Title

Dynamic Routing for Ride-Sharing

Authors

Dessouky, Maged M
Hu, Shichun

Publication Date

2021-11-01

DOI

10.7922/G2D798QK

Data Availability

The data associated with this publication are available at:
https://doi.org/10.7910/DVN/P6ROMD

Dynamic Routing for Ride-Sharing

November 2021

A Research Report from the National Center for Sustainable Transportation

Maged M. Dessouky, University of Southern California

Shichun Hu, University of Southern California

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. NCST-USC-RR-21-22	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A
4. Title and Subtitle Dynamic Routing for Ride-Sharing		**5. Report Date** November 2021
		6. Performing Organization Code N/A
7. Author(s) Maged M. Dessouky, Ph.D., https://orcid.org/0000-0002-9630-6201 Shichun Hu, Ph.D., https://orcid.org/0000-0001-9185-2599		**8. Performing Organization Report No.** N/A
9. Performing Organization Name and Address University of Southern California METRANS Transportation Center University Park Campus, VKC 367 MC:0626 Los Angeles, California 90089-0626		**10. Work Unit No.** N/A
		11. Contract or Grant No. USDOT Grant 69A3551747114
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE, Washington, DC 20590		**13. Type of Report and Period Covered** Final Report (August 2020 – August 2021)
		14. Sponsoring Agency Code USDOT OST-R

15. Supplementary Notes
DOI: https://doi.org/10.7922/G2D798QK
Dataset DOI: https://doi.org/10.7910/DVN/P6ROMD

16. Abstract
The research report explored the use of High Occupancy Vehicle (HOV) lanes and meeting points in a ride-sharing system where drivers have their own origin and destination. A two-stage heuristic algorithm is proposed, which consists of an insertion heuristic to solve the pickup and delivery problem (PDP) problem and a second stage algorithm that can solve the meeting points problem optimally in polynomial time. The experimental results show that the HOV lanes and meeting points can increase the efficiency of a dynamic ride-sharing system.

17. Key Words Ride-Sharing, HOV Lane, Meeting Points	**18. Distribution Statement** No restrictions.

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 35	**22. Price** N/A

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

About the National Center for Sustainable Transportation

The National Center for Sustainable Transportation is a consortium of leading universities committed to advancing an environmentally sustainable transportation system through cutting-edge research, direct policy engagement, and education of our future leaders. Consortium members include: University of California, Davis; University of California, Riverside; University of Southern California; California State University, Long Beach; Georgia Institute of Technology; and University of Vermont. More information can be found at: ncst.ucdavis.edu.

Disclaimer

Acknowledgments

This study was funded, partially or entirely, by a grant from the National Center for Sustainable Transportation (NCST), supported by the U.S. Department of Transportation (USDOT) through the University Transportation Centers program. The authors would like to thank the NCST and the USDOT for their support of university-based research in transportation, and especially for the funding provided in support of this project.

NCST

Dynamic Routing for Ride-Sharing

A National Center for Sustainable Transportation Research Report

November 2021

Maged M. Dessouky, Department Chair, Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California

Shichun Hu, Ph.D., Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California

[page intentionally left blank]

TABLE OF CONTENTS

List of Figures

Dynamic Routing for Ride-Sharing

EXECUTIVE SUMMARY

Traffic congestion is considered to be a major source of greenhouse gas emissions and has become one of the causes for significant economic costs, wasted time and public health risks (Levy et al. 2010; Pi et al. 2021; Schrank et al. 2019). In order to reduce the negative impact of traffic congestion, people have striven to find different methods to tackle this problem. Ride-sharing, defined as a joint-trip of more than two participants that share a vehicle and requires coordination with respect to itineraries, has the potential to help mitigate congestion (Furuhata et al.2013). Although this idea dates back to the 1940s, it is only until recent development of internet, global-positioning-systems (GPS) and wireless communications that ride-sharing can fully realize its potential.

A good ride-sharing system should provide quick response to passenger requests while optimizing the routes which is not an easy task especially in the situation when passengers request dynamically. One way to mitigate the effect of uncertainty is to allow passengers to walk while waiting for the drivers. At the same time, we would like to fully utilize the incentives of ride-sharing provided by the government agencies: the increasing use of High Occupancy Vehicle (HOV) lanes. Therefore, in this report, we study and formulate the dynamic pickup and delivery problem with HOV lanes and meeting points with application to ride-sharing.

In our ride-sharing context, the drivers are travelling toward their own destinations and can make detours to pick up or drop off additional passengers where the passengers have flexible pickup and drop-off locations. We propose a two-stage heuristic algorithm which consists of an insertion heuristic to solve the pickup and delivery problem (PDP) and a second stage algorithm that can solve the meeting points problem optimally in polynomial time.

Our experimental results show that both the HOV lanes and meeting points can increase the efficiency of a dynamic ride-sharing system. A good combination of HOV lanes and meeting points can provide passengers with lower commuting cost and faster commuting experience.

1 Introduction

Traffic congestion, as prevalent as it may seem in everyday life, has become an increasingly important issue impeding the social development in modern societies (Pi et al. 2021). It is one of the causes for significant economic costs, wasted time and public health risks. According to the 2021 Urban Mobility Report by Schrank et al. (2021), the total cost of congestion in 2019 was $190 billion in the U.S. and the total amount of delayed time was 8.7 billion hours with an extra usage of 3.5 billion gallons of fuel. As a comparison, these measures were $101 billion, 4.3 billion and 1.7 billion in 2020 when the government ordered stay-at-home restrictions due to the pandemic. These two sets of data can best show how the decrease in number of vehicles on the road can alleviate the traffic congestion problem and how less traffic congestion can result in significant reduction (around 50%) in social costs, and improvements in social efficiency and emissions. Moreover, the Harvard Center for Risk Analysis (HCRA) at the School of Public Health conducted a research study in 83 urban areas to evaluate the public health impacts of traffic congestion (Levy et al. 2010). The results indicated that traffic congestion led to 4,000 premature deaths with a public health cost of around $31 billion in 2000. If no effective methods are taken by 2030, it is projected that there will be 1,900 premature deaths and $17 billion in social costs annually.

Ride-sharing has the potential to help mitigate congestion, since 40% of traffic congestion (FHA and FTA 2013) is due to road bottleneck (inadequate physical capacity) and the average vehicle occupancy rate was 1.18 for work commute trips (McGuckin and Fucci 2017). “Ride-sharing is a joint-trip of more than two participants that share a vehicle and requires coordination with respect to itineraries” (Furuhata et al.2013). By taking advantage of the vacant seats in most passenger vehicles, ride-sharing could increase the efficiency of the transportation system, reduce traffic congestion, decrease fuel usage and mitigate pollution. A good ride-sharing system should provide automated matching which means that the system should help drivers and riders find suitable matches (Agatz et al. 2012). Recent technologies such as global-positioning-systems (GPS), wireless communication via satellite, cellular and paging networks, which enable 2-way communication with mobile fleets, and real-time information services make it possible to dynamically estimate travel times and route vehicles. In the past few years, there have been a plethora of apps such as UberPool and LyftPool that have developed technologies to help match drivers with passengers in real time. The matching between the drivers and passengers in ride-sharing can be viewed as a pickup and delivery problem (PDP). However, most of the developed techniques and models for PDPs assume known static data as their input, for instance, the customer demands, travel costs, and travel times are all known in advance. On the contrary, in the real world, operations in any transportation network contain a fairly high level of uncertainties including variable arrival of new service requests, request locations, cancellation of existing requests, unknown demand sizes, etc. One way to mitigate the effect of uncertainty is to allow passengers to walk while waiting for the drivers. The idea of applying this to ride-sharing services is not new. In fact, Uber announced Uber Express Pool in early 2018 (Stock 2018). However, how they determine the pickup and drop-off locations (i.e., the meeting points) is not well understood in the academic community. Thus, in this report, we study and formulate the dynamic pickup and delivery problem with meeting points with application to ride-sharing.

At the same time, we would like to fully utilize the incentivization of ride-sharing provided by the government agencies: the increasing use of High Occupancy Vehicle (HOV) lanes and a policy of reduced toll rates for high occupancy vehicles on many roads and bridges. For example, in Southern California a portion of the freeways waive the toll rate to vehicles which have two or more people. If a vehicle has the required number of people, then HOV lanes can be used to save travel time especially during peak hours. That is, there may be an incentive to take the detour to pick up additional passengers to qualify to ride on the HOV lanes or discounted toll rates. Therefore, ride-sharing could provide a cost reduction and time savings under congestion.

In this research, we will consider a dynamic PDP with meeting points considering HOV lanes with the objective of minimizing the total travel time. The rest of the report is organized as follows: Section 2 presents a literature of the PDP, dynamic PDP and related work. Section 3 mathematically formulates the model followed by our solution algorithms in Section 4. Then we run experiments in Section 5 and conclude in Section 6.

2 Literature Review

The dynamic ride-sharing problem we study in this project is similar to the dynamic dial-a-ride problem (DARP) which is also known as a subset of the general dynamic pickup and delivery problem (PDP) when transporting goods instead of passengers. According to Berbeglia et al. (2010), the dynamic PDPs are divided into dynamic vehicle routing problems with pickup and delivery (Dynamic VRPPD), dynamic stacker crane problems (Dynamic SCPs) and dynamic dial-a-ride problems (Dynamic DARPs). They argue that dynamic DARPs are different from dynamic VRPPD because DARPs generally have more constraints such as tigher time windows and maximum ride times.

We take the categorization methods by Ho et al. (2018) and Pillac et al. (2013); that is, the general PDP topic is categorized into static and dynamic problems, deterministic and stochastic problems or a mix-match. Most DARPs study the static and deterministic version where decisions are made before the operation starts and the information is known with certainty at the time of the decision. As a result, papers in this category either focus on algorithmic improvement or consider different problem features inspired by real life applications (e.g., Braekers et al. 2014; Posada et al. 2017; Zhang et al. 2015).

The papers that took the algorithmic improvement approach can be further categorized into two categories, those focusing on developing exact methods and the other on heuristics. Lu and Dessouky (2004) formulated the problem as a 0-1 integer-programming problem, and a branch-and-cut algorithm is used to optimally solve the problem. Cordeau (2006) and Ropke et al. (2007) provided an alternative formulation and applied a branch-and-cut algorithm to optimally solve the problem which is outperformed by a new branch-and-cut-and-price algorithm in their later paper (Ropke and Cordeau 2009). More recently, Aziez et al. (2020) introduced two new formulations of the problem and managed to obtain optimality for 41 instances from existing benchmark problem sets that contain up to 100 nodes.

Even though researchers have improved the exact algorithms over the years to solve the PDP problem with more nodes, the size of the problem that can be solved optimally is rather small compared to actual size problems. Therefore, another solution approach is to develop heuristics and metaheuristics to effectively solve large instances. Construction insertion heuristics is one such approach. A parallel regret insertion heuristic was proposed by Diana and Dessouky (2004). Wong and Bell (2006) proposed a heuristic including parallel insertions, re-insertions and exchanges. Marković et al. (2015) introduced an insertion-based heuristic that accounts for operational requirements such as different passenger needs and different specifications of the objective function. Chassaing et al. (2016) proposed an evolutionary local search-based heuristic that incorporated a new greedy randomized heuristic for calculating the initial solution and a dynamic probability management mechanism to improve convergence in the local search.

Metaheuristics are also widely implemented in solving the PDP problem with large instances. Tabu search has been one of the most commonly used metaheuristics (Cordeau and Laporte 2005). The Tabu search heuristic developed by Cordeau and Laporte (2003) proposed a neighborhood evaluation procedure to minimize route duration and ride times. This early work in Tabu search has since inspired many recent researchers to extend their work to satisfy more complex and real-life constraints (Ho et al. 2018). Additionally, many other metaheuristics have been applied to the PDP. Sombuntham and Kachitvichayanukul (2010) used the particle swarm optimization algorithm for the PDP with multiple depots. Parragh et al. (2010) proposed a competitive variable neighborhood search-based heuristic for the static multi-vehicle DARP. Catay (2009) applied the ant colony optimization algorithm and Zidi et al. (2012) applied simulated annealing to solve the DARP with multiple objectives.

Among the papers that consider different problem features from real life applications, Braekers et al. (2014) studied the PDP with heterogeneous passengers, service operators that have a heterogeneous fleet and multiple depots. Zhang et al. (2015) addressed a patient transportation problem derived from real life. The difference of this problem from other PDPs is that transporting patients using ambulances imposes extra medical constraints such as the need to disinfect the ambulance at the end of each trip. Posada et al. (2017) investigated the problem concerning door-to-door transportation systems for the elderly and/or disabled. This system differs from normal PDPs in that these people with special needs require a more flexible system and the pickup and delivery trips among them are usually integrated with public transit systems that has fixed routes. In the dynamic and stochastic version of the PDPs, uncertainties exist at the time of the operation and in the passengers' information. Therefore, despite the fact that there is a growing number of papers studying this version, the research is not as much as the static and deterministic version.

We next review the literature in dynamic PDPs.

2.1 Dynamic PDPs

According to Berbeglia et al. (2010), Psaraftis (1988) represented one of the early studies on dynamic PDP. He considered a single vehicle problem with the objective to minimize a weighted function of total service time and passenger dissatisfaction. The solution approach was based on first establish an algorithm for the static case and then adapt it to the dynamic case where the author used re-optimization every time a new request arrives. This method of addressing the dynamism in the problems is popular among the literature. These papers all consider new requests as a trigger to the re-optimization of the current solution which is often constructed statically at the beginning of the operation. The difference lies in the methods they use to search for better solutions as new requests come in. Attanasio et al. (2004) developed a parallel algorithm which first constructed a static solution based on known requests and then used an insertion algorithm and tabu search to reoptimize the current solution whenever there is a new request. Coslovich et al. (2006) proposed an algorithm that maintains a solution repository. The algorithm then chooses from the repository to insert new requests. Häll and Peterson (2013) assessed the power of ruin and recreate methods in the re-optimization phase of a dynamic PDP. They found that ruin methods based on removal of sequences of requests can obtain better solution quality. Additional to the efforts in developing a better route search algorithm, other research focus on tuning the re-optimization frequency during the planning horizon to find better solutions (Agatz et al. 2011; Kleiner et al. 2011; Zou 2017). In this approach, one separates the time horizon into small time segments and re-optimizes at the end of each segment. To find the best solution, researchers tune the length of each time segment and adjust the wait time a driver can spend after picking up passengers. Instead of working on different route searching algorithms and re-optimization frequencies, Sayarshad and Chow (2015) incorporate pricing decisions in determining how a new request is accommodated.

Note that all the above papers studying the dynamic PDP in a deterministic environment where information of requests and the system are known at the time of the decision. However, the reality often contains extra complexities such as traffic jam, vehicle breakdowns, request cancellations and passenger no-shows. This new stream of dynamic and stochastic PDPs has attracted more researchers despite their complexities. Xiang et al. (2008) studied this problem and proposed a fast heuristic to re-optimize the route. This heuristic consists of a local search strategy and a secondary objective function to drive the search out of local optima. Schilde et al. (2011) studied a dynamic and stochastic PDP where outbound trips can possibly trigger inbound trips. They proposed four different metaheuristics and their results indicated that using the stochastic information on return transports leads to an average improvement of around 15%. Núñez et al. (2014) used a multi-objective model predictive control method to predict the future scenarios to obtain optimal control in a stochastic environment. Most recently, Ulmer et al. (2020) considered a restaurant meal delivery problem with random ready times. They proposed a route-based Markov decision process (MDP) to model the problem and an anticipatory customer assignment policy to address the stochasticity.

2.2 Ride-Sharing with Walking

In the previous sections of the literature review, each passenger (customer) is picked up and dropped off at their exact locations. In this report, we study the case where passengers are provided with the option of walking to a pickup or drop-off location.

In public transport, we are used to the idea of walking since a public transit system usually has a fixed transportation route and utilizes the benefits of massive transit (Mohring 1972; Fielbaum et al. 2020). It is a well-studied topic in public transport (Hurdle 1973; Chang and Schonfeld 1991; Tirachini 2014; Lyu et al. 2019; Pei et al. 2019) and can provide some intuition in that there are many good solution approaches to jointly optimize the costs of service providers and passengers. In the context of private transportation services, we encounter walking scenarios as well: (1) when passengers request a ride in a limited access area such as universities, and (2) when passengers try to find more taxis at a larger intersection. A study by Nie (2017) showed that taxi services that require walking achieves higher efficiency than those that do not under high-demand conditions. Furthermore, in the context of ride-sharing, according to Stiglic et al. (2015), the idea of applying walking in ride-sharing can improve the matching rate and mileage savings. A recent study by Papoutsis et al. (2021) also indicated that ride-sharing with exact locations is an obstacle to mass carpooling utilization. Yan et al. (2020) used Uber commercial data to show that by allowing passengers to walk, advantages such as increased capacity utilization and mitigated price variability can be observed under optimized dynamic pricing algorithms and dynamic waiting matching strategy. Therefore, there exists a need to incorporate walking into ride-sharing systems. Compared to ride-sharing with exact locations, there is not much literature investigating how to integrate walking in a ride-sharing system. Li et al. (2015) proposed a novel dynamic programming method to search for the optimal route with multiple meeting points in an on-demand ride-sharing system. Li et al. (2018) formulated the problem as a mixed integer linear program (MILP) and proposed a Tabu-based meta-heuristic algorithm. Specifically, their results show that introducing meeting points to ride-sharing system saves the total travel time by 2.7%-3.8% for a small-scale ride-sharing system. Zhao et al. (2018) used a space-time network representation on ride-sharing problems with flexible pickup and drop-off locations and proposed a Lagrangian relaxation inspired solution approach. Smet (2021) studied the same problem, but they also consider the problem of how to decide which users act as drivers in a ride-sharing system.

3 Problem Description

In this section, we formally present the models. As previously mentioned, we focus on the PDP with meeting points with specific application to ride-sharing. In our ride-sharing context, the drivers are travelling toward their own destinations and can make detours to pick up or drop off additional passengers where the passengers have flexible pickup and drop-off locations. We also consider the utilization of high occupancy (HOV) lanes to maximize the savings in time brought by ride-sharing.

Although we are interested in solving the dynamic version of the problem, we first present a mathematical model of the static version to formalize our problem description. Furthermore, the decomposition of the static formulation is incorporated in our dynamic solution approach which is presented in Section 4.

Given a set of passengers $\mathbb{P} = \{1, \dots, n\}$ and a set of drivers $\mathbb{V} = \{1, \dots, m\}$, each passenger $p \in \mathbb{P}$ (driver $v \in \mathbb{V}$) has an origin $O_p(O_v)$, a destination $D_p(D_v)$. Each driver has a maximum in-vehicle time H_v while each passenger has a maximum walking distance L_p to the deviated origin and destination and a maximum wait time I_p before the vehicle arrives. The passengers share the same walking speed of W. We need to output the deviated origin $\left(O_p^d\right)$ of a passenger's request and the deviated destination $\left(D_p^{\mathrm{d}}\right)$ while satisfying all the constraints provided by the passengers' information. The objective is to minimize the total travel time.

3.1 Model

We first create a network $G(\mathbb{N}, \mathbb{A})$ with n passengers and m drivers. The node set $\mathbb{N} = \{1, \dots, 2n + 2m\} = \mathbb{O}_p \cup \mathbb{D}_p \cup \mathbb{O}_v \cup \mathbb{D}_v$ where $\mathbb{O}_p = \{1, \dots, n\}$, $\mathbb{D}_p = \{n+1, \dots, 2n\}$, $\mathbb{O}_v = \{2n+1, \dots, 2n+m\}$ and $\mathbb{D}_v = \{2n+m+1, \dots, 2n+2m\}$. The arc set $\mathbb{A} = \{A_{i,j} | i, j \in N\}$. Note that the incorporation of HOV lanes may generate multiple arcs between two nodes. For simplicity, since HOV lanes are always chosen when possible (due to less travel time), instead of representing them with multiple arcs, we will introduce a constraint to indicate whether the travel time associated with HOV lanes are valid. The deviated locations will not be included in the network graph, no arcs connected to them, but they are among the decision variables. Let $c_{i,j}$ denote the travel time between node i and j, $d_{i,j}$ denote the distance between node i and j, $\beta_{i,j}$ denote the time discount factor between node i and j when HOV lane is chosen, and H denotes the number of people required to go on a HOV lane. (r_i^x, r_i^y) denotes the coordinates of node $i \in \mathbb{N}$ where (l_i^x, l_i^y) denotes the coordinates of the deviated locations of node $i \in \mathbb{O}_p \cup \mathbb{D}_p$. Both sets of coordinates are in $\mathbb{R}^2$ space. $L_i, i \in \mathbb{O}_p \cup \mathbb{D}_p$ denote the maximum walking distance of each passenger to the deviated locations, E denotes the average driving speed and U denotes the capacity for each vehicle (i.e., the maximum number of passengers in a vehicle). We also denote $g_{i,v}$ as the load indicator to show whether a passenger is picked up or delivered:

$$g_{i,v} = \begin{cases} 1, & \text{if } i \in \mathbb{O}_p \\ -1, & \text{if } i \in \mathbb{D}_p \\ 0, & \text{otherwise.} \end{cases}$$

The formulation of this PDP is below. The three sets of decision variables are:

$$y_{i,j,v} = \begin{cases} 1, & \text{if vehicle } v \text{ travels from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$
$$b_{i,j,v} = \begin{cases} 1, & \text{if node } i \text{ is visited before node } j \text{ on vehicle } v \\ 0, & \text{otherwise} \end{cases}$$
$$a_{i,j,v} = \begin{cases} 1, & \text{if HOV lane is valid for vehicle } v \text{ from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$
$$\left(l_i^x, l_i^y\right) \quad \forall i \in \mathbb{O}_p \cup \mathbb{D}_p$$

The mathematical formulation is then:

$$\min \sum_{v\in V}\sum_{i\in N}\sum_{j\in N}(1-\alpha_{i,j,v}\beta_{i,j})c_{i,j}y_{i,j,v} + M\sum_{j\in O_p}\left(1-\sum_{v\in V}\sum_{i\in N}y_{i,j,v}\right)$$

$$\text{s.t.} \sum_{v\in V}\sum_{j\in N} y_{i,j,v} \leq 1 \quad \forall i \in \mathbb{N}\backslash\mathbb{D}_v \tag{1}$$

$$\sum_{v\in V}\sum_{i\in N} y_{i,j,v} \leq 1 \quad \forall j \in \mathbb{N}\backslash\mathbb{O}_v \tag{2}$$

$$\sum_{j\in N} y_{i,j,v} = \sum_{j\in N} y_{j,i,v} \quad \forall i \in \mathbb{O}_p \cup \mathbb{D}_p, v \in \mathbb{V} \tag{3}$$

$$\sum_{j\in N} y_{i,j,i-2n} = \sum_{j\in N} y_{j,i+m,i-2n} \quad \forall i \in \mathbb{O}_v \tag{4}$$

$$b_{k,i,v} \leq b_{k,j,v} + (1-y_{i,j,v}) \quad \forall i \in \mathbb{N}\backslash\mathbb{D}_v, j \in \mathbb{N}\backslash\mathbb{O}_v, k \in \mathbb{N}\backslash\{i\} \text{ and } v \in \mathbb{V} \tag{5}$$

$$b_{k,j,v} \leq b_{k,i,v} + (1-y_{i,j,v}) \quad \forall i \in \mathbb{N}\backslash\mathbb{D}_v, j \in \mathbb{N}\backslash\mathbb{O}_v, k \in \mathbb{N}\backslash\{i\} \text{ and } v \in \mathbb{V} \tag{6}$$

$$y_{i,j,v} \leq b_{i,j,v} \quad \forall A_{i,j} \in A, v \in \mathbb{V} \tag{7}$$

$$b_{i,i,v} = b_{i,k,v} = 0 \quad \forall i \in \mathbb{N}, k \in \mathbb{O}_v \text{ and } v \in \mathbb{V} \tag{8}$$

$$b_{j,i,v} = 0 \quad \forall v \in \mathbb{V}, i \in \mathbb{O}_p \text{ and } j = i+n, i \in \mathbb{O}_v \text{ and } j = i+m \tag{9}$$

$$b_{i,j,v} = 1 \quad \forall v \in \mathbb{V}, i \in \mathbb{O}_v \text{ and } j = i+m \tag{10}$$

$$\sum_{v\in V} b_{i,j,v} = 1 \quad i \in \mathbb{O}_p \text{ and } j = i+n \tag{11}$$

$$b_{i,k,v} = b_{i+n,k,v} \quad \forall v \in \mathbb{V}, i \in \mathbb{O}_p, k \in \mathbb{D}_v \tag{12}$$

$$g_{i,v} + \sum_{i\in N} b_{i,j,v}g_{i,v} \leq U \quad \forall j \in \mathbb{N}, v \in \mathbb{V} \tag{13}$$

$$(l_i^x - r_i^x)^2 + \left(l_i^y - r_i^y\right)^2 \leq L_i^2 \quad \forall i \in \mathbb{O}_p \cup \mathbb{D}_p \tag{14}$$

$$\left(l_i^x - l_j^x\right)^2 + \left(l_i^y - l_j^y\right)^2 = d_{i,j}^2 \quad \forall i,j \in \mathbb{O}_p \cup \mathbb{D}_p \tag{15}$$

$$\left(r_i^x - l_j^x\right)^2 + \left(r_i^y - l_j^y\right)^2 = d_{i,j}^2 \quad \forall i \in \mathbb{O}_v, j \in \mathbb{O}_p \tag{16}$$

$$\left(l_i^x - r_j^x\right)^2 + \left(l_i^y - r_j^y\right)^2 = d_{i,j}^2 \quad \forall i \in \mathbb{D}_p, j \in \mathbb{D}_v \tag{17}$$

$$c_{i,j}E = d_{i,j} \quad \forall A_{i,j} \in \mathbb{A} \tag{18}$$

$$\sum_{k \in N} b_{k,i,v} g_{k,v} \geq H - M(1 - \alpha_{i,j,v}) \quad \forall i, j \in \mathbb{N}, v \in \mathbb{V} \tag{19}$$

$$\sum_{i \in N} \sum_{j \in N} (1 - \alpha_{i,j,v} \beta_{i,j}) c_{i,j} y_{i,j,v} \leq H_v \quad \forall v \in \mathbb{V} \tag{20}$$

$$\alpha_{i,j,v} = 0 \; or \; 1 \tag{21}$$

$$y_{i,j,v} = 0 \; or \; 1 \tag{22}$$

$$b_{i,j,v} = 0 \; or \; 1 \tag{23}$$

The objective is to minimize the total travel cost (first term) plus the minimization of unserved passengers (second term) where M is a weighting factor. M is set to a large number when it is desired to serve as many requests as possible and for that solution to minimize the travel cost.

Constraint sets (1) and (2) are network flow constraints imposing that one passenger is served by one driver or no driver. Constraint set (3) ensures that the origin and destination of a passenger be assigned to the same driver. Constraint set (4) ensures that the origin and destination of a driver is assigned to the same driver. Constraint sets (5) and (6) ensure that if node i is immediately before node j ($y_{i,j,v} = 1$), then we have $b_{k,i,v} = b_{k,j,v}$ for all $k \in \mathbb{N}\backslash\{i\}, v \in \mathbb{V}$. Similarly, constraint set (7) enforces that if $y_{i,j,v} = 1$, $b_{i,j,v} = 1$ and if $b_{i,j,v} = 0$, $y_{i,j,v} = 0$. Constraint sets (8) - (12) are prior constraints that enforce the deviated origins to be ahead of the deviated destinations. They also enforce the drivers' origins are ahead of their corresponding destinations. Constraint set (13) is the capacity constraint. Constraint set (14) ensures that the deviated locations are within the passengers' walking ranges. Constraint sets (15) - (17) describe how the actual distance costs between nodes are calculated; that is, even though we determine the pickup and delivery sequence based on node sets $\mathbb{O}_p$ and $\mathbb{D}_p$, we calculate the distances using the deviated locations. Constraint set (18) describes the relationship between distance and time to travel from node i to node j. Note if HOV lane is eligible to be taken from node i to node j then a discount is applied in the objective function. Constraint set (19) ensures the time discount factor for an arc is activated only when the HOV eligibility threshold is reached. Constraint set (20) ensures that for all the vehicles, the time a vehicle spends in the operation does not exceed its maximum in-vehicle time H_v. Since constraint set (20) may result in certain passengers not being served by any vehicle, we add an extra term in the objective function to avoid the infeasibility of the formulation. Therefore, these set of constraints ensure that the objective function is based on the deviated locations that minimizes the total time costs while maximizing the number of passengers served in the system.

As we can see from the above formulation, the PDP and the location selection problem are simultaneously solved. The location selection problem itself is a non-linear problem which causes extra complexity on the NP-hard PDP. In order to solve these two problems, we propose to separately solve them. We delete constraint sets (11) - (15) from the above formulation and simply let $c_{i,j} = \sqrt{\left(r_i^x - r_j^x\right)^2 + \left(r_i^y - r_j^y\right)^2}\,/E$ for all$i, j \in \mathbb{N}$. As a result, we have a model dedicated to solving the multi-vehicle PDP. The output of this model is a list of vehicle routes with each passenger assigned to a vehicle. Then for each vehicle route $\mathbb{Z} = \{Z_1, \ldots, Z_{2q+2}\}$, it contains one driver and q passengers. Let the first node in the route be $Z_1 \in \mathbb{O}_v = \{2n + 1, \ldots, 2n + m\}$, the last node be $Z_{2q+2} \in \mathbb{D}_v = \{2n + m + 1, \ldots, 2n + 2m\}$ and the nodes in between $Z_i \in \mathbb{O}_p \cup \mathbb{D}_p = \{1, \ldots, 2n\}$ be the pickup and delivery locations of the q passengers. We then establish the following quadratic model:

$$\min \sum_{i=1}^{2q+1} d_{Z_i, Z_{i+1}}$$

$$\text{s.t. } \left(r_{Z_i}^x - l_{Z_i}^x\right)^2 + \left(r_{Z_i}^y - l_{Z_i}^y\right)^2 \le L_{Z_i}^2 \qquad \forall i = 2, \ldots, 2q + 1$$

$$\left(r_{Z_1}^x - l_{Z_2}^x\right)^2 + \left(r_{Z_1}^y - l_{Z_2}^y\right)^2 \le d_{Z_1, Z_2}^2$$

$$\left(r_{Z_{2q+2}}^x - l_{Z_{2q+1}}^x\right)^2 + \left(r_{Z_{2q+2}}^y - l_{Z_{2q+1}}^y\right)^2 \le d_{Z_{2q+1}, Z_{2q+2}}^2$$

$$\left(l_{Z_{i+1}}^x - l_{Z_i}^x\right)^2 + \left(l_{Z_i+1}^y - l_{Z_i}^y\right)^2 \le d_{Z_i, Z_{i+1}}^2 \qquad \forall i = 2, \ldots, 2q$$

$$d_{Z_i, Z_{i+1}} \ge 0 \qquad \forall i = 1, 2q + 2$$

4 Dynamic Solution Algorithm

In this section, we describe our solution approach for solving the dynamic version of the above static problem. In the dynamic version, instead of requests being known at the beginning of the day, the requests arrive dynamically throughout the day as well as the driver's departure time. Figure 1 describes how each new passenger request is dealt with from a high-level point of view. Once a request is received by the ride-sharing system, it first generates a feasible set of vehicles based on the personal preferences of the request. That is, starting at the vehicle's current location, if it fails to reach any point within the maximum walking circle of the request location, then this vehicle is not feasible for this new request. This procedure provides a basic filter for assigning requests to vehicles. If the feasible set turns out to be empty, the system rejects the request. Otherwise, for each vehicle in the feasible set, the system calls on our routing algorithm to calculate a route and meeting points for this new request. That is, it inserts the request in the current route of the vehicle and determines the meeting points with minimal increase in travel time for that vehicle. During this procedure, if an insertion to a certain vehicle will cause any violation of the driver's maximum detour time T_v or the maximum waiting time I_p of an existing request on that vehicle, then this vehicle is removed from the feasible vehicle set. After this procedure, if the set is empty, the request is rejected. Otherwise, the request is accepted with the route and meeting points that has the minimal increase in travel time among all the feasible vehicles. Next, we introduce our algorithms in greater detail. We first present our routing algorithm and then the incorporated location selection algorithm that is used to generate the meeting points given a route.

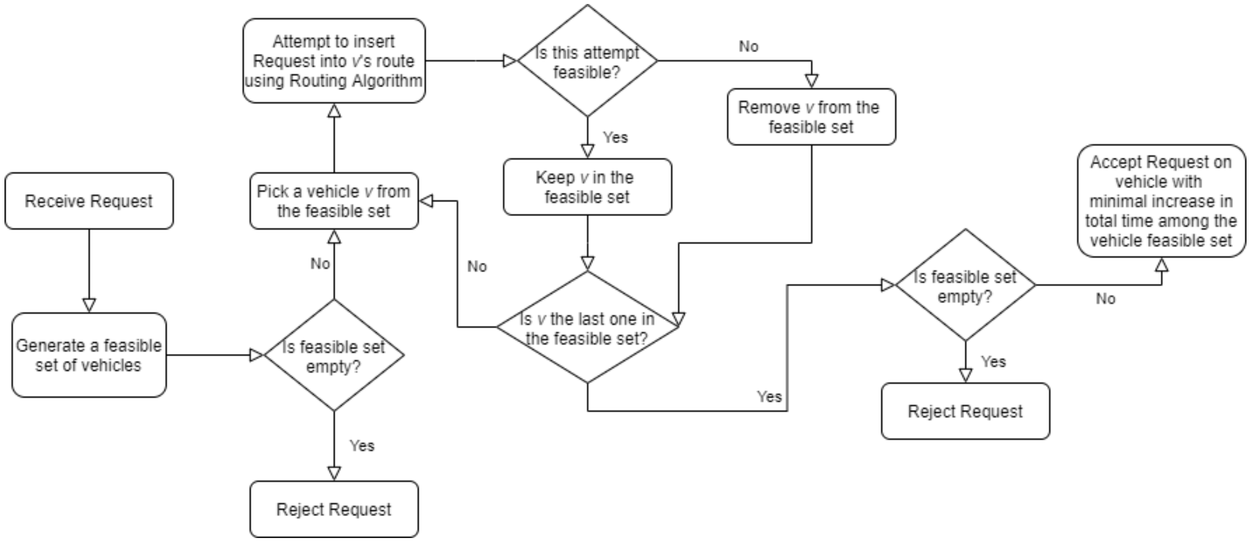


Figure 1. The Overall Solution Framework

4.1 The Routing Algorithm

In this section, we introduce how we match the passengers with the drivers and route them. We use an insertion algorithm to determine the ordering of passengers in the route. We use an insertion procedure because it is shown to be fast and effective in solving dynamic routing problems (Berbeglia et al. 2010; Pillac et al. 2013; Ho et al. 2018).

In order to introduce our algorithm in detail, we first introduce some extra notation. We denote $\mathbb{V}_{p,t}^{\mathrm{F}}$ as the feasible set of drivers for passenger p at time t. We also denote $\mathbb{P}_{v,t}$ as the set of passengers assigned to driver v at time t, $\mathbb{R}_{v,t}$ as the current route of driver v at time t which contains a sequence of deviated locations O_p^{d} and D_p^{d} for $p \in \mathbb{P}_{v,t}$. Then, we describe how a new request from passenger p is routed in Algorithm 1.

First, a feasible set $\mathbb{V}_{p,t}^{\mathrm{F}}$ is first created by checking all vehicles to see if they can reach any point on $\odot O_p$ (a circle with a radius of L_p centered at O_p) within maximum waiting time I_p. Immediately after a passenger p submits a request at time t (see Algorithm 1), passenger p will go through each vehicle $v \in \mathbb{V}_{p,t}^{\mathrm{F}}$ to see if it can be inserted. For each vehicle$v$, the algorithm tries to insert passenger p by first checking the capacity constraint. After that, temporary routes are generated with meeting points calculated as well (the next section describes Algorithm 2 to compute the meeting points). Then, for each temporary route, it checks the maximum in-vehicle time constraint for the driver and the maximum wait time constraint for the passengers in $\mathbb{P}_{v,t}$ and passenger p as well. Lastly, if a temporary route survives all feasibility checks, it is then added to the potential route set Φ. If Φ turns out to be empty, then vehicle v will not be added to $\mathbb{V}_{p,t+1}^{\mathrm{F}}$. Otherwise, not only v is added to $\mathbb{V}_{p,t+1}^{\mathrm{F}}$, but also the corresponding $\mathbb{R}_{v,t+1}$ and $\mathbb{P}_{v,t+1}$ are updated. Once all the vehicles in $\mathbb{V}_{p,t}^{\mathrm{F}}$ are checked, we compare the vehicles in

$\mathbb{V}^{\mathrm{F}}_{p,t+1}$ and find the v whose corresponding route $\mathbb{R}_{v,t+1}$ has the minimal increase in travel time and assign p to that v.

Algorithm 1: The Routing Algorithm

Input : Passenger p's information, O_p, D_p, L_p and I_p
The feasible set $\mathbb{V}^{\mathsf{F}}_{p,t}$ and $\mathbb{R}_{v,t}, \mathbb{P}_{v,t}$ corresponding to $v \in \mathbb{V}^{\mathsf{F}}_{p,t}$
Output: a vehicle v^{min} that passenger p is assigned to and the $\mathbb{R}_{v^{\mathsf{min}},t+1}, \mathbb{P}_{v^{\mathsf{min}},t+1}$ corresponding to this v^{min}

$\mathbb{V}^{\mathsf{F}}_{p,t+1} = \emptyset$
for $v \in \mathbb{V}^{\mathsf{F}}_{p,t}$ **do**
  create an empty set Φ to save potential routes
  for $i = 1, \ldots, |\mathbb{R}_{v,t}| + 1$ **do**
    insert O_p to become the i^{th} in $\mathbb{R}_{v,t}$, generating $\mathbb{R}'_{v,t}$
    if *the insertion will cause capacity U be violated at any point* **then**
      continue
    else
      for $j = i, \ldots, |\mathbb{R}'_{v,t}| + 1$ **do**
        insert D_p to become the j^{th} in $\mathbb{R}'_{v,t}$, generating $\mathbb{R}''_{v,t}$
        calculate O^{d}_p and D^{d}_p using Algorithm 2 and replace O_p, D_p in $\mathbb{R}''_{v,t}$
        if *H_v is violated or any I_p for $p \in \mathbb{P}_{v,t}$ is violated* **then**
          continue
        else
          add $\mathbb{R}''_{v,t}$ to set Φ
  if $\Phi \neq \emptyset$ **then**
    add v to $\mathbb{V}^{\mathsf{F}}_{p,t+1}$
    add p to $\mathbb{P}_{v,t+1}$
    $\mathbb{R}_{v,t+1}$ = the route in Φ with minimal increase in travel time
  else
    reject request q
Let $v^{\mathsf{min}} \in \mathbb{V}^{\mathsf{F}}_{p,t+1}$ be the vehicle with the minimal increase in travel time, assign p to v^{min} and output $\mathbb{R}_{v^{\mathsf{min}},t+1}$ and $\mathbb{P}_{v^{\mathsf{min}},t+1}$

4.2 The Location Selecting Algorithm

The location selection algorithm determines the meeting points (deviated points) given a route ordering for a vehicle.

In order to solve the location selection problem, we first acknowledge that the problem is equivalent to this following problem: given two fixed points, O and D, given $2n$ circles $\{\odot 1, \ldots, \odot 2n\}$ each with its center and radius, and the sequence of connecting the circles, how to determine the $2n$ points, each within its corresponding circle (including the boundary), such that the total distance connecting O to the circles in the given sequence and then to D is the shortest?

Let's first take a look at the one circle case. As shown in Figure 2, we want to find a point P within $\odot A$ (including the boundary) such that the length of $|OP| + |PD|$ is the shortest. It is trivial to see that when l_{OD} intersects with $\odot A$, point P is any point on the line segment of l_{OD} that is within $\odot A$ (this includes the scenario where l_{OD} is tangent to $\odot A$). Since there may be multiple points, we could generally provide a single point solution for these two scenarios: point P is the projection of point A onto l_{OD}. If l_{OD} does not intersect with $\odot A$, then point P is a point such that l_{AP} is the bisector of $\angle OPD$.

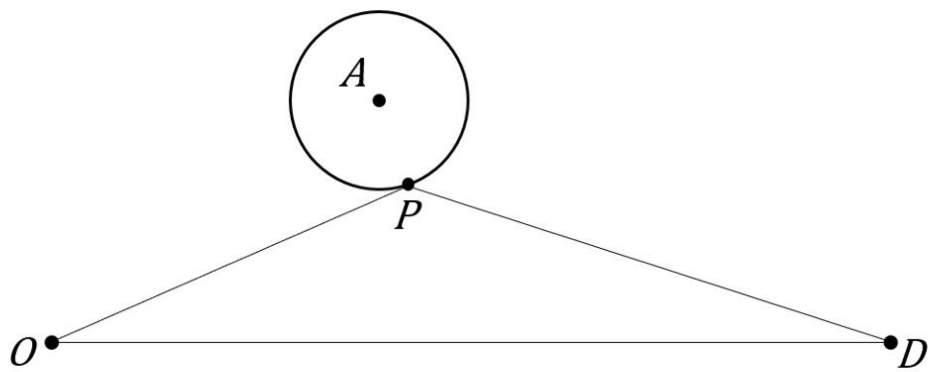


Figure 2. The One Circle Case

In fact, the latter description generalizes all the scenarios and the following propositions describes how to determine point P.

Proposition 1. Given two fixed points O and D, and $\odot A$, the optimal point P within $\odot A$ that minimizes $|OP| + |PD|$ is the point on A such that l_{AP} is the bisector of $\angle OPD$.

Proof. When l_{OD} intersects with $\odot A$, P is on l_{OD} because the distance among three points is the shortest when they are on the same line. When projecting A onto l_{OD}, we have $l_{AP} \perp l_{OD}$ and $\angle OPD = \pi$. Therefore, P is indeed the point on A where l_{AP} is the bisector of $\angle OPD$.

When l_{OD} does not intersect with $\odot A$, we know that the distance of any point on an ellipse to the ellipse's two foci is a constant. As shown in Figure 3, suppose O and D are the foci of an ellipse and suppose that P is on that ellipse, then we have $|OP| + |PD| = 2a$ where a is a constant. We can see that $2a$ is minimized when P is on l_{OD} and that $2a = |OD|$. However, when $2a$ is minimized, P is not on $\odot A$. Therefore, we increase $2a$ until P is on $\odot A$. As a result, we have an ellipse that is tangent to $\odot A$ and has O and D as its foci.

Since for an ellipse, the path for a light beam starting at one focus will always travel through a point P on the ellipse boundary and then reach the other focus, by Fermat's Law, this path is the shortest path from O to P and then to$D$. We also know that $\angle 1 = \angle 2$. Therefore, l_{AP} is the bisector of $\angle OPD$. ■

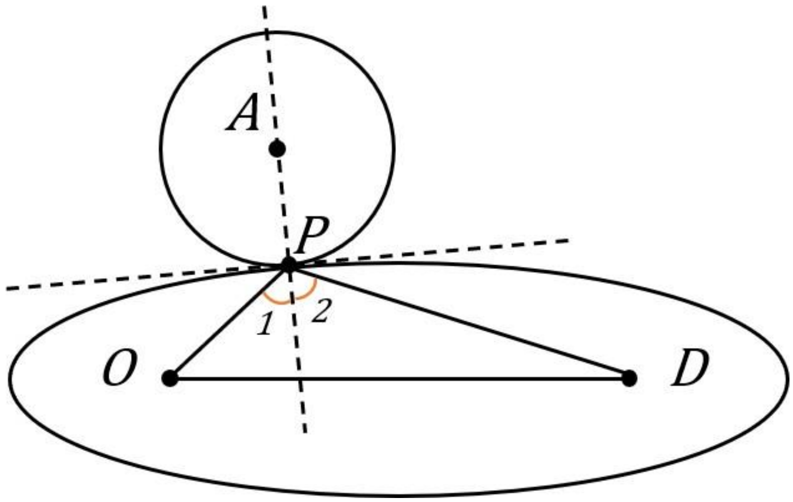


Figure 3. Ellipse Tangent to Circle A

We now have arrived at the geometrical property of point P. Additionally, based on this property, an algebraic solution approach can be found according to Eberly (2008) (referred as the Eberly Algorithm for the rest of the report). The problem of finding P is reduced to solving a unary quadratic equation whose solution is well known and is in closed-form. Therefore, in the one circle case, the optimal meeting point P can be found within $O(1)$ time complexity.

Next, we show how this one circle case can be extended to the case of multiple circles in our solution approach. We first introduce how it is applied to the two-circle case and then the $2q$-circle case where q is the number of passengers. As shown in Figure 4, we have a pair of circles A and B which are the origin and destination of a single passenger. As usual, we have the origin and destination of the driver as well, denoted as points O and D. According to Proposition 1, we find the initial point of P_A, denoted as P_A^0, given points O, B and $\odot A$. Similarly, we find P_B^0, given points A, D and $\odot B$, the total distance is denoted as d^0. Next, we search in the neighbourhood of P_A^0. Given a fixed range and spacing, we pick K points in the neighbourhood of P_A^0. For each neighborhood point denoted as $(P_A^0)_k$, we find the corresponding optimal point $(P_B^0)_k$ (given fixed point $(P_A^0)_k$, point D and $\odot B$). After that, we calculate the total distance d_k^0 associated with $(P_A^0)_k$ and select the neighbor and its corresponding P_B who have the shortest total distance to be P_A^1 and P_B^1. The corresponding total distance is denoted as d^1. We also denote $\epsilon_1 = d^1 - d^0$. Similarly, we search the K neighbourhood points of P_B^1 and obtain P_A^2, P_B^2 and d^2. We iterate for a finite number of iterations F or until the error is smaller than a given precision ϵ. Then we have found the optimal pair of points P_A^* and P_B^*.

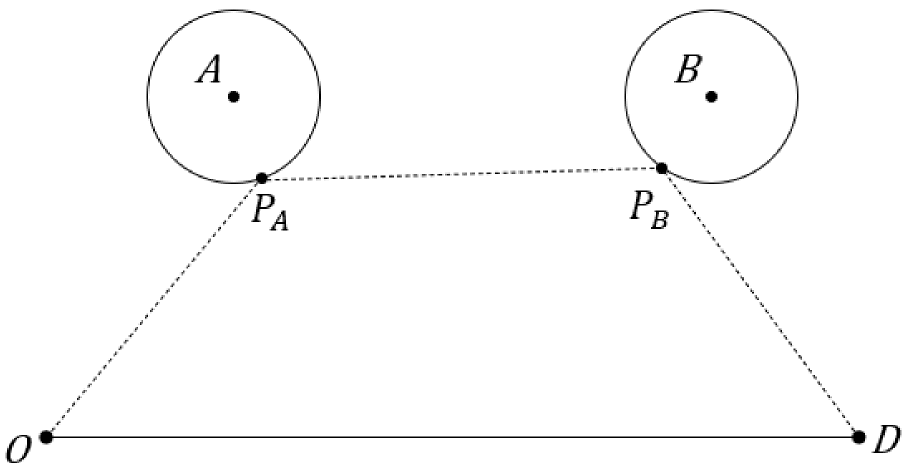


Figure 4. The Two-Circle Case

We next describe how the above procedures can be applied to the $2q$-circle case (q passengers). Using the same set of notation as in Section 3, Algorithm 2 describes our solution approach.

Given a vehicle route $\mathbb{Z}$ of q passengers, Z_1 and Z_{2q+2} are the origin and destination of the driver which are fixed points. Additionally, we have $\odot Z_2, \dots, \odot Z_{2q+1}$ representing the circles centered at points $Z_2, \dots, Z_{2q+1}$. The idea of Algorithm 2 is to find a pair of P_A^* and P_B^* at a time where $A = Z_i$ and $B = Z_{2q+3-i}$ for $i = 2, \dots, q+1$. When determining a pair of $P_{Z_i}^*$ and $P_{Z_{2q+3-i}}^*$, we use exactly the same procedures as the two-circle case we introduced in above. The only difference is that we have an extra step of propagating the $\left(P_{Z_{i+1}}^{f-1}\right)_k, \dots, \left(P_{Z_{2q+2-i}}^{f-1}\right)_k$ during the neighbourhood search in each iteration. The way to do that is to apply Proposition 1 and obtain the optimal points. For example, given $\odot Z_{i+1}$ and fixed points $\left(P_{Z_i}^{f-1}\right)_k$ and $P_{Z_{i+2}}^{f-1}$, we determine $\left(P_{Z_{i+1}}^{f-1}\right)_k$. This procedure terminates after F iterations or if the total distance error ϵ_f is smaller than a given error parameter ϵ.

Since the one circle case can be solved in $O(1)$ time, the time complexity of Algorithm 2 is at most $O(KFn^2)$ where K is a given parameter describing the number of neighbourhood searches, F is the number of iterations which is associated with given parameter ϵ and n is the total number of passengers in the system. Therefore, we have deviced an algorithm for solving the quadratically constrained quadratic program shown at the end of Section 3. We next show in Proposition 2 that this algorithm is optimal when parameter $\epsilon \to 0$.

Algorithm 2: The Location Selecting Algorithm

Input : Vehicle route $\mathbb{Z}$, q number of passengers in route $\mathbb{Z}$, and location coordinates $(r^x_{Z_i}, r^y_{Z_i})$, radius L_{Z_i} for each point Z_i, $i = 1, \ldots, 2q+2$

Output: Meeting points $P^\star_{Z_i} = (l^x_{Z_i}, l^y_{Z_i})$ for $i = 2, \ldots, 2q+1$

for $i = 2, \ldots, 2q+1$ **do**
    let $O = Z_{i-1}$, $D = Z_{i+1}$, and $A = Z_i$
    solve for initial point $P^0_{Z_i}$ and calculate d^0 by Theorem 1 and the Eberly Algorithm.
for $i = 2, \ldots, q+1$ **do**
    for $f = 1, \ldots, F$ **do**
        if *f is odd* **then**
            for $k = 1, \ldots, K$ **do**
                obtain $\left(P^{f-1}_{Z_i}\right)_k$
                calculate the corresponding $\left(P^{f-1}_{Z_j}\right)_k$ points for $j = i+1, \ldots, 2q+3-i$
                calculate the total distance d^{f-1}_k
            pick $\left\{\left(P^{f-1}_{Z_i}\right)_k\right\}^{2q+3-i}_{i=2}$ with lowest d^{f-1}_k and set them as $\left\{P^f_{Z_i}\right\}^{2q+3-i}_{i=2}$ and d^f
        else
            for $k = 1, \ldots, K$ **do**
                obtain $\left(P^{f-1}_{Z_{2q+3-i}}\right)_k$
                calculate the corresponding $\left(P^{f-1}_{Z_j}\right)_k$ points for $j = i, \ldots, 2q+2-i$
                calculate the total distance d^{f-1}_k
            pick $\left\{\left(P^{f-1}_{Z_i}\right)_k\right\}^{2q+3-i}_{i=2}$ with lowest d^{f-1}_k and set them as $\left\{P^f_{Z_i}\right\}^{2q+3-i}_{i=2}$ and d^f
        obtain $\epsilon_f = d^f - d^{f-1}$
        if $\epsilon_f < \epsilon$, $P^\star_{Z_i} = P^f_{Z_i}$, $P^\star_{Z_{2q+3-i}} = P^f_{Z_{2q+3-i}}$, **break**
    if $f = F$, $P^\star_{Z_i} = P^F_{Z_i}$, $P^\star_{Z_{2q+3-i}} = P^F_{Z_{2q+3-i}}$.

Proposition 2. When the given error parameter $\epsilon \to 0$, the solution of Algorithm 2 goes to optimal.

Proof. We prove by induction. Previously, we have shown that when there is only one circle in the sequence, there exists a closed-form optimal solution. Therefore, when the number of circles $k = 1$, the statement holds.

When there is only one passenger on vehicle route $\mathbb{Z}$, that is $k = 2$ and $q = 1$ as shown in Figure 4. For every neighborhood of P_A, the corresponding optimal P_B is calculated and vice versa. Therefore, if one of the points P_A^* and P_B^* is found during the neighbourhood search, the other one is found immediately. The quadratic model at the end of Section 3.1 is convex due to the fact that both the objective and the constraints are convex. Therefore, an optimal solution exists. By carefully selecting the neighborhood search parameters (spacing, search range and etc.), we can find the global optima. We use P_A^* as an example. First, we observe that it is within a bounded region. As shown in Figure 5, P_A^* is within the range of P_A^l and P_A^r on the circle boundary. P_A^l and P_A^r are determined by picking extreme points P_B^1 and P_B^2 on $\odot B$. Since our initial point P_A^0 is obtained by fixed points O, B and $\odot A$, it will be within the solution region as well. Then, an original search range radius of $\pm\frac{\pi}{4}$ from P_A^0 would be enough to cover the solution region. Additionally, we observe that, there is no local optima because moving away from P_A^* will result in non-decreasing total distances (the route of $O - P_A - P_B - D$). To better illustrate this, see Figure 6. The figure on the left is an example of a two-circle case with green crosses being the origin and destination of a driver and the blue dots being the origin and destination of a passenger. The 10 red dots are equally spaced P_A on the circumference of $\odot A$ and they are numbered 1 to 10 from left to right. The plot on the right shows how the total minimum distance of the route changes when we fix P_A and obtain the corresponding optimal P_B. We can see that it is indeed a convex plot. This guarantees that as long as we are approaching P_A^*, we will obtain the optimal objective value. As we calculate the total distance for each neighborhood point, we can pick the two consecutive points that have a decrease and then an increase in the total distance, and set it to be the search bounds for the next iteration. As a result, if we set $\epsilon \to 0$, we will obtain the optimal points P_A^* and P_B^*. Therefore, when the number of circles$k = 2$, the statement holds.

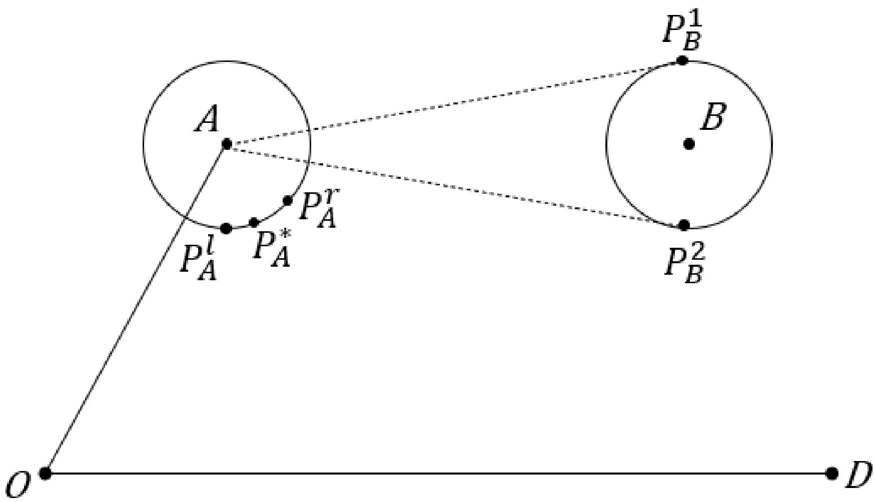
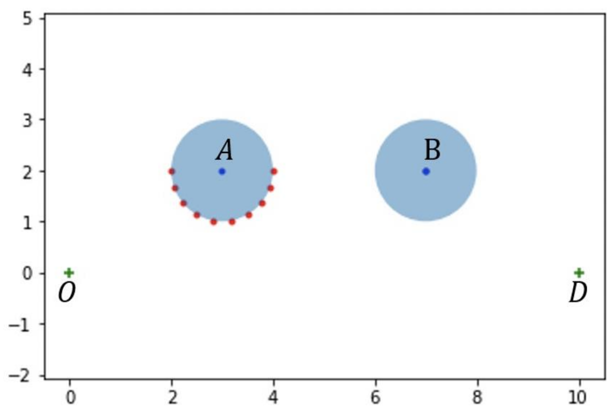
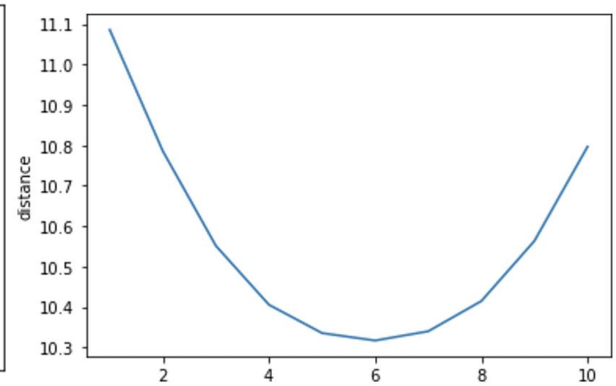


Figure 5. The Solution Region



(a) The Two-Circle Example with Points



(b) The Total Distance Plot for Selected Points

Figure 6. Distance Information of the Studied Region

Assume that the statement holds when $k = 2q - 1$, then we have that Algorithm 2 goes to optimal when $\epsilon \to 0$. Therefore, when $k = 2q$ and $\epsilon \to 0$, the $2q$ circles can be divided into $2q - 1$ circles and 1 circle. By assumption, we can find the optimal meeting points for the first $2q - 1$ circles for each fixed meeting point of the last circle. And by fixing the $(2q-1)^{\text{th}}$ circle, we can find the optimal meeting point of the last circle. Therefore, the problem when $k = 2q$ has been reduced in the same fashion as when $k = 2$ which we have already shown to be true. As a result, the statement holds when $k = 2q$ and the proof is complete.

5 Experiments

In this section, we present our experimental analysis. The main purpose of these experiments is to see how HOV lanes and meeting points contribute to the efficiency of a dynamic ride-sharing system. In general, there will be four different types of settings: 1) the control group which has no HOV lanes and every passenger request is served at the exact location instead of meeting points; 2) the HOV experimental group which has HOV lanes but no meeting points; 3) the meeting points experimental group which has no HOV lanes but meeting points, and 4) the experimental group with both the HOV lanes and the meeting points.

All experiments are run on a 20 by 20 grid where OD pairs are randomly generated. For each driver, the normal driving speed on average (without HOV lane) is 36 miles per hour and the vehicle capacity is set to 4 (including the driver). The maximum detour time is set to be 1.5 times the driver's direct travel time. For each passenger request, it comes with a randomly generated maximum walking time within 0 to 20 minutes (for the control setting, this will be 0 for all requests) and they have the same walking speed of 3 miles per hour. The requests submission follows a Poisson process with a mean of 850 requests an hour. We performed 20 replications, each has 300 passengers and 100 drivers.

As for the performance measures, there are four major measures: 1) the average time spent in the ride-sharing system for each driver; 2) the average extra time needed for each served request; 3) the average in-vehicle time (IVT) ratio for each request, and 4) the percentage of requests served in the ride-sharing system. The first measure monitors the cost to the drivers. The second and the third measures monitor the cost to the passengers. In detail, the second measure is calculated by summing the extra time needed for all served requests (excluding the direct travel time for the driver) and then divide the summation by the number of served requests. The third measure is obtained by averaging the in-vehicle time ratio among the served requests. The IVT ratio is the ratio between the actual in-vehicle time and the direct travel time (without HOV lanes) of the passenger. The last measure is to check and compare the efficiency of the ride-sharing system under different settings.

There are also four other measures that help with understanding the results: 1) the average direct distance of the drivers; 2) the average miles each vehicle travels; 3) the percentage of vehicles with passengers, and 4) the average requests served per loaded vehicle.

We first present the results for the HOV experimental group. In this group, we change normal lanes to HOV lanes with the requirement of 2+ or 3+ persons in a vehicle. The driving speed on HOV lanes is around 50 miles per hour if the requirement is 2+ persons in a vehicle or around 70 miles per hour if the requirement is 3+. Table 1 shows the results under different HOV lane requirements. The first column is the control group while the “HOV2” means all roads contain HOV lanes that requires 2+ persons in a vehicle. Similarly, “HOV3” means all roads contain HOV lanes that requires 3+ persons in a vehicle. As we can see, HOV lanes indeed reduce the average IVT ratio since they provide a boost in driving speed. They also help increase the percentage of requests served in the system since the time saved in using HOV lanes can be utilized to serve more requests and serving more requests increases the possibility of using the HOV lanes. Another observation is that HOV2 outperforms HOV3 in the percentage of requests served. This is due to HOV3 has higher requirements for entering the HOV lanes thus forcing the drivers to pick up more passengers along the way to be eligible for the speed boost, resulting in an increase in average request served per vehicle but a decrease in the percentage of loaded vehicles.

Table 1. Performance Measures for HOV Experimental Group

	Original	**HOV2**	**HOV3**
Avg Time per Vehicle (min)	22.35	21.66	20.80
Avg Extra Time per Request (min)	4.03	2.70	2.31
Avg IVT Ratio	1.16	0.91	0.85
% Requests Served	40.50%	51.98%	48.43%
Avg Direct Distance per Vehicle (mile)	10.48	10.48	10.48
Avg Distance Travelled per Vehicle (mile)	13.41	15.82	14.93
Avg # of Requests Served	2.34	2.52	2.84
% of Loaded Vehicles	52.05%	61.95%	51.30%

Next, we present the results for the meeting points experimental group. In this group, we change the maximum walking time from 0 to 10 to 20 minutes. Table 2 shows the results. As we increase the maximum walking time, more requests are served in the system. This is due to the fact that the larger walking time allows closer meeting points among locations so more passengers can be served within the same amount of travel distance. Since no HOV lanes are involved in this experimental group, there will not be any speed boost to help save time. But if we take a closer look, we can find that around a 4.2% increase in time spent per vehicle and 4.3% increase in average IVT ratio bring around 10.7% increase in usage of the ride-sharing system. The percentage of requests served in the 20-min-walk case is the same as that of the HOV3 in Table 1 with fewer miles travelled per vehicle and fewer empty vehicles. The reduction in miles travelled while maintaining the same level of system usage indicates its ability in reducing detours.

Table 2. Performance Measures for Meeting Points Experimental Group

	Original	**10-min-walk**	**20-min-walk**
Avg Time per Vehicle (min)	22.35	23.28	24.28
Avg Extra Time per Request (min)	4.03	4.33	4.71
Avg IVT Ratio	1.16	1.21	1.28
% Requests Served	40.50%	44.83%	48.32%
Avg Direct Distance per Vehicle (mile)	10.48	10.48	10.48
Avg Distance Travelled per Vehicle (mile)	13.41	13.97	14.57
Avg # of Requests Served	2.34	2.59	2.77
% of Loaded Vehicles	52.05%	52.05%	52.40%

Lastly, we want to see how the combination of HOV lanes and meeting points together contribute to the efficiency of the ride-sharing system. As shown in Table 3 and Table 4, we find that increase in maximum walking time always results in higher system usage for ride-sharing while HOV lanes always results in time efficiency which significantly increases the passengers' in-vehicle experience and reduce the passengers' cost (due to sharing the ride).

Table 3. Average IVT Ratio with HOV Lanes and Meeting Points

Avg IVT Ratio	**Original**	**HOV2**	**HOV3**
Original	1.16	0.91	0.85
10-min-walk	1.21	0.94	0.86
20-min-walk	1.28	0.97	0.88

Table 4. Percentage Requests Served with HOV Lanes and Meeting Points

% Requests Served	Original	HOV2	HOV3
Original	40.50%	51.98%	48.43%
10-min-walk	44.83%	56.52%	52.88%
20-min-walk	48.32%	59.57%	56.37%

6 Conclusion

In this project, we explored the use of HOV lanes and meeting points in a ride-sharing system where drivers have their own origin and destination. In order to adapt to the dynamic environment, we proposed a two-stage heuristic algorithm which consists of an insertion heuristic to solve the PDP problem and a second stage algorithm that can solve the meeting points problem optimally in polynomial time. Our experimental results show that the HOV lanes and meeting points can increase the efficiency of the dynamic ride-sharing system. When we choose a maximum walking time of 10 minutes and a common HOV requirements of 2+ persons in the vehicle, the system could serve 39% more requests while reducing the average in-vehicle time ratio by 19%. There- fore, we can say that a good combination of HOV lanes and meeting points can provide passengers with lower commuting cost and faster commuting experience.

References

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

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

Attanasio A, Cordeau JF, Ghiani G, Laporte G (2004) Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing 30(3):377–387.

Aziez I, Côté JF, Coelho LC (2020) Exact algorithms for the multi-pickup and delivery problem with time windows. European Journal of Operational Research.

Berbeglia G, Cordeau JF, Laporte G (2010) Dynamic pickup and delivery problems. European Journal of Operational Research 202(1):8–15.

Braekers K, Caris A, Janssens GK (2014) Exact and meta-heuristic approach for a general heterogeneous dial- a-ride problem with multiple depots. Transportation Research Part B: Methodological 67:166–186.

Catay B (2009) Ant colony optimization and its application to the vehicle routing problem with pickups and deliveries. Natural Intelligence for Scheduling, Planning and Packing Problems, 219–244 (Springer).

Chang SK, Schonfeld PM (1991) Multiple period optimization of bus transit systems. Transportation Research Part B: Methodological 25(6):453–478.

Chassaing M, Duhamel C, Lacomme P (2016) An els-based approach with dynamic probabilities management in local search for the dial-a-ride problem. Engineering Applications of Artificial Intelligence 48:119–133.

Cordeau JF (2006) A branch-and-cut algorithm for the dial-a-ride problem. Operations Research 54(3):573–586.

Cordeau JF, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological 37(6):579–594.

Cordeau JF, Laporte G (2005) Tabu search heuristics for the vehicle routing problem. Metaheuristic Optimization via Memory and Evolution, 145–163 (Springer).

Coslovich L, Pesenti R, Ukovich W (2006) A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European Journal of Operational Research 175(3):1605–1615.

Diana M, Dessouky MM (2004) A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transportation Research Part B: Methodological 38(6):539–557.

Eberly D (2008) Geometric tools: Computing a point of reflection on a sphere. URL https://www. geometrictools.com/Documentation/SphereReflections.pdf.

Fielbaum A, Jara-Diaz S, Gschwender A (2020) Beyond the mohring effect: Scale economies induced by transit lines structures design. Economics of Transportation 22:100163.

Furuhata M, Dessouky M, Ordóñez F, Brunet ME, Wang X, Koenig S (2013) Ridesharing: The state-of-the-art and future directions. Transportation Research Part B: Methodological 57:28–46.

Häll CH, Peterson A (2013) Improving paratransit scheduling using ruin and recreate methods. Transportation Planning and Technology 36(4):377–393.

Ho SC, Szeto W, Kuo YH, Leung JM, Petering M, Tou TW (2018) A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological 111:395–421.

Hurdle V (1973) Minimum cost locations for parallel public transit lines. Transportation Science 7(4):340–350.

Kleiner A, Nebel B, Ziparo VA (2011) A mechanism for dynamic ride sharing based on parallel auctions.Twenty-Second International Joint Conference on Artificial Intelligence.

Levy JI, Buonocore JJ, Von Stackelberg K (2010) Evaluation of the public health impacts of traffic congestion: a health risk assessment. Environmental health 9(1):1–12.

Li RH, Qin L, Yu JX, Mao R (2015) Optimal multi-meeting-point route search. IEEE Transactions on Knowledge and Data Engineering 28(3):770–784.

Li X, Hu S, Fan W, Deng K (2018) Modeling an enhanced ridesharing system with meet points and time windows. PloS one 13(5):e0195927.

Lu Q, Dessouky M (2004) An exact algorithm for the multiple vehicle pickup and delivery problem. Transportation Science 38(4):503–514.

Lyu Y, Chow CY, Lee VC, Ng JK, Li Y, Zeng J (2019) Cb-planner: A bus line planning framework for customized bus systems. Transportation Research Part C: Emerging Technologies 101:233–253, ISSN 0968-090X.

Marković N, Nair R, Schonfeld P, Miller-Hooks E, Mohebbi M (2015) Optimizing dial-a-ride services in maryland: benefits of computerized routing and scheduling. Transportation Research Part C: Emerging Technologies 55:156–165.

McGuckin N, Fucci A (2017) Summary of travel trends: National household travel survey (nhts). 2017.

Mohring H (1972) Optimization and scale economies in urban bus transportation. The American Economic Review 62(4):591–604.

Nie YM (2017) How can the taxi industry survive the tide of ridesourcing? evidence from shenzhen, china. Transportation Research Part C: Emerging Technologies 79:242–256.

Núñez A, Cortés CE, Sáez D, De Schutter B, Gendreau M (2014) Multiobjective model predictive control for dynamic pickup and delivery problems. Control Engineering Practice 32:73–86.

Papoutsis P, Fennia S, Bridon C, Duong T (2021) Relaxing door-to-door matching reduces passenger waiting times: a workflow for the analysis of driver gps traces in a stochastic carpooling service. Transportation Engineering 100061, ISSN 2666-691X.

Parragh SN, Doerner KF, Hartl RF (2010) Variable neighborhood search for the dial-a-ride problem. Computers & Operations Research 37(6):1129–1138.

Pei M, Lin P, Du J, Li X (2019) Operational design for a real-time flexible transit system considering passenger demand and willingness to pay. IEEE Access 7:180305–180315.

Pi M, Yeon H, Son H, Jang Y (2021) Visual cause analytics for traffic congestion. IEEE Transactions on Visualization and Computer Graphics 27(3):2186–2201.

Pillac V, Gendreau M, Guéret C, Medaglia AL (2013) A review of dynamic vehicle routing problems. European Journal of Operational Research 225(1):1–11.

Posada M, Andersson H, Häll CH (2017) The integrated dial-a-ride problem with timetabled fixed route service. Public Transport 9(1-2):217–241.

Psaraftis HN (1988) Dynamic vehicle routing problems. Vehicle routing: Methods and studies 16:223–248. Ropke S, Cordeau JF (2009) Branch and cut and price for the pickup and delivery problem with time windows. Transportation Science 43(3):267–286.

Ropke S, Cordeau JF, Laporte G (2007) Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks: An International Journal 49(4):258–272.

Sayarshad HR, Chow JY (2015) A scalable non-myopic dynamic dial-a-ride and pricing problem. Transportation Research Part B: Methodological 81:539–554.

Schilde M, Doerner KF, Hartl RF (2011) Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers & Operations Research 38(12):1719–1730.

Schrank D, Albert L, Eisele B, Lomax T (2021) 2021 urban mobility report. Texas A&M Transportation Institute. The Texas A&M University System.

Schrank D, Eisele B, Lomax T (2019) 2019 urban mobility report. Texas A&M Transportation Institute. The Texas A&M University System.

Smet P (2021) Ride sharing with flexible participants: a metaheuristic approach for large-scale problems. International Transactions in Operational Research 28(1):91–118.

Sombuntham P, Kachitvichayanukul V (2010) A particle swarm optimization algorithm for multi-depot vehicle routing problem with pickup and delivery requests. World Congress on Engineering 2012. July 4-6, 2012. London, UK., volume 2182, 1998–2003 (Citeseer).

Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2015) The benefits of meeting points in ride-sharing systems. Transportation Research Part B: Methodological 82:36–53, ISSN 0191-2615.

Stock E (2018) Introducing express pool: Walk a little to save a lot. URL https://www.uber.com/newsroom/expresspool/#:~:text=Express%20POOL%20has%20been%20piloting,Washington%2C%20D.C.%20on%20February%2022.

Tirachini A (2014) The economics and engineering of bus stops: Spacing, design and congestion. Transportation Research Part A: Policy and Practice 59:37–57.

Ulmer MW, Thomas BW, Campbell AM, Woyak N (2020) The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. Transportation Science.

(US) FHA, (US) FTA (2013) 2013 Status of the Nation's Highways, Bridges, and Transit Conditions and Performance Report to Congress (Government Printing Office).

Wong KI, Bell MG (2006) Solution of the dial-a-ride problem with multi-dimensional capacity constraints. International Transactions in Operational Research 13(3):195–208.

Xiang Z, Chu C, Chen H (2008) The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. European Journal of Operational Research 185(2):534–551.

Yan C, Zhu H, Korolko N, Woodard D (2020) Dynamic pricing and matching in ride‐hailing platforms. Naval research logistics. 67(8):705-724.

Zhang Z, Liu M, Lim A (2015) A memetic algorithm for the patient transportation problem. Omega 54:60–71.

Zhao M, Yin J, An S, Wang J, Feng D (2018) Ridesharing problem with flexible pickup and delivery locations for app-based transportation service: mathematical modeling and decomposition methods. Journal of Advanced Transportation 2018.

Zidi I, Mesghouni K, Zidi K, Ghedira K (2012) A multi-objective simulated annealing for the multi-criteria dial a ride problem. Engineering Applications of Artificial Intelligence 25(6):1121–1131.

Zou H (2017) An Online Cost Allocation Model for Horizontal Supply Chains. Ph.D. thesis, University of Southern California.

Data Summary

Products of Research

The main research products will be peer-reviewed journal articles, book chapters and/or conference proceedings targeted towards the transportation science research community, plus supplemental materials such as tables, numerical data used for graphs, etc. No personal data will be used in the project, so there is no threat of identity theft.

Data Format and Content

All research products will be available online in digital form. Manuscripts will appear in a common document-viewing format, such as PDF, and supplemental materials such as tables and numerical data will be in a tabular format, such as Microsoft Excel spreadsheet, tab-delimited text, etc.

Data Access and Sharing

All participants in the project will publish the results of their work. Papers will be published in peer-reviewed scientific journals, books published in English, conference proceedings, or as peer-reviewed data reports. The primary data source is generated using simulation and is described in detail in the report. Since it is randomly generated using a computer, the hardware specifications of the computer and the codes are stored in the database (https://doi.org/10.7910/DVN/P6ROMD).

Reuse and Redistribution

USC's policy is to encourage, wherever appropriate, research data to be shared with the general public through internet access. This public access will be regulated by the university in order to protect privacy and confidentiality concerns, as well to respect any proprietary or intellectual property rights. Administrators will consult with the university's legal office to address any concerns on a case-by-case basis, if necessary. Terms of use will include requirements of attribution along with disclaimers of liability in connection with any use or distribution of the research data, which may be conditioned under some circumstances.

The following citation should be used when referencing the dataset:

> Hu, Shichun, 2021, "NCST Report - Dynamic Ride-Sharing with HOV Lanes and Meeting Points", https://doi.org/10.7910/DVN/P6ROMD, Harvard Dataverse, V1

Appendix: List of Notations

$\mathbb{P}$	The passengers set that index from 1 to n
$\mathbb{P}_{v,t}$	The set of passengers assigned to driver v at time t
$\mathbb{V}$	The drivers set that index from 1 to m
$\mathbb{V}_{p,t}^{\mathrm{F}}$	The feasible set of drivers for passenger p at time t
$\mathbb{R}_{v,t}$	The current route of driver v at time t
$O_p(O_v)$	The origin of passenger p (driver v)
$D_p(D_v)$	The destination of passenger p (driver v)
O_p^{d}	The deviated origin of passenger p
D_p^{d}	The deviated destination of passenger p
L_p	The maximum walking distance of passenger p
I_p	The maximum wait time of passenger p before the vehicle arrives
W	The constant walking speed for all passengers
H_v	The maximum in-vehicle time of driver v
T_v	The maximum detour time for driver v
E	The average driving speed for all drivers
U	The constant capacity for all drivers' vehicles
$G(\mathbb{N}, \mathbb{A})$	The graph G with node set $\mathbb{N}$ and arc set $\mathbb{A}$
$\mathbb{O}_p(\mathbb{O}_v)$	The node set of origins of all passengers (drivers)
$\mathbb{D}_p(\mathbb{D}_v)$	The node set of destinations of all passengers (drivers)
$c_{i,j}$	The travel time between node i and j
$d_{i,j}$	The Euclidean distance between node i and j
$\beta_{i,j}$	The time discount factor between node i and j

H	The number of people required to go on a HOV lane
(r_i^x, r_i^y)	The coordinates of node i
(l_i^x, l_i^y)	The coordinates of the deviated location of node i
$g_{i,v}$	The load indicator for driver v at node i
$y_{i,j,v}$	The binary indicator of whether driver v travels from node i to node j
$b_{i,j,v}$	The binary indicator of whether driver v visits node i before node j
$\alpha_{i,j,v}$	The binary indicator of whether driver v can use HOV lane when travelling from node i to node j