delivery and Transfer points (VRPPD-T) in which the centralized system offers itineraries to drivers in order to deliver all passengers from their origins to their destinations while minimizing the total cost of transportation. The cost of transportation is defined as the weighted sum of total travel times of all participants and the total distance traveled by the vehicles. Furthermore, a constraint which implies that all passengers must be delivered to their destinations within an allowable time window is considered. Riders may transfer between multiple vehicles. Therefore, the system sets the location where vehicles transfer passengers from one vehicle to another.
In this section, a Mixed Integer Programming(MIP) formulation is presented to solve the VRPPD-T. In this approach, some dummy points are added to the set of points represented by T where these points are representative of transfer points between each pair of vehicles. In fact, we consider a transfer point between any two vehicles that may be visited if needed. The geographical position of these transfer points is calculated as the middle of the two clusters visited just before the transfer point. The service time of transfer points is considered to be zero. The distance of the two clusters is determined through the Euclidian distance between the last point visited in the first cluster and the first point visited in the second cluster. The MIP model, parameters, and variables are as follows:

Sets and parameters

P: The set of passengers

S: The set of points including the passengers' origins and destinations and vehicles' start positions

$costD$: The cost of distance traveled by vehicles per meter

$costT$: The cost of customers' travel times per minute

dis_{ij}: The distance between each pair of locations $i \in S$ and point $j \in S$ in meter

o_p: The point where passenger $p \in P$ starts his/her trip.

d_p: The point related to passenger's $p \in P$ destination

o_k: The point where the driver of vehicle k starts his trip

Q: Maximum number of passengers that vehicles can carry

K: The set of vehicles

TW: All participants should be serviced within TW.

C: The set of clusters

T: The dummy set of transfer points where each member defines a transfer point between two vehicles

SP: The set of origins and destinations of all passengers

SD: The set of points where drivers of vehicles start their trips

S: The set of points, where passengers and drivers start and passengers end their trips, $S = SP \cup SD$

$Tr_{k.l}$: Transfer point related to vehicle $k \in K$ and $l \in K$

A+: The set of arcs $(i.j)$: $(i.j \in C) \cup (i \in SD.j \in C) \cup (i \in T.j \in C) \cup (i \in C.j \in T)$

A: The set of arcs $(i.j)$: $(i.j \in C) \cup (i \in SD.j \in C)$

S_{ck}: service time of cluster $c \in C \cup SD \cup T$ when visited by vehicle $k \in K$, the service time of points related to drivers' positions and transfer points are considered zero.

co_p: Cluster which includes the origin of passenger $p \in P$

cd_p: Cluster which includes the destination of passenger $p \in P$

$dis_{(i-j)m}$: The distance between the middle of last points visited in clusters $i \in C$ and $j \in C$ with the first point visited in cluster $m \in C$

dis_i: The distance traveled inside cluster $i \in C$ which is determined at the second step based on the best order of points included in the cluster

TT_i: sum of travel time of passengers inside cluster $i \in C$ which is determined at the second step based on the best order of points included in the cluster

ca_i: Change in the capacity of a vehicle after visiting point $i \in C \cup SD \cup T$

Mca_i: Maximum number of passengers accompanied with each other visiting cluster $i \in C$

$speed_k$: Speed of vehicle $k \in K$

M: A big Number

Variables

x_{ijk}: A binary variable equal to 1 if vehicle $k \in K$ travels from cluster i to cluster j where $(i.j) \in A$

y_{ik}: A binary variable equal to 1 if cluster $i \in C$ with vehicle $k \in K$

TT_p: Travel time of passenger $p \in P$ between clusters.

AT_{ik}: Arrive time of vehicle $k \in K$ to the starting point of cluster $i \in C$ or transfer point $i \in T$.

cup_i: Capacity of the vehicle used when visiting point $i \in C \cup SD \cup T$

$DisTC_{klc}$: Distance traveled from the transfer point Tr_{kl} to cluster $c \in C$ in the route of vehicles $k \in K$ and $l \in K$

$DisCT_{kl}$: Distance traveled from clusters to transfer point Tr_{kl} in the route of vehicles $k \in K$ and $l \in K$

MIP Formulation (1)

Min $costD \times \sum_{(i.j)\in A} \sum_k x_{ijk} \times dis_{ij} + costT \times \sum_{p\in P} TT_p + \sum_{i\in c} costD \times dis_i + \sum_{i\in c} costT \times TT_i + costD \times \sum_{k.l\in K} DisCT_{kl} + costD \times \sum_{k.l\in K} \sum_{c\in C} DisTC_{klc}$ (32)

s.t.

$AT_{jk} \geq AT_{ik} + S_{ik} + dis_{ij}/speed_k - M(1 - x_{ijk})$ $\forall (i.j) \in A, \forall k \in K$ (33)

$AT_{Tr_{kl}k} \geq AT_{ik} + S_{ik} + 0.5 \times DisCT_{kl}/speed_k - M(1 - x_{iTr_{kl}k})$ $\forall k.l \in K; \forall i \in C$ (34)

$AT_{ik} \geq AT_{Tr_{kl}k} + 0.5 \times DisTC_{kli}/speed_k - M(1 - x_{Tr_{kl}ik})$ $\forall k.l \in K; \forall i \in C$ (35)

$AT_{cd_pk} \geq AT_{co_pk}$ $\forall k \in K; \forall p \in P$ (36)

$AT_{cd_pk} \geq AT_{Tr_{kl}k} + M(y_{co_pl} + y_{cd_pk} - 2)$ $\forall k.l \in K; \forall p \in P$ (37)

$AT_{co_pk} \leq AT_{Tr_{kl}k} + M(y_{co_pl} + y_{cd_pk} - 2)$ $\forall k.l \in K; \forall p \in P$ (38)

$y_{ik} = \sum_{(j.i)\in A+} x_{jik}$ $\forall k \in K; \forall i \in C$ (39)

$\sum_k y_{ik} = 1$ $\forall i \in C$ (40)

$AT_{ik} + S_{ik} \leq TW$ $\forall k \in K; \forall i \in C \cup T$ (41)

$\sum_{j\in C} x_{o_kjk} = 1$ $\forall k \in K; \forall i \in C$ (42)

$DisCT_{kl} \geq dis_{ij} + M(x_{iTr_{kl}k} + x_{jTr_{kl}l} - 2)$ $\forall k.l \in K; \forall i.j \in C$ (43)

$DisTC_{klm} \geq dis_{(i-j)m} + M(x_{Tr_{kl}mk} + x_{Tr_{kl}ml} x_{iTr_{kl}k} + x_{jTr_{kl}l} - 3) \forall k.l \in K; \forall i.j.m \in C$ (44)

$cu_j \geq cu_i + ca_i - M(1 - \sum_{k\in K} x_{ijk})$ $\forall (i.j) \in A +, \forall k \in K$ (45)

$$cu_i + Mcap_i \leq Q \qquad \forall i \in C \qquad (46)$$

$$\sum_{(i.j)\in A+} x_{iTr_{kl}l} = \sum_{(i.j)\in A+} x_{iTr_{kl}k} \qquad \forall k.l \in K \qquad (47)$$

$$y_{ik} \in \{0,1\}; x_{ijk} \in \{0,1\} \qquad \forall i.j \in C; \forall k \in K \qquad (48)$$

$$DisCT_{kl} \geq 0; DisCT_{kl} \geq 0; TT_p \geq 0; AT_{ik} \geq 0 \qquad \forall i \in C; \forall k.l \in K; \forall p \in P \qquad (49)$$

Equation (32) represents the objective function of the presented model, minimizing the total cost of passengers' travel times inside and between clusters. The cost of total distance traveled is included in the objective function as well. Constraints (33)-(35) define the trips' time flow. Constraints (36)-(38) force the model to satisfy precedence constraints and ensure that passengers' destinations are visited after their origin in each vehicle. Constraint (39) defines the relationship between two sets of variables. Constraint (40) ensures that all clusters are visited on the vehicle routes. Constraint (41) forces the model to satisfy time window constraints. Constraint (42) directs the drivers out of their original positions. Constraints (43)-(44) define the distance traveled from clusters to transfer points and vice versa. Constraints (45) and (46) ensure that vehicles capacities are not exceeded. Constraints (47) ensures that if a transfer point is used in the solution, both vehicles are passed through it.

II.4 ALGORITHMS

In the first step of the solution process, a Simulated Annealing (SA) metaheuristic algorithm is executed to reach a good solution as a start point for the rest of the solving procedure. The detailed steps of the SA algorithm are provided in Figure 7. In the second step, the solution generated by the SA algorithm is refined by separately solving a common VRPPD for each vehicle. However, the set of customers serviced by each vehicle remains constant, as established by the solution produced by the SA algorithm, while taking into account the constraints related to capacity and time windows. In step three, based on the solution returned in the second step, the set of points including the passengers' origins and destinations are clustered into disjointed groups by running the clustering algorithm presented in Figure 8. The clustering algorithm searches for the passengers who are delivered directly or with a little increase in their travel times but have origin and destination points that are far away. Transferring these passengers may increase the feasible solution space and enhance the objective value of the problem. Furthermore, it separates the customers with the most increase in their travel times into disjoint clusters in the hope of achieving better travel times using transfer points.

In the third step, we fix the order of points visited inside each cluster based on the solution produced by the clustering algorithm in the second step (Figure 9). As result, the size of the original problem reduces to a smaller one where we can solve it with a common MIP solver. In fact, instead of solving a VRPPD-T with a size of 2*|P| points where |P| is the number of passengers, a VRP with the size of |C| points is solved where |C| is the number of clusters returned in the second step. The new problem tries to find the order of clusters visited by each vehicle, minimizing the cost, considering the capacity and time window constraints, and ensuring the origin of each passenger is visited before its destination (precedence constraint). The sets, parameters, formulation, and details of solving the VRPPD-T are presented in section 2.

After solving formulation (1), we once again solve a common VRPPD for the vehicles while the visiting points of vehicles are fixed based on the solution provided by formulation (1) in hopes of

reducing total transportation costs. Here, the geographic positions of transfer points and the origin and destination of each passenger in each vehicle are also known and fixed as the one returned by solving formulation (1). Moreover, we give the VRPPD a starting solution which is the solution returned by solving formulation (1). Then the cost returned by the solver is returned as the cost of the original problem.

Step 0: ***Run the Simulated Annealing (SA) algorithm and save the routes returned by the algorithm***
Step 1: ***Solve a VRPPD for each vehicle separately, considering that points visited by vehicles are fixed and known based on the solution returned in step 0***
Step 2: ***Run the clustering algorithm and save the clusters and the order of points in each cluster***
Step 3: ***Solve the VRPPD-T with reduced size using formulation (1) and save the returned solution***
Step 4: ***Solve a VRPPD considering that points visited by vehicles are fixed and known based on the solution returned in step 3***
Step 5: ***Return the solution of the* VRPPD *as the best solution to the problem***

END

Figure 7. The algorithm of solving VRPPD-T

Step 0: ***Initialization***:
T0=0.01, T=T0, alpha=0.995, Max_it=5000, it=1
Step 2: ***Create random solution***
Set x as a random solution, by generating a random permutation, with the length of *n*, i.e, number of riders +buses-1.
Step 3: ***Calculate cost of x***
Step 3.1: Considering the sum of traveled distances of buses and sum of travel time of riders, and penalty costs (penalty for capacity and time window constraints) calculate the cost of solution x.
Step 3.2: Set BestSolution=x and Best Cost=Cost of solution x
Step 4: **IF** It< Max_it, **THEN**
go to step 4.1, otherwise go to **step 5**
END IF
Step 4.1: ***Creating neighborhood***:
Random= Generate a random value within [0,1]
Step 4.1.1: **IF** Random < 0.75, **THEN**
go to step *4.1.2*, otherwise go to *step 4.1.3*
END IF

Step 4.1.2: set xnew = a neighborhood of x by changing the visiting order of the rider(s) through a bus and go to *step 4.2*

Step 4.1.3: set xnew = a neighborhood of x by changing the assigned bus of a rider

Step 4.2: **IF** best cost for x< best cost for xnew, **THEN**

set x=xnew and go to *step 4.5*, otherwise, go to *step 4.3*

END IF

Step 4.3: p= exp(cost xnew – cost x)/T*Cost x

Step 4.4: Accept x= xnew by p -probability and reject- and x= xnew by (1-p) and go to *step 4.5*

Step 4.5: **Cost calculation** for xnew

Step 4.6: **IF** Cost for xnew > best cost, **THEN**

set **bestsol**= xnew

END IF

Step 4.7: Reducing the temperature: set T = alpha*T0 (0<alpha<1)

Step 5: Set It=It+1 and go to **step 4**

Step 6: **END**

Figure 8. SA algorithm

Step 0: ***Initialization***:

Make vector "All" by concatenating routes returned in step 1 for vehicles and insert -1 before the start position and after the end position of the route of each vehicle in "All"

Step 1: ***Calculate the difference of each passenger's travel time in the returned solution and its travel time when delivered directly***

For each passenger

Calculate the difference between the passenger's travel time in the returned solution and its travel time when delivered directly. Save the value as $dif[i]$.

End For

Step 2: ***Separate passengers with the most increase in travel times into disjoint clusters***

For the first n1 customers with the greatest $dif[i]$

Insert -1 before and after the position of the passenger's origin and destination in vector "All"

End For

Step 3: ***Find the passengers with the least travel times and the most distance between their origin and destination points***

Find the n2 passengers with the greatest values of $costD \times distance(o_i.d_i) - costT \times dif[i]$

Step 4: ***Separate passengers found in the last step into disjoint clusters***

For each passenger found in step 3:

Insert "-1" before and after the position of the passenger's origin and

destination in vector "All"

End For

Step 5: ***In the routes returned by the SA algorithm, find the n3 arcs with the greatest distance between the two edges of the arc***

Step 6: ***Cut the routes by inserting "-1" between the two edges of each arc found in step 5***

Step 7: ***Eliminate one of each two -1 right after each other in vector "All"***

Step 8: ***Consider the points between each two "-1" in vector "All" as a separate cluster and save the order of points visited in each cluster the same as the order of points in vector "All"***

Step 9: ***Return the clusters and the order of points in each cluster.***

END

Figure 9. Clustering algorithm

II.5 EXAMPLE

A hypothetical rail transit line that has four stations fitted to urban and suburban conditions was considered to examine and evaluate the efficiency of the algorithm. In this example, a transit system with three buses and 30 riders (passengers) has been considered. The time window for passengers was assumed to be 45 minutes, and 15 seats were assigned to each bus. Two different networks (with and without transfer points) were designed to evaluate the performance of transfer points in the on-demand network design problem. The total cost of each bus per kilometer (including the bus driver and the vehicle traveling costs) has been calculated as 7.31 $/km [79] and the value of time for each passenger has been calculated as $20 per hour [80]. The proposed algorithm was coded in C++, and the Formulation (1) and the VRPDD models were solved by CPLEX 12.3. Figure 10 shows the origin of riders and drivers including the stations, and Figure 11 represents the destination of riders including the stations.

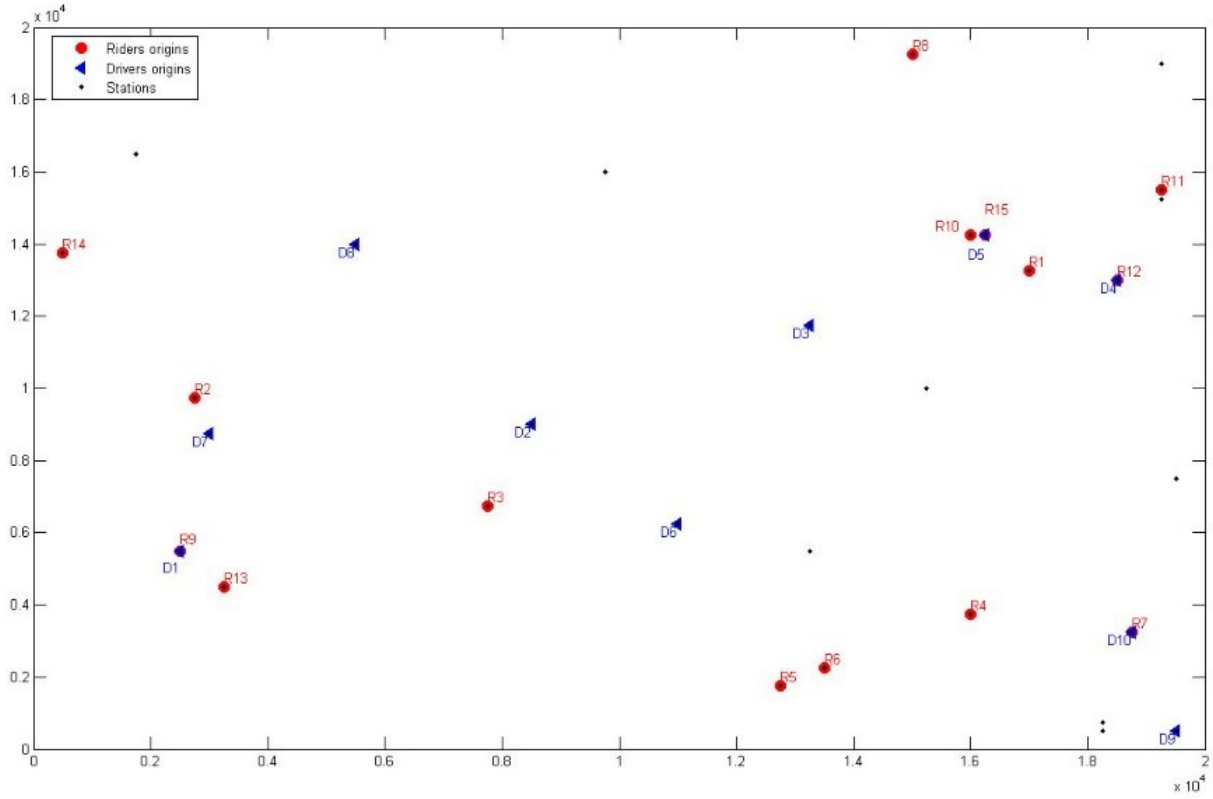

Figure 10. Origin locations of riders and drivers including the transfer stations

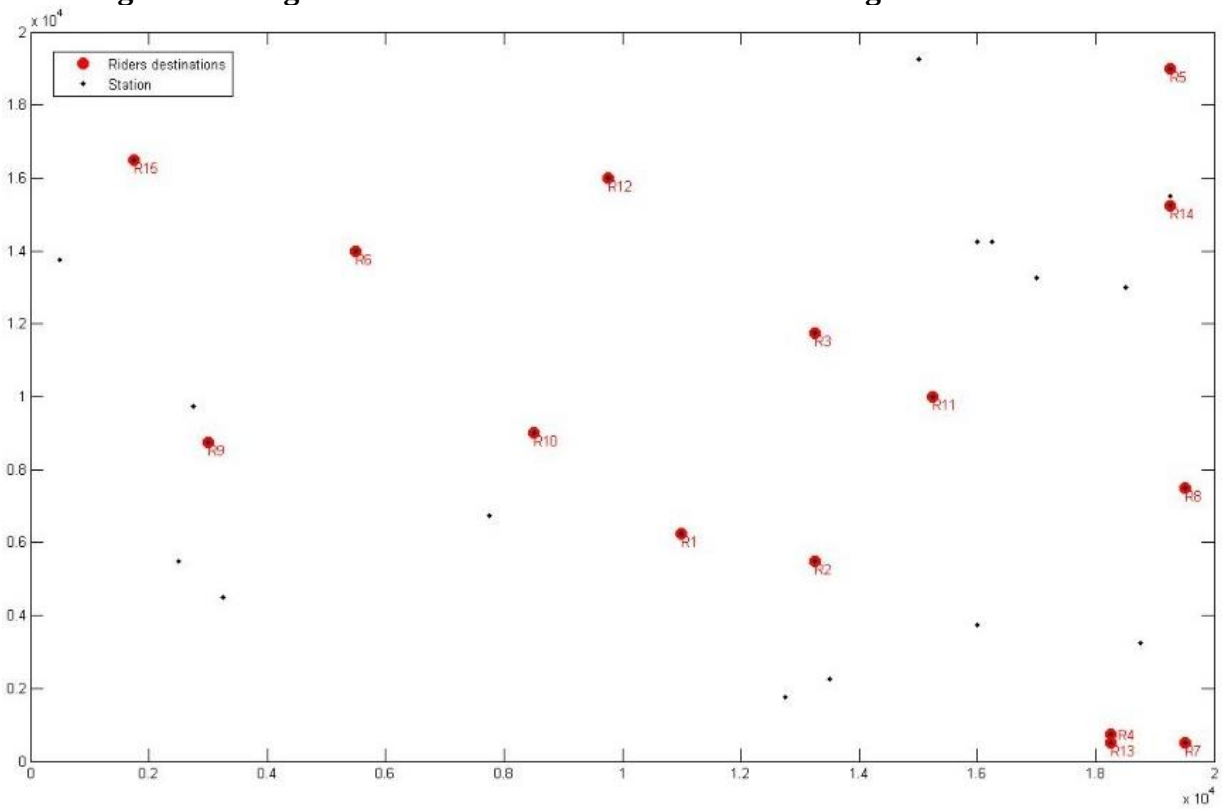

Figure 11. Destination locations of riders including the transfer stations

Table 3 and 4 show the results of proposed models for two different scenarios with transfer points and without transfer points. An essential consideration in the model when factoring in the transfer point is the timing of the second bus's departure such that the simultaneous arrival of both buses at the transfer point is ensured. In Table 3, the stations are denoted using the letter 's'. The results after running each model 10 times show that the best total cost for the case without transfer points is $454.60 and for the case with transfer point is $442.20. In addition, Table 5 provides a comparison between the final results of the two models to figure out the effect of considering transfer points in the network

Figures 12 and 13 show the itineraries of the matched buses and riders with and without considering transfer points. One noticeable outcome in the model that considered transfer points was that the second bus started its travel later, but it arrived to the transfer point with the third bus at the same time. In this way, the waiting time of the riders can be decreased in a way that reduces the total costs in the model.

Table 3. Results of algorithms for the proposed example with transfer points

Bus	Route														Cost
B1	R9	R6	R29	R28	R15	R17	R22	R3	R1	R23	R13	R4	R20		191.753
B2	R5	R11	R27	R24	R14	R18	R30	R10	R12	R8	S	R2	R16	R7	175.478
B3	R25	R21	S	R26	R19										75.0115

Table 4. Results of algorithms for the proposed example without transfer points

Bus	Route													Cost
B1	R9	R6	R29	R28	R15	R17	R22	R3	R1	R23	R13	R4	R20	191.75
B2	R5	R24	R25	R21	R8	R2	R16	R7						133.67
B3	R11	R27	R14	R18	R30	R10	R12	R26	R19					129.23

Table 5. Comparison of results of algorithms for the two cases

Fleet	Total traveled distance without considering transfer points (km)	Total traveled distance with considering transfer points (km)	Total cost without considering transfer points ($)	Total cost with considering transfer points ($)
B1	22.2	22.2	191.8	191.8
B2	15.7	19.5	133.7	175.5
B3	15.4	9.4	129.2	75.0
Total	454.653	442.243	454.6	442.2

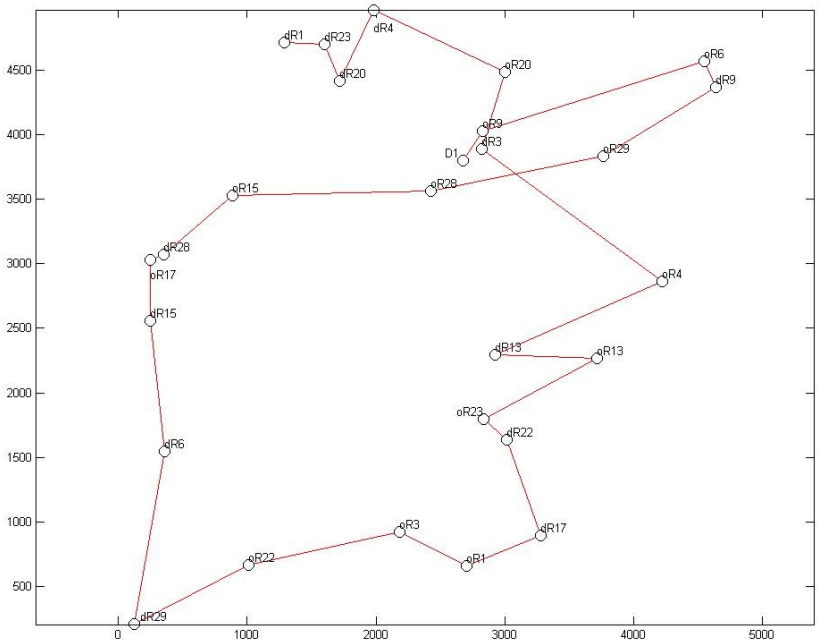

Figure 12 (a). Itineraries of the matched buses and riders in the solution with considering the transfer points (bus 1)

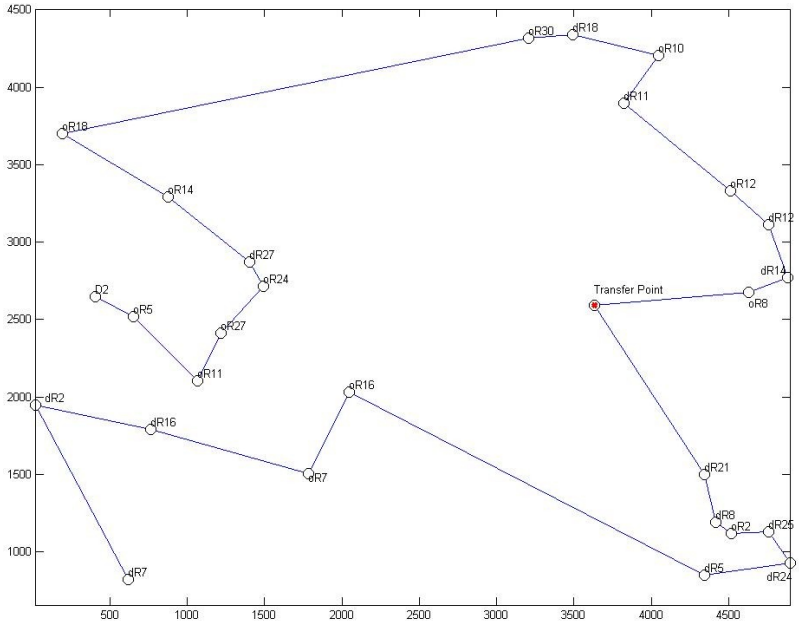

Figure 12 (b). Itineraries of the matched buses and riders in the solution with considering the transfer points (bus 2)

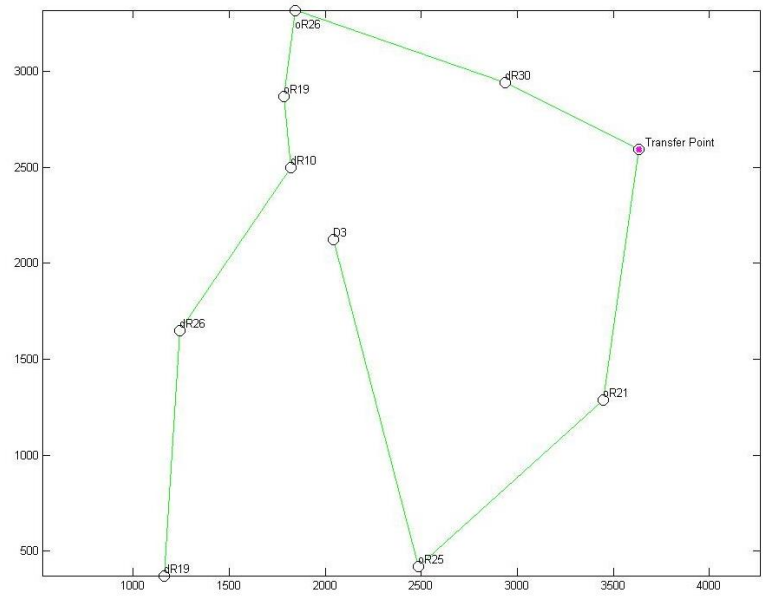

Figure 12 (c). Itineraries of the matched buses and riders in the solution with considering the transfer points (bus 3)

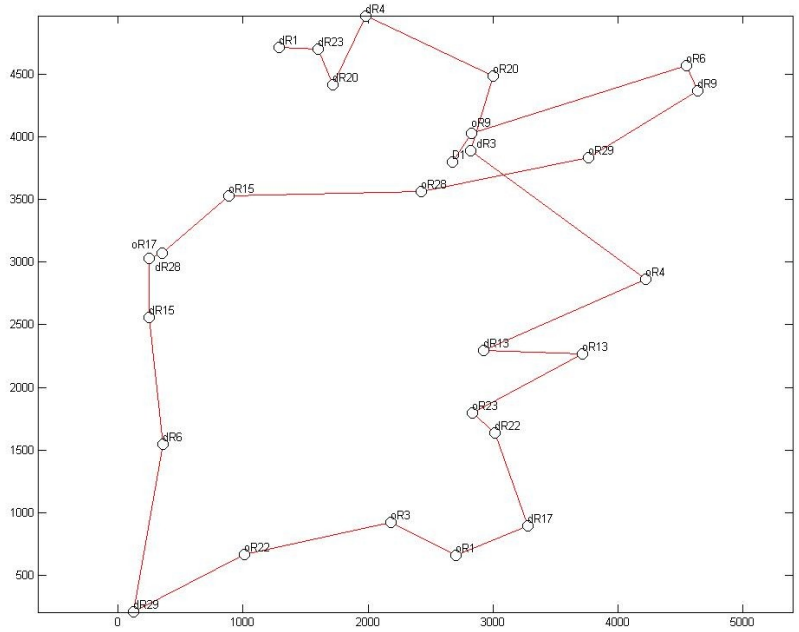

Figure 13 (a). Itineraries of the matched buses and riders in the solution without considering the transfer points (bus 1)

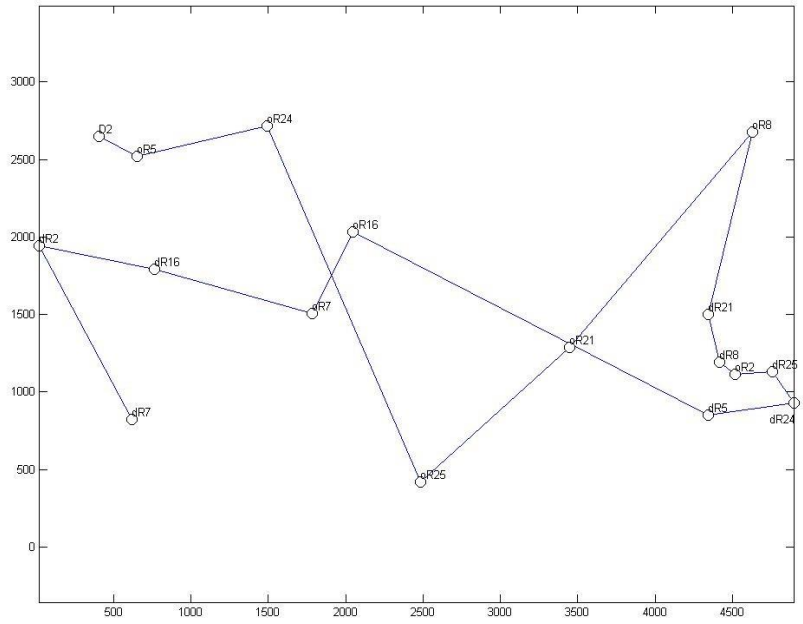

Figure 13 (b). Itineraries of the matched buses and riders in the solution without considering the transfer points (bus 2)

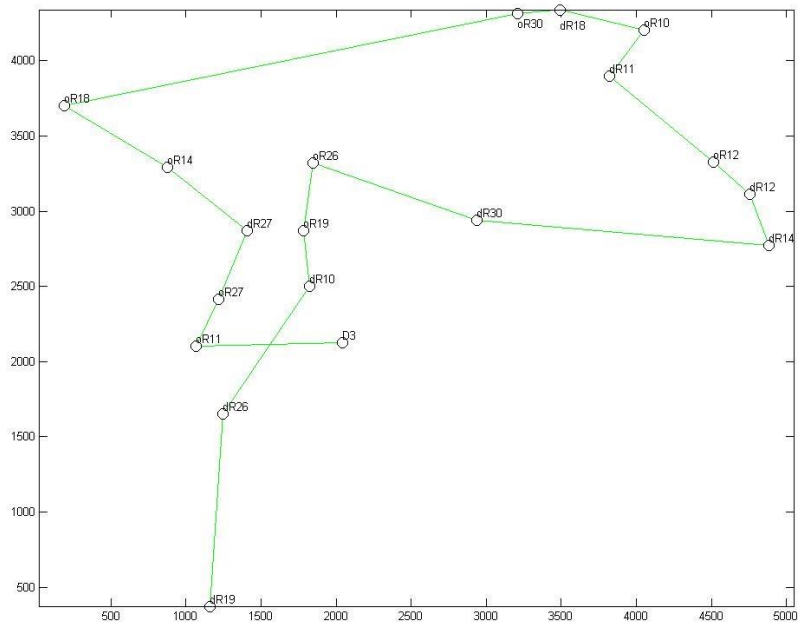

Figure 13 (c). Itineraries of the matched buses and riders in the solution without considering the transfer points (bus 3)

II.6 CONCLUSION

This study focused on improving on-demand transit systems. Emerging advanced communication technologies have opened new doors and allowed public transportation systems to incorporate intelligent approaches that boost their efficiency and performance. On-demand services have already demonstrated noticeable efficiency gains compared to conventional fixed-route transit services. However, applying these emerging technologies to on-demand services may drive even greater improvements in performance and efficiency while reducing total travel costs. This study developed an algorithm for an optimal on-demand network design accommodating transfer points for the users. This integrated design approach included two main challenges: a complex structure in which passenger boarding, alighting, and transferring activities occur simultaneously at a single station and the joint network design and passenger-route assignment which ostensibly expands the solution space. The proposed optimization problem considered time windows and simultaneous pickup/delivery in its constraints. As the results of an example on a hypothetical network showed, an algorithm that considers transfer points can save up to 2.73% of total costs in the whole transit system.
Future studies can add multi-modal features to the model by using various modes of transportation with different characteristics. Considering other algorithms and solving methods for the model may also increase the quality of the final results.

REFERENCES

1. Dailey, D.J., D. Loseff, and D. Meyers, *Seattle smart traveler: dynamic ridematching on the World Wide Web.* Transportation Research Part C: Emerging Technologies, 1999. **7**(1): p. 17-32.
2. Ordóñez, F. and M.M. Dessouky, *Dynamic Ridesharing*, in *Leading Developments from INFORMS Communities*. 2017. p. 212-236.
3. Mohamed, M.J., T. Rye, and A. Fonzone, *Operational and policy implications of ridesourcing services: A case of Uber in London, UK.* Case Studies on Transport Policy, 2019. **7**(4): p. 823-836.
4. Liu, X., et al., *A passenger-to-driver matching model for commuter carpooling: Case study and sensitivity analysis.* Transportation Research Part C: Emerging Technologies, 2020. **117**: p. 102702.
5. Ashkrof, P., et al., *Understanding ride-sourcing drivers' behaviour and preferences: Insights from focus groups analysis.* Research in Transportation Business & Management, 2020: p. 100516.
6. Nickkar, A., Y.-J. Lee, and S. Dadvar, *Impact of Automated Vehicles on Optimal Demand-Responsive Feeder Transit Network Design.* International Journal of Urban Planning and Smart Cities (IJUPSC), 2021. **2**(1): p. 84-100.
7. Nickkar, A., Y.-J. Lee, and M. Meskar. *Sensitivity analysis for the optimal automated demand responsive feeder transit system*. in *17th International Conference on Automated People Movers and Automated Transit Systems*. 2020. American Society of Civil Engineers Reston, VA.
8. Nickkar, A., Y.-J. Lee, and M. Meskar, *Developing an optimal algorithm for demand responsive feeder transit service accommodating temporary stops.* Journal of Public Transportation, 2022. **24**: p. 100021.
9. Cordeau, J.-F. and G. Laporte, *The dial-a-ride problem: models and algorithms.* Annals of Operations Research, 2007. **153**(1): p. 29-46.
10. Braekers, K., A. Caris, and G.K. Janssens, *Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots.* Transportation Research Part B: Methodological, 2014. **67**: p. 166-186.
11. Masson, R., F. Lehuédé, and O. Péton, *The Dial-A-Ride Problem with Transfers.* Computers & Operations Research, 2014. **41**: p. 12-23.
12. Lee, Y. J., & Nickkar, A. (2021). *U.S. Patent Application No. 17/017,084*.
13. Lee, Y.-J., et al., *Development of an Algorithm for Optimal Demand Responsive Relocatable Feeder Transit Networks Serving Multiple Trains and Stations.* Urban Rail Transit, 2019. **5**(3): p. 186-201.
14. Berbeglia, G., J.-F. Cordeau, and G. Laporte, *Dynamic pickup and delivery problems.* European Journal of Operational Research, 2010. **202**(1): p. 8-15.
15. Masoud, N. and R. Jayakrishnan, *A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system.* Transportation Research Part B: Methodological, 2017. **106**: p. 218-236.
16. Masoud, N. and R. Jayakrishnan, *A decomposition algorithm to solve the multi-hop Peer-to-Peer ride-matching problem.* Transportation Research Part B: Methodological, 2017. **99**: p. 1-29.

17. Herbawi, W. and M. Weber. *The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm*. in *2012 IEEE Congress on Evolutionary Computation*. 2012.
18. Stiglic, M., et al., *The benefits of meeting points in ride-sharing systems.* Transportation Research Part B: Methodological, 2015. **82**: p. 36-53.
19. Agatz, N.A.H., et al., *Dynamic ride-sharing: A simulation study in metro Atlanta.* Transportation Research Part B: Methodological, 2011. **45**(9): p. 1450-1464.
20. Boyacı, B., K.G. Zografos, and N. Geroliminis, *An optimization framework for the development of efficient one-way car-sharing systems.* European Journal of Operational Research, 2015. **240**(3): p. 718-733.
21. Tafreshian, A. and N. Masoud, *Trip-based graph partitioning in dynamic ridesharing.* Transportation Research Part C: Emerging Technologies, 2020. **114**: p. 532-553.
22. Silva, B.C.H., et al., *Quota travelling salesman problem with passengers, incomplete ride and collection time optimization by ant-based algorithms.* Computers & Operations Research, 2020. **120**: p. 104950.
23. Wang, X., N. Agatz, and A. Erera, *Stable Matching for Dynamic Ride-Sharing Systems.* Transportation Science, 2018. **52**(4): p. 850-867.
24. Ma, R., et al., *A novel algorithm for peer-to-peer ridesharing match problem.* Neural Computing and Applications, 2019. **31**(1): p. 247-258.
25. Aydin, O.F., I. Gokasar, and O. Kalan, *Matching algorithm for improving ride-sharing by incorporating route splits and social factors.* PLOS ONE, 2020. **15**(3): p. e0229674.
26. Di Febbraro, A., E. Gattorna, and N. Sacco, *Optimization of Dynamic Ridesharing Systems.* Transportation Research Record, 2013. **2359**(1): p. 44-50.
27. Nourinejad, M. and M.J. Roorda, *Agent based model for dynamic ridesharing.* Transportation Research Part C: Emerging Technologies, 2016. **64**: p. 117-132.
28. Kucharski, R. and O. Cats, *Exact matching of attractive shared rides (ExMAS) for system-wide strategic evaluations.* Transportation Research Part B: Methodological, 2020. **139**: p. 285-310.
29. Nourinejad, M. and M. Ramezani, *Ride-Sourcing modeling and pricing in non-equilibrium two-sided markets.* Transportation Research Part B: Methodological, 2020. **132**: p. 340-357.
30. Balardino, A.F. and A.G. Santos. *Heuristic and Exact Approach for the Close Enough Ridematching Problem*. 2016. Cham, Springer International Publishing.
31. Najmi, A., D. Rey, and T.H. Rashidi, *Novel dynamic formulations for real-time ride-sharing systems.* Transportation Research Part E: Logistics and Transportation Review, 2017. **108**: p. 122-140.
32. Yang, H., et al., *Optimizing matching time interval and matching radius in on-demand ride-sourcing markets.* Transportation Research Part B: Methodological, 2020. **131**: p. 84-105.
33. Kumar, P. and A. Khani, *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, 2021. **122**: p. 102891.
34. Ghoseiri, K., A. Haghani, and M. Hamed, *Real-time rideshare matching problem.* 2010, Mid-Atlantic Universities Transportation Center.
35. Qian, X., et al., *Optimal assignment and incentive design in the taxi group ride problem.* Transportation Research Part B: Methodological, 2017. **103**: p. 208-226.

36. Peng, Z., et al., *Stable ride-sharing matching for the commuters with payment design.* Transportation, 2020. **47**(1): p. 1-21.
37. Özkan, E., *Joint pricing and matching in ride-sharing systems.* European Journal of Operational Research, 2020. **287**(3): p. 1149-1160.
38. Xia, J., et al., *A New Model for a Carpool Matching Service.* PloS one, 2015. **10**(6): p. e0129257-e0129257.
39. Xu, H., F. Ordóñez, and M. Dessouky, *A traffic assignment model for a ridesharing transportation market.* Journal of Advanced Transportation, 2015. **49**(7): p. 793-816.
40. Xu, H., et al., *Complementarity models for traffic equilibrium with ridesharing.* Transportation Research Part B: Methodological, 2015. **81**: p. 161-182.
41. Schuetz, P. and A. Caflisch, *Multistep greedy algorithm identifies community structure in real-world and computer-generated networks.* Physical Review E, 2008. **78**(2): p. 026112.
42. Wolfler Calvo, R., et al., *A distributed geographic information system for the daily car pooling problem.* Computers & Operations Research, 2004. **31**(13): p. 2263-2278.
43. Duan, R. and S. Pettie, *Linear-time approximation for maximum weight matching.* Journal of the ACM (JACM), 2014. **61**(1): p. 1-23.
44. Bei, X. and S. Zhang. *Algorithms for trip-vehicle assignment in ride-sharing*. in *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.
45. Chen, W., et al., *A Ride-Sharing Problem with Meeting Points and Return Restrictions.* Transportation Science, 2019. **53**(2): p. 401-426.
46. Wang, X., M. Dessouky, and F. Ordonez, *A pickup and delivery problem for ridesharing considering congestion.* Transportation Letters, 2016. **8**(5): p. 259-269.
47. IRS, *2021 Standard Mileage Rates*. 2020, International Foundation of Employee Benefit Plans.
48. Transparency Market Research, *Automative Market Research Report*. 2019.
49. Rayle, L., et al., *Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco.* Transport Policy, 2016. **45**: p. 168-178.
50. Shaheen, S.A., N.D. Chan, and T. Gaynor, *Casual carpooling in the San Francisco Bay Area: Understanding user characteristics, behaviors, and motivations.* Transport Policy, 2016. **51**: p. 165-173.
51. Huang, D., et al., *A two-phase optimization model for the demand-responsive customized bus network design.* Transportation Research Part C: Emerging Technologies, 2020. **111**: p. 1-21.
52. Lyu, Y., et al., *CB-Planner: A bus line planning framework for customized bus systems.* Transportation Research Part C: Emerging Technologies, 2019. **101**: p. 233-253.
53. Calvo, R.W. and A. Colorni, *An effective and fast heuristic for the Dial-a-Ride problem.* 4OR, 2007. **5**(1): p. 61-73.
54. Tong, L., et al., *Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing.* Transportation Research Part C: Emerging Technologies, 2017. **85**: p. 451-475.
55. Psaraftis, H.N., *A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem.* Transportation Science, 1980. **14**(2): p. 130-154.
56. Psaraftis, H.N., *An Exact Algorithm for the Single Vehicle Many-to-Many Dial-A-Ride Problem with Time Windows.* Transportation Science, 1983. **17**(3): p. 351-357.

57. Sexton, T.R. and L.D. Bodin, *Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times: I. Scheduling.* Transportation Science, 1985. **19**(4): p. 378-410.
58. Sexton, T.R. and L.D. Bodin, *Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times: II. Routing.* Transportation Science, 1985. **19**(4): p. 411-435.
59. Dumas, Y., J. Desrosiers, and F. Soumis, *The pickup and delivery problem with time windows.* European Journal of Operational Research, 1991. **54**(1): p. 7-22.
60. Baldacci, R., E. Bartolini, and A. Mingozzi, *An Exact Algorithm for the Pickup and Delivery Problem with Time Windows.* Operations Research, 2011. **59**(2): p. 414-426.
61. Bettinelli, A., A. Ceselli, and G. Righini, *A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows.* Mathematical Programming Computation, 2014. **6**(2): p. 171-197.
62. Cherkesly, M., et al., *Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks.* European Journal of Operational Research, 2016. **250**(3): p. 782-793.
63. Cherkesly, M., G. Desaulniers, and G. Laporte, *Branch-Price-and-Cut Algorithms for the Pickup and Delivery Problem with Time Windows and Last-in-First-Out Loading.* Transportation Science, 2015. **49**(4): p. 752-766.
64. Ghilas, V., E. Demir, and T. Van Woensel, *An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines.* Computers & Operations Research, 2016. **72**: p. 12-30.
65. Lee, Y.-J. and A. Nickkar, *Optimal Automated Demand Responsive Feeder Transit Operation and Its Impact.* 2018.
66. Lu, Q. and M. Dessouky, *An Exact Algorithm for the Multiple Vehicle Pickup and Delivery Problem.* Transportation Science, 2004. **38**(4): p. 503-514.
67. Ropke, S. and J.-F. Cordeau, *Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows.* Transportation Science, 2009. **43**(3): p. 267-286.
68. Torkjazi, M. and N. Huynh, *Effectiveness of Dynamic Insertion Scheduling Strategy for Demand-Responsive Paratransit Vehicles Using Agent-Based Simulation.* Sustainability, 2019. **11**(19): p. 5391.
69. van Engelen, M., et al., *Enhancing flexible transport services with demand-anticipatory insertion heuristics.* Transportation Research Part E: Logistics and Transportation Review, 2018. **110**: p. 110-121.
70. Gong, M., et al., *Transfer-based customized modular bus system design with passenger-route assignment optimization.* Transportation Research Part E: Logistics and Transportation Review, 2021. **153**: p. 102422.
71. Mahéo, A., P. Kilby, and P.V. Hentenryck, *Benders Decomposition for the Design of a Hub and Shuttle Public Transit System.* Transportation Science, 2019. **53**(1): p. 77-88.
72. Guo, R., W. Guan, and W. Zhang, *Route Design Problem of Customized Buses: Mixed Integer Programming Model and Case Study.* Journal of Transportation Engineering, Part A: Systems, 2018. **144**(11): p. 04018069.
73. Wang, H., *Routing and Scheduling for a Last-Mile Transportation System.* Transportation Science, 2019. **53**(1): p. 131-147.
74. Wu, J., et al., *A modular, adaptive, and autonomous transit system (MAATS): A in-motion transfer strategy and performance evaluation in urban grid transit networks.* Transportation Research Part A: Policy and Practice, 2021. **151**: p. 81-98.

75. Chen, X., et al., *Customized bus route design with pickup and delivery and time windows: Model, case study and comparative analysis.* Expert Systems with Applications, 2021. **168**: p. 114242.
76. Dou, X., Q. Meng, and K. Liu, *Customized bus service design for uncertain commuting travel demand.* Transportmetrica A: Transport Science, 2021. **17**(4): p. 1405-1430.
77. Wang, L., et al., *Integrating Passenger Incentives to Optimize Routing for Demand-Responsive Customized Bus Systems.* IEEE Access, 2021. **9**: p. 21507-21521.
78. Ma, H., M. Yang, and X. Li, *Integrated optimization of customized bus routes and timetables with consideration of holding control.* Computers & Industrial Engineering, 2023. **175**: p. 108886.
79. Litman, T., *Evaluating Public Transit Benefits and Cost*, in *Best Practices Guidebook*. 2018.
80. Victoria Transport Policy Institute, *Transportation Cost and Benefit Analysis II – Travel Time Costs* 2020.