IISE Transactions

ISSN: 2472-5854 (Print) 2472-5862 (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

An integrated car-and-ride sharing system for mobilizing heterogeneous travelers with application in underserved communities

Miao Yu & Siqian Shen

To cite this article: Miao Yu & Siqian Shen (2020) An integrated car-and-ride sharing system for mobilizing heterogeneous travelers with application in underserved communities, IISE Transactions, 52:2, 151-165, DOI: 10.1080/24725854.2019.1628377

To link to this article: https://doi.org/10.1080/24725854.2019.1628377

IISE TRANSACTIONS
2020, VOL. 52, NO. 2, 151–165
https://doi.org/10.1080/24725854.2019.1628377

An integrated car-and-ride sharing system for mobilizing heterogeneous travelers with application in underserved communities

Miao Yu and Siqian Shen

Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA

ABSTRACT

The fast-growing carsharing and ride-hailing businesses are generating economic benefits and societal impacts in modern society, while both have limitations to satisfy diverse users, e.g., travelers in low-income, underserved communities. In this article, we consider two types of users: Type 1 drivers who rent shared cars and Type 2 passengers who need shared rides. We propose an integrated car-and-ride sharing (CRS) system to enable community-based shared transportation. To compute solutions, we propose a two-phase approach where in Phase I we determine initial car allocation and Type 1 drivers to accept; in Phase II we solve a stochastic mixed-integer program to match the accepted Type 1 drivers with Type 2 users, and optimize their pick-up routes under a random travel time. The goal is to minimize the total travel cost plus expected penalty cost of users' waiting and system overtime. We demonstrate the performance of a CRS system in Washtenaw County, Michigan by testing instances generated based on census data and different demand patterns. We also demonstrate the computational efficacy of our decomposition algorithm benchmarked with the traditional Benders decomposition for solving the stochastic model in Phase II. Our results show high demand fulfillment rates and effective matching and scheduling with low risk of waiting and overtime.

ARTICLE HISTORY
Received 18 November 2018
Accepted 30 May 2019

KEYWORDS
Carsharing; ride-hailing; stochastic vehicle routing; two-stage stochastic integer programming; decomposition algorithm

1. Introduction

1.1. Overview

In recent years, shared-mobility has shown its flexibility and strength in providing convenience in personal travel and reducing traffic congestion on public roads by reducing car ownership (Shaheen *et al.*, 2015; Martin, 2016). With increasing concerns about climate change, congestion, and fossil fuel dependency, shared-mobility is attracting increasing levels of attention and is undergoing a fast rise in popularity and industrial growth (Chan and Shaheen, 2012). Two main mobility-sharing types are carsharing and ride-hailing. The former provides users with access to car rental service on an hourly basis (Millard-Ball, 2005). The latter, a service with the aim of grouping travelers with common itineraries (Chan and Shaheen, 2012), has evolved rapidly with the development of smartphone technologies. Although both carsharing and ride-hailing provide useful alternatives to owning personal cars, they focus on different user groups. Carsharing users usually rent cars for running errands with multiple short stops. To use a carsharing service, users are required to meet driving eligibility requirements. Ride-hailing serviced target on users with short, and often one-time ride needs. It especially benefits people who are not able to drive, unfamiliar with the city transit system, and/or commuting with poor transit access.

Carsharing business started from a station-based model in the United States in the late-1990s and individuals join a membership program by paying fixed monthly or annual fee. Service providers often ask users to reserve a vehicle in advance, pick up and return the vehicle to the same station, which is often located in a population-dense region. In recent years, carsharing has evolved to allow users to pick up and drop off vehicles in different locations, and some free-floating carsharing options only require their users to return vehicles to a zone in service. However, this flexibility comes at the cost of vehicle stock imbalance and drives up the price to use the service.

With the support of GPS technology and broad adoption of smartphones, ride-hailing services have emerged in recent years. They allow customers to request drivers through a smartphone app. As a result, its efficiency and service quality rely on hundreds of thousands of drivers who participate in real time. Working as a driver for ride-hailing service platforms has become a new career choice for many people and thus contributes to employment growth. However, to become a driver, one needs to pass a series of tests including having a vehicle that meets the model year requirement. Many people have decided to rent or lease a car to drive for ride-hailing jobs from platforms such as HyreCar. However, they still bear the cost of operating vehicles and uncertain demand. Clewlow and Mishra (2017) show that between 30 and 50% of ride-hailing drivers lose money. Furthermore, those car rental services are only provided in large metropolitan areas. Nevertheless, ride-hailing service are attracting considerably more users than carsharing, both from the

CONTACT Siqian Shen siqian@umich.edu

passenger side and the driver side: More than 250 000 000 users globally for ride-hailing verse 5 000 000 users for carsharing.

Both carsharing and ride-hailing services rely on several characteristics of existing transportation systems to be successful, including limited parking, limited public transportation, walkability, high population density, and mixed-use neighborhoods (Muheim and Reinhardt, 1999; Brook, 2004). In cities with a high population density, the waiting time for both drivers and passengers can be easily reduced in ride-hailing and thus attract more users with a high efficiency (Agatz *et al.*, 2012; Alonso-Mora *et al.*, 2017). In fact, both carsharing and ride-hailing services are targeting users who are young, educated, have a moderate income, and live in urban areas (Smith, 2016). However, for the neighborhoods that do not meet those characteristics, the services are less likely to be successful, and the coverage for such neighborhoods is therefore limited.

Given the variety of limitations we listed above, in this article, we propose a Car-and-Ride Sharing (CRS) system, in which carsharing and ride-hailing services are integrated and co-provided to boost the mobility of heterogeneous travelers based on their characteristics and special needs. We aim to design a financially and operationally self-sustainable system, such that users with ride-hailing demands are served by drivers with carsharing needs. We aim to build an affordable, reliable, and incentive system to foster the connections within and in between communities that do not have full access to the existing carsharing and ride-hailing systems.

1.2. Application in underserved communities

Transportation is a scarce resource in metropolitan Detroit: 40% of residents do not own cars, and another 40% do not have access to vehicles (Firth, 2016). Despite the fact that the surrounding suburban areas house more than five times the employment opportunities as the city of Detroit, only 9% of the jobs are taken by residents in Detroit. In Detroit, more than 10 000 residents face tremendous problems in commuting to their jobs in suburban communities that do not offer public transit. Even worse, some suburban municipalities have decided to reduce, and in some cases eliminate, much of its public bus service, due to financial and safety concerns (McLaughlin, 2015).

Residents in underserved communities are experiencing financial, technical, skill-based, informational and social barriers to the use of shared-mobility services. For both service providers and users, the adoption of the existing shared-mobility forms in underserved communities poses significant challenges. From the service providers' perspective, income and crime rate influence the willingness of drivers to serve particular areas, and therefore residents in those areas have fewer drivers and a higher service price (Thebault-Spieker *et al.*, 2015). Meanwhile, due to the lack of personally owned vehicles, residents in low-income communities are disadvantaged in finding work as a driver for ride-hailing (Gross *et al.*, 2012).

For validating the design of CRS, we consider a service region in underserved communities that mainly involves, e.g., underserved populations, such as jobless, elderly, and disabled, to whom transportation is a scarce resource. We partition the service region into zones, and shared cars are located *a priori* in designated parking spaces in each zone. We classify two types of users: Type 1 who want shared cars for private use, but also have spare time outside their travel time windows to serve as drivers; and Type 2 who have ride-hailing needs, but cannot drive themselves. We aim to build a self-sustained CRS system to encourage Type 1 drivers to "serve" Type 2 users at the capacity to receive profit to compensate for their payment for car rental.

To implement CRS, we build a reservation system that collects Type 1 drivers' rental information along with their available time windows for serving others, as well as Type 2 users' ride-hailing requests with the origin and destination of their trips and time windows for pick-up. Note that both carsharing and ride-hailing have two primary forms: Reservation-based and on-demand. From the users' perspective, on-demand service is more appealing, as it provides an instant solution for their travel needs. However, in this article, we assume that all the requests are collected prior to the optimization phase for the following reasons. First, an on-demand ride-hailing service usually requires a large number of drivers and passengers to achieve matching efficiency. We consider travel needs from underserved communities, and thus the demand is sparsely distributed. Second, Type 1 drivers who serve others to gain more vehicle access often know their available time sufficiently early before committing to the service. Third, the purposes of users' trips can be generally categorized as job commute, hospital visit, job interview, and grocery shopping, which all have a fixed schedule and can be known in advance. Based on this, we propose a reservation-based CRS system and assume that all supplies and demands are known in advance.

1.3. Contributions and main results

This article focuses on developing a new CRS system for serving heterogeneous travelers whose demand cannot be solely met by either carsharing or ride-hailing business models. The contributions of this paper are two-fold:

1. The classification of two types of travelers leads to self-sustained operations of the system via supply–demand matching. However, combining two types of services results in substantial operational challenges. We decompose the problem into two phases, and successfully reduce operational complexity while obtaining good-quality results, as shown by our computational studies. We also demonstrate the importance of vehicle relocation, as we can achieve higher demand fulfillment rates even when we only allow round trips of shared cars.
2. We extend the basic model to a stochastic integer program to capture the randomness of vehicle travel time and service time. We develop efficient decomposition

algorithms for optimizing the large-scale stochastic model with finite samples. When applied to synthetic data, our proposed model achieves high levels of operational efficiency (measured by waiting and overtime of the system), compared with the deterministic model using the expected values.

The proposed community-based CRS system can be applied beyond the scope of serving underserved populations, as it provides a solution for communities where there is a mixture of travelers with different driving abilities and time flexibility.

1.4. Structure of this article

The remainder of this article is organized as follows. In Section 2, we review the literature related to carsharing, ride-hailing, and their optimization models and algorithms. In Section 3, we describe the problem and formulate a two-phase approach to optimize the integrated CRS design and operations. We further present a stochastic programming variant of the Phase II model by considering uncertain travel time and service time. In Section 4, we apply an efficient way to decompose the proposed models and derive valid cuts based on the integer L-shaped method. In Section 5, we demonstrate the effectiveness of the CRS system and present computational results via testing diverse instances. We conclude this article and present future research directions in Section 6.

2. Literature review

For carsharing, Laporte *et al.* (2015) classify the literature of carsharing under five main topics: Station location, fleet dimensioning, station inventory, rebalancing incentives, and vehicle repositioning. Nourinejad and Roorda (2014) propose a dynamic optimization-simulation model to study the relationship between fleet size and reservation time (i.e., the time between reservation and picking up vehicles). Nair and Miller-Hooks (2014) use a bilevel mixed-integer linear program to optimize station location and capacity, as well as vehicle inventories. To handle vehicle imbalance in a one-way carsharing system, Kek *et al.* (2009) introduce a spatial-temporal network to model the movement of vehicles and determine the workforce needed for relocation. Similar approaches have been used by De Almeida Correia and Antunes (2012) to optimize parking locations, and by Fan (2014) to optimize the allocation and relocation in carsharing systems that allow one-way car rental.

For ride-hailing, Agatz *et al.* (2012) conduct a comprehensive survey of optimization approaches for fleet management and other related operational problems in dynamic ride-hailing. Alonso-Mora *et al.* (2017) propose a general model for a dynamic real-time high-capacity ride-pooling system that solves an assignment problem on a graph of feasible trips and compatible vehicles. To match ride-hailing requests to available cars, as well as to route shared vehicles, most papers consider variants of the Vehicle Routing Problem (VRP) for seeking static ride-hailing solutions. Toth and Vigo (2014) summarize the formulations and solution approaches of VRP variants, including multi-depot VRP, VRP with time windows, and VRP with simultaneous pick-up and delivery. Taş *et al.* (2013) develop heuristic approaches for VRP with soft time windows and stochastic travel time. In addition to routing cost, they also consider the cost of late and early arrivals, which are related to the waiting time and idle time considered in our stochastic formulation discussed later.

For designing carsharing systems, He *et al.* (2016) optimize service zone selection for sharing electric vehicles under uncertain carsharing demand and fuel price. Brandstatter *et al.* (2016) further study facility location problems for locating charging stations in electric vehicle sharing systems. Lu *et al.* (2018) study a carsharing fleet allocation problem with stochastic one-way and round-trip demands. They propose a two-stage stochastic program to minimize the total cost of parking lots/permits and car allocation, as well as penalty cost from unfulfilled demand. Zhang *et al.* (2018) extend the models and solution approaches in Lu *et al.* (2018) for optimizing fleet allocation and service operations of electric vehicle sharing with vehicle-to-grid selling under random travel demand and electricity price. To the best of our knowledge, this article is the first to combine carsharing system design with ride-hailing service optimization, which has special application domains as we described above.

3. Problem formulations

We consider a fleet of K shared vehicles available to be reserved that are distributed in I zones. Let $L = \{1, 2, ..., |L|\}$ be a set of reservations received from Type 1 drivers. Each $l \in L$ is associated with a tuple, $(o_l, d_l, [s_l, t_l], [g_l, h_l])$, which includes the pick-up zone $o_l \in I$, return zone $d_l \in I$ of a rental car, time window $[s_l, t_l]$ during which a car is needed, and the time window $[g_l, h_l]$ during which Type 1 driver l is able to provide ride-hailing service to Type 2 users. Let J be a set of ride-hailing demand from Type 2 (non-driver) users. Each $j \in J$ is associated with a tuple, $(o'_j, d'_j, e_j, g'_j, h'_j)$, where o'_j and d'_j represent the trip's origin and destination, respectively, e_j is the total service time needed (including driving time from o'_j to d'_j and the time to load passengers), and $[g'_j, h'_j]$ is the available time window for picking up the corresponding Type 2 user at origin o'_j. In this article, we consider finite and discrete time for decision making in our models. We define a binary parameter vector $w = (w_{jl}, j \in J, l \in L)^T$, where $w_{jl} = 1$ if ride-hailing $j \in J$ can be served by carsharing trip $l \in L$ and 0 otherwise. We set $w_{jl} = 1$ if $|[g'_j, h'_j + e_j] \cap [g_l, h_l]| \geq e_j$, meaning that Type 2 user $j \in J$ can be served within the time window specified by Type 1 driver $l \in L$.

For each Type 1 demand $l \in L$, we charge r_l^{car} per period of car use, dependent on pick-up and drop-off locations. Each driver l also earns r_l^{drive} for every time period of serving a Type 2 user. For each Type 2 demand $j \in J$, we charge r_j^{ride} per period dependent on the origin, destination, and time window. A car in use will incur a service cost c^{ser} (including maintenance, insurance, and other types of cost) per period and will incur an idle cost c_i^{idle} (including c^{ser} and parking cost) per period if it sits idle at zone $i \in I$. When a car needs to be relocated, it will incur a relocation cost c^{rel} per hour. Note that service cost c^{ser}, idle cost c_i^{idle}, and relocation cost c^{rel} are not paid by Type 1 driver, but are system-based costs to minimize.

3.1. Solution approach overview

We consider a two-phase approach for the design and operations of the CRS system: In Phase I, we maximize the fulfillment of Type 1 demand over a spatial-temporal network that describes all users' demand and availability, while maintaining the financial self-sustainability of the system. We prioritize Type 1 drivers with a high possibility of providing ride-hailing service with a preliminary match of Type 1 drivers and Type 2 users when making the decision. In Phase II, we match Type 1 and Type 2 users and optimize pick-up routes and schedules under a stochastic vehicle travel time and passenger loading (service) time, which may result in users' waiting and overtime. By decomposing the problem into two phases, it reduces the size of the ride-hailing scheduling and routing problem by solely considering the accepted Type 1 drivers, and hence, improves the computational time overall.

We also develop a two-stage stochastic mixed-integer programming formulation for Phase II, which balances vehicles' total travel cost and the expected penalty cost of users' waiting and vehicle-use overtime. To solve the large-scale formulation with many samples of uncertain travel time and service time, we propose a driver-based decomposition algorithm, which first determines a sequence of Type 2 users assigned to each Type 1 driver, and then for each driver, the algorithm decides an optimal schedule to arrive at each Type 2 user's location, which can be quickly solved via linear programming. We conduct computational studies by testing instances based on serving underserved populations in Washtenaw County, Michigan and consider diverse demand patterns of Type 1 and Type 2 users. We show that our decomposition algorithm outperforms the standard Benders decomposition approach, and demonstrate the cost and service-quality performance of the CRS system.

3.2. Phase I: Carsharing planning and operations

Given service requests from Type 1 drivers and Type 2 users, we first implement Phase I to decide which Type 1 requests to fulfill and pass the solutions to Phase II. We construct a spatial-temporal network, $G = (N, A)$, to capture carsharing reservations from Type 1 drivers over T periods, where each node $n_{it} \in N$ represents a zone $i \in I$ at period $t \in \{0, 1, \dots, T\}$. We partition A into three types of arcs, $A = A^{\text{travel}} \cup A^{\text{idle}} \cup A^{\text{rel}}$, as follows:

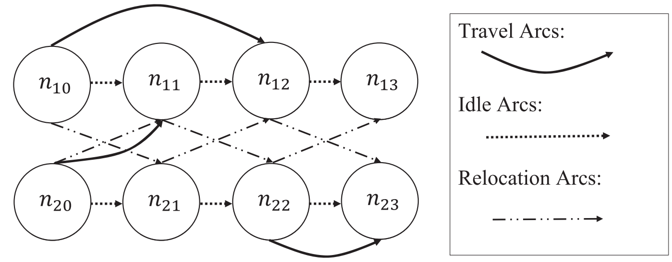

Figure 1. An example spatial-temporal network with two zones and three periods.

1. Travel arcs $(n_{it}, n_{i't'}) \in A^{\text{travel}}$ are created for each Type 1 demand $l \in L$ where $i = o_l$, $i' = d_l$, $t = s_l$, and $t' = t_l$. The amount of flow on arc $a_l \in A^{\text{travel}}$ indicates whether or not a vehicle is being rented by a Type 1 driver $l \in L$. We set $f_{a_l} = (t_l - s_l)(r_l^{\text{car}} - c^{\text{ser}}) - r_l^{\text{drive}}(h_l - g_l)$ as the unit flow revenue and $u_{a_l} = 1$ as the capacity of arc $a_l \in A^{\text{travel}}$, for all $l \in L$.
2. Idle arcs $(n_{it}, n_{i,t+1}) \in A^{\text{idle}}$ are created for $i \in I$ and $t \in \{0, 1, \dots, T-1\}$. The amount of flow on arc $(n_{it}, n_{i,t+1}) \in A^{\text{idle}}$ indicates the number of vehicles that are idle in zone i from t to $t+1$. We set unit flow revenue and capacity of arc $a \in A^{\text{idle}}$ as $f_a = -c_i^{\text{idle}}$ and $u_a = K$, respectively.
3. Relocation arcs $(n_{it}, n_{i',t'}) \in A^{\text{rel}}$ are created for $i \neq i' \in I$ and $t' > t \in \{0, 1, \dots, T-1\}$. The amount of flow on $(n_{it}, n_{i',t'}) \in A^{\text{rel}}$ indicates the number of vehicles being relocated from zone i at t to zone i' at t'. We set the unit flow revenue and capacity of arc $a \in A^{\text{rel}}$ as $f_a = -c^{\text{rel}}(t' - t)$ and $u_a = K$, respectively.

Figure 1 shows a spatial-temporal network example with $I = \{1, 2\}$ and $T = 3$. In the network, there are three Type 1 driver demands: A round trip reserving a car from $t = 0$ to $t = 2$ at zone 1, a round trip demand from $t = 2$ to $t = 3$ at zone 2, and a one-way demand from $t = 0$ to $t = 1$ traveling from zone 2 to zone 1. Note that by allowing vehicle relocation, we can use two vehicles to serve the three demand trips in the example network.

We define integer decision vector $x = (x_i, i \in I)^T$ where x_i is the number of cars initially located in zone $i \in I$, integer decision vector $y = (y_a, a \in A)^T$ where y_a indicates the amount of flow on arc $a \in A$, and binary integer decision vector $z = (z_{jl}, j \in J, l \in L)^T$ where $z_{jl} = 1$ indicates that Type 2 user $j \in J$ can be potentially served by Type 1 driver $l \in L$, and $z_{jl} = 0$ otherwise. Let $\delta^+(n_{it}), \delta^-(n_{it})$ be the sets of arcs to which node n_{it} is their tail and head nodes, respectively, and formulate an integer program **P1** as follows:

$$[\mathbf{P1}] \ \text{maximize}_{x,y,z} \ \sum_{a \in A} f_a y_a + \sum_{j \in J} r_j^{\text{ride}} \sum_{l \in L} e_j z_{jl}, \tag{1a}$$

$$\text{subject to} \ \sum_{i \in I} x_i \leq K, \tag{1b}$$

$$\sum_{a \in \delta^+(n_{it})} y_a - \sum_{a \in \delta^-(n_{it})} y_a = \begin{cases} x_i & \text{if } t = 0 \\ 0 & \text{if } t \in \{1, \dots, T-1\}, \forall i \in I \\ -x_i & \text{if } t = T, \end{cases} \tag{1c}$$

$$y_a \le u_a \quad \forall a \in A, \tag{1d}$$

$$\sum_{j\in J} z_{jl} e_j \le y_{a_l}(h_l - g_l) \quad \forall l \in L, \tag{1e}$$

$$z_{jl} \le w_{jl} \quad \forall j \in J, \forall l \in L, \tag{1f}$$

$$\sum_{l\in L} z_{jl} \le 1 \quad \forall j \in J, \tag{1g}$$

$$x \in \mathbb{Z}_{\ge 0}^{|I|}, y \in \mathbb{Z}_{\ge 0}^{|A|}, z \in \{0,1\}^{|J|\times|L|}. \tag{1h}$$

The objective function (1a) maximizes the potential total profit from both carsharing and ride-hailing operations, which is also equivalent to maximizing the total number of Type 1 and Type 2 requests served in the system. Constraint (1b) allows the total number of vehicles used in the CRS system to be no more than K. Constraints (1c) are flow balance constraints based on the spatial-temporal network. Constraints (1d) are arc capacity constraints. Constraints (1e) ensure that the accepted carsharing requests from all Type 1 drivers can potentially provide sufficient time to serve accepted ride-hailing requests from all Type 2 users. Constraints (1f) ensure that $z_{jl} = 1$ only if a Type 1 driver $l \in L$ can serve a Type 2 user $j \in J$. Constraints (1g) ensure that each ride-hailing request will be served at most once.

3.3. Phase II: Ride-hailing routing and scheduling

After solving **P1**, we obtain solutions as the set of Type 1 drivers to accept (i.e., all the $l \in L$ with $y_{a_l} = 1$ in the optimal solution). We also locate vehicles in zones following the solution of the x-variables, but discard the values of the z-variables. We then construct a Phase II model to find the routing and scheduling decisions for both Type 1 and Type 2 requests that can be accepted. Note that our final service acceptance and driver–user matching decisions will be made after Phase II, which could be different from the estimated solutions in Phase I. In practice, both types of users will be notified whether we can provide carsharing or ride-hailing service to them after solving Phase II.

We define a network based on the CRS service region as $G' = (V, E)$ where V is the node set and $E = \{(u,v) : u, v \in V, u \ne v\}$ is the edge set. Let L' be the set of accepted Type 1 reservations (given by the subset of drivers in L with $y_{a_l} = 1$ in the optimal solution of Phase I), and $V = V_0 \cup V_1 \cup V_2$ where $V_0 = \{o_l : l \in L'\}$ and $V_1 = \{d_l : l \in L'\}$ are the sets of pick-up and return locations for approved Type 1 carsharing requests from solving P1, respectively. Set $V_2 = \{v_j : j \in J\}$ contains each Type 2 ride-hailing request. We assume that all the arcs can be traveled along both directions, but can have different travel time, and therefore G' is directed. We use c_{uv} to denote the estimated travel time between two nodes such that $(u,v) \in E$. For each $v_j \in V_2, j \in J$, we calculate the estimated travel time in the following way: We set $c_{u,v_j} = c_{u,o'_j}$ and $c_{v_j,u} = c_{d'_j,u}$ for all $u \in V$, where o'_j, d'_j represent the origin and destination of Type 2 user $j \in J$.

We define binary decision vector $\alpha = (\alpha^l_{uv}, (u,v) \in E, l \in L')^T$ such that $\alpha^l_{uv} = 1$ indicates that Type 1 driver $l \in L'$ travels along arc (u, v), and 0 otherwise. We define binary vector $\beta = (\beta_j, j \in J)^T$ such that $\beta_j = 1$ indicates that Type 2 user $j \in J$ is served, and 0 otherwise. We define continuous decision vector $\gamma = (\gamma_v, v \in V)^T$ such that $\gamma_v \ge 0$ is the planned time of a vehicle arriving at location $v \in V$. We formulate the Phase II problem as follows, which is a variant of the VRP with time windows and multiple depots:

$$[\mathbf{P2}]\ \text{maximize}_{\alpha,\beta,\gamma} \quad \sum_{j\in J} r_j^{\text{ride}} e_j \beta_j, \tag{2a}$$

$$\text{subject to} \quad \sum_{l\in L'}\sum_{u\in V} \alpha^l_{uv_j} = \beta_j \quad \forall v_j \in V_2, \tag{2b}$$

$$\sum_{v:(o_l,v)\in E} \alpha^l_{o_l v} - \sum_{v:(v,o_l)\in E} \alpha^l_{v o_l} = 1 \quad \forall o_l \in V_0, l \in L', \tag{2c}$$

$$\sum_{u:(u,d_l)\in E} \alpha^l_{u d_l} - \sum_{u:(d_l,u)\in E} \alpha^l_{d_l u} = 1 \quad \forall d_l \in V_1, l \in L', \tag{2d}$$

$$\sum_{u:(u,v)\in E} \alpha^l_{uv} - \sum_{u:(v,u)\in E} \alpha^l_{vu} = 0 \quad \forall v \in V_2, l \in L', \tag{2e}$$

$$\alpha^l_{uv} = 0 \quad \forall u \in V_0, v \in V, u \ne o_l, l \in L', \tag{2f}$$

$$\gamma_{o_l} + c_{o_l v} - T\left(1 - \alpha^l_{o_l v}\right) \le \gamma_v \quad \forall o_l \in V_0, (o_l, v) \in E, \tag{2g}$$

$$\gamma_{v_j} + e_j + c_{v_j u} - T\left(1 - \sum_{l\in L'} \alpha^l_{v_j u}\right) \le \gamma_u \quad \forall v_j \in V_2, (v_j, u) \in E, \tag{2h}$$

$$g_l \le \gamma_{o_l} \le \gamma_{d_l} \le h_l \quad \forall l \in L', \tag{2i}$$

$$g'_j \le \gamma_{v_j} \le h'_j \quad \forall v_j \in V_2, \tag{2j}$$

$$\alpha \in \{0,1\}^{|E|\times|L'|}, \beta \in \{0,1\}^{|J|}, \tag{2k}$$

where the objective function (2a) maximizes the total profit from providing ride-hailing services. Constraints (2b) ensure that $\beta_j = 1$ if the origin o'_j for Type 2 user $j \in J$ has been visited. Constraints (2c)–(2e) balance vehicle flows for each approved Type 1 driver $l \in L'$. Constraints (2f) forbid flows for approved Type 1 driver $l \in L'$ to visit nodes representing pick-up or return location for Type 1 driver $l' \in L'$, for all $l' \ne l$. Constraints (2g)–(2h) formulate the arrival time at each origin v of a Type 2 user to be greater than the arrival time at node u (which is either pick-up location of a Type 1 driver or destination location of another Type 2 user) plus travel and service time, if a trip travels from u to v directly. Constraints (2i)–(2j) ensure that the departure and return time for Type 1 driver, as well as departure time for each Type 2 user, fall into the corresponding time windows.

3.4. Phase II model variant under stochastic travel time and service time

In practice, uncertainties exist that affect the operations of the proposed CRS system. For example, travel time and service time can be random, due to varying road conditions, weather, and traffic. Therefore, the primary task in this section is to propose a two-stage stochastic programming formulation that incorporates random travel and service times for Phase II.

Let $\tilde{c}_{uv}$ be the random travel time along arc $(u, v) \in E$ and $\tilde{e}_j$ be the random service time for ride $j \in J$. Under uncertainty, one or both of the following scenarios could happen: (i) the Type 1 driver may arrive late to pick-up the scheduled Type 2 user; and (ii) the Type 1 driver may return the shared vehicle later than the scheduled return time. Therefore, our goal is to optimize the start time of each ride to maximize the revenue from ride-hailing operation minus the expected penalty cost due to Type 2 users' waiting and system overtimes. (We define the overtime as the sum of the late time of returning vehicles by all Type 1 drivers.) We denote p^{w} and p^{o} as the unit penalty cost of waiting and overtime, respectively.

Let $\bar{e}_j$ be an estimated service time for Type 2 user $j \in J$, which can be taken as the mean value of $\tilde{e}_j$. We revise **P2** and present its stochastic variant as:

$$[\mathbf{SP2}] \quad \underset{\alpha,\beta,\gamma}{\text{maximize}} \sum_{j\in J} r_j^{\text{ride}} \bar{e}_j \beta_j - \mathbb{E}(Q(\alpha, \gamma, \tilde{c}, \tilde{e})), \quad \text{subject to } (2b)-(2k), \tag{3}$$

where $Q(\alpha, \gamma, \tilde{c}, \tilde{e})$ is the total penalty cost of the random waiting time and overtime given solution (α, γ) and uncertainty $(\tilde{c}, \tilde{e})$, and $\mathbb{E}(\cdot)$ denotes the expectation of random variable $\cdot$. To approximate $\mathbb{E}(Q(\alpha, \gamma, \tilde{c}, \tilde{e}))$, we apply the Sample Average Approximation (SAA) method (Kleywegt *et al.*, 2002). The idea is to generate a finite set of samples following the Monte Carlo sampling approach and approximate the expectation of its sample average function.

Let Ω denote the set of all sampled scenarios, and the probability of realizing each scenario is $1/|\Omega|$ when applying the SAA approach, that is to say

$$\mathbb{E}(Q(\alpha, \gamma, \tilde{c}, \tilde{e})) = \sum_{\omega\in\Omega} \frac{1}{|\Omega|} Q(\alpha, \gamma, \tilde{c}(\omega), \tilde{e}(\omega)),$$

where $\tilde{c}(\omega), \tilde{e}(\omega)$ are realizations of $\tilde{c}, \tilde{e}$ in scenario ω, respectively. We define auxiliary decision variables $W_j(\omega)$ as the waiting time for Type 2 ride request $j \in J$, and $O_l(\omega)$ as the overtime for Type 1 driver service $l \in L'$ for each scenario $\omega \in \Omega$. Given a (α, γ)-solution and realized value $(\tilde{c}(\omega), \tilde{e}(\omega))$ of $(\tilde{c}, \tilde{e})$ in scenario $\omega \in \Omega$, we specify the sample-based linear program for computing the value of $Q(\alpha, \gamma, \tilde{c}(\omega), \tilde{e}(\omega))$ as

$$\text{minimize} \sum_{j\in J} p^{\mathrm{w}} W_j(\omega) + \sum_{l\in L'} p^{\mathrm{o}} O_l(\omega), \tag{4a}$$

$$\text{subject to } \gamma_u + \tilde{c}(\omega)_{uv} - T\left(1 - \alpha^l_{uv}\right) \le \gamma_v + W_j(\omega) \quad \forall (u, v) = \left(o_l, o'_j\right) \in E, \tag{4b}$$

$$\gamma_{o'_j} + W_j(\omega) + \tilde{e}_j(\omega) + \tilde{c}_{d'_j o'_{j'}}(\omega) - T\left(1 - \sum_{l\in L'} \alpha^l_{d'_j o'_{j'}}\right) \le \gamma_{o'_{j'}} + W_{j'}(\omega) \quad \forall \left(d'_j, o'_{j'}\right) \in E, \tag{4c}$$

$$\gamma_{o'_j} + W_j(\omega) + \tilde{e}_j(\omega) + \tilde{c}_{d'_j d_l}(\omega) - T\left(1 - \alpha^l_{d l_j d_l}\right) \le \gamma_{d_l} + O_l(\omega) \quad \forall \left(d'_j, d_l\right) \in E, \tag{4d}$$

$$W_j(\omega) \ge 0 \quad \forall j \in J, \tag{4e}$$

$$O_l(\omega) \ge 0 \quad \forall l \in L'. \tag{4f}$$

Here the objective function (4a) minimizes the total penalty cost of all Type 2 users' waiting time and all Type 1 drivers' delay in returning vehicles. Let $S_j(\omega)$ be the actual service starting time for Type 2 user $j \in J$ and $S'_l(\omega)$ be the actual vehicle return time for Type 1 driver $l \in L'$ for each sampled scenario $\omega \in \Omega$, respectively. We have:

- $S_j(\omega) = \gamma_v + W_j(\omega)$ for $v \in V_2$ representing Type 2 user $j \in J$, and sampled scenario $\omega \in \Omega$;
- $S'_l(\omega) = \gamma_v + O_l(\omega)$ for $v \in V_1$ representing Type 1 driver $l \in L'$, and sampled scenario $\omega \in \Omega$.

Note that $S_j(\omega)$ and $S'_l(\omega)$ are related to the right-hand sides of constraints (4b)–(4c) and (4d), respectively. Therefore, for each Type 1 driver $l \in L'$, the corresponding constraint (4b) calculates the actual service start time of its firstly served Type 2 user j and ensures that it is no earlier than the planned time that driver l departs origin o_l plus the travel time from o_l to o'_j. Similarly, each constraint in (4c) propagates this time relationship for all the subsequent Type 2 users who will be served by Type 1 driver l, and ensures that their actual service start time will be no earlier than the actual service start time of their predecessors plus the realized service time and travel time before driver l arrives. Lastly, constraint (4d) calculates the actual time of returning the vehicle by Type 1 driver $l \in L'$ and ensures that it is no earlier than the time of completing service in the last Type 2 user's location plus the travel time to the return location.

4. Solution approaches

4.1. An integer L-shaped approach via driver-based decomposition

To solve **SP2** efficiently, we propose an algorithm based on the integer L-shaped method (Laporte and Louveaux, 1993). We decompose the problem into a relaxed master problem, which contains variables of all the decisions made before realizing the values of $(\tilde{c}, \tilde{e})$, and a series of subproblems with recourse decisions. Unlike the standard decomposition approach that creates a subproblem for each scenario, we will reformulate our problem by first deciding routes for individual drivers and then creating driver-based subproblems. Specifically, the master problem matches and assigns sequences of Type 2 users to each Type 1 driver. Subproblems are then formulated for each Type 1 driver, to find an optimal schedule to pick up assigned Type 2 users, and each subproblem aims to minimize the expected penalty cost of the assigned Type 2 users' total waiting time and the corresponding Type 1 driver's overtime use of a vehicle. We initially set the lower bound of the expected penalty cost for each Type 1 driver as zero. Then we iteratively generate cuts from each subproblem and add them to the master problem to improve the lower bound.

We define decision variables $\theta = (\theta_l, l \in L')^T$, such that θ_l is the optimal objective value (i.e., the optimal expected penalty cost of waiting time and overtime) of the subproblem formulated for Type 1 driver l, for each $l \in L'$. We formulate a relaxed master problem in the current iteration as:

$$[\mathbf{MP}] \quad \underset{\alpha,\beta,\theta}{\text{maximize}} \ \sum_{j\in J} r_j^{\text{ride}} \bar{e}_j \beta_j - \sum_{l\in L'} \theta_l, \tag{5a}$$

$$\begin{aligned} &\text{subject to} \quad (2b)-(2k), \\ &L(\alpha, \theta_l) \geq 0 \quad \forall l \in L', \end{aligned} \tag{5b}$$

where in constraint (5b), $L(\alpha, \theta_l) \geq 0$ denotes the set of cuts for improving the lower bound of θ_l, generated from solving subproblems (described later) in previous iterations. In the above model, constraints (2b)–(2f) enforce sufficient vehicles to cover matched Type 2 users as explained in the previous section. After solving **MP**, we obtain a tentative optimal solution $\bar{\alpha}$ that can be used to recover the routing sequence for each driver $l \in L'$ as follows. We use the non-zero values of $\bar{\alpha}_{uv}^l$ for $v \in V_2$ to obtain the set of Type 2 users assigned to driver $l \in L'$ and then perform a depth-first search to recover the path to visit assigned Type 2 users for each Type 1 driver. For each $l \in L'$, let $(\sigma^l(1), \sigma^l(2), ..., \sigma^l(n_l))$ be such a sequence with n_l denoting the total number of Type 2 users assigned to driver l. Let $\sigma^l(0)$ and $\sigma^l(n_l + 1)$ be the depots where Type 1 driver $l \in L'$ picks up and returns a vehicle. The only decisions left are the planned arrival time at each Type 2 user's pick up location and the planned time for returning each vehicle. Alternatively, for each Type 1 driver $l \in L'$, we can decide the time to allocate in between $\sigma^l(n)$ and $\sigma^l(n+1)$ for $n = 0, ..., n_l$.

Recall that $S_i(\omega)$ is the actual service start time at node i if $i \in V_2$ or actual vehicle return time at node i if $i \in V_1$. The subproblem for each Type 1 driver $l \in L'$ is equivalent to the following linear program:

$$[\mathbf{SUBP}_l] \ \underset{W,O,S,\gamma}{\text{minimize}} \ \frac{1}{|\Omega|}\left(\sum_{i=1}^{n_l} p^{\text{w}} W_{\sigma^l(i)}(\omega) + p^{\text{o}} O_l(\omega)\right), \tag{6a}$$

$$\begin{aligned} &\text{subject to} \ (2i), \ (2j), \\ &S_{\sigma^l(1)}(\omega) \geq \gamma_{\sigma^l(0)} + c_{\sigma^l(0),\sigma^l(1)}(\omega) \quad \omega \in \Omega, \end{aligned} \tag{6b}$$

$$\begin{aligned} S_{\sigma^l(i+1)}(\omega) \geq S_{\sigma^l(i)}(\omega) + e_{\sigma^l(i)}(\omega) + c_{\sigma^l(i),\sigma^l(i+1)}(\omega) \\ \omega \in \Omega, i = 1, ..., n_l, \end{aligned} \tag{6c}$$

$$S_{\sigma^l(i)}(\omega) = \gamma_{\sigma^l(i)} + W_{\sigma^l(i)}(\omega) \quad i = 1, ..., n_l, \omega \in \Omega, \tag{6d}$$

$$S_{\sigma^l(n_l+1)}(\omega) = \gamma_{\sigma^l(n_l+1)} + O_l(\omega) \quad \omega \in \Omega, \tag{6e}$$

$$W_{\sigma^l(i)}(\omega) \geq 0 \quad \forall i = 1, ..., n_l, \omega \in \Omega, \tag{6f}$$

$$O_l(\omega) \geq 0 \quad \forall \omega \in \Omega, \tag{6g}$$

where constraints (6b) ensure that the actual service start time at the first assigned Type 2 user is not earlier than the actual time Type 1 driver leaves a depot, plus the travel time from the depot to the Type 2 user's pick-up location. Constraints (6c) ensure that the actual service start time at the $(i+1)$th node (or actual vehicle return time if such a node represents a depot) of Type 1 driver l is no earlier than the actual service start time at the ith Type 2 user, plus the time for completing the service at the ith Type 2 user and the travel time from the ith user to the $(i+1)$th user. For all scenarios $\omega \in \Omega$, constraints (6d) let the actual time of serving each assigned Type 2 user be the planned arrival time plus the waiting time of the user. Constraints (6e) let the actual vehicle return time be the planned time plus overtime. Both waiting time and overtime variables are non-negative according to constraints (6f) and (6g). These constraints are analogous to constraints (4b)–(4d). Each subproblem $\mathbf{SUBP}_l$ will output an optimal schedule for Type 1 driver $l \in L'$ to serve the assigned Type 2 users (given by **MP**), and also the minimum expected penalty cost given the current visiting sequence. Moreover, $\mathbf{SUBP}_l$ $\forall l \in L'$ are linear programs without big-M coefficients, which can be solved very efficiently to return cuts (described below) to **MP**.

After solving **MP**, we obtain an integer solution $\bar{\alpha}$ as well as solutions $\bar{\theta}_l$ for all $l \in L'$. Let $\hat{\theta}_l$ be the optimal objective obtained by solving $\mathbf{SUBP}_l$ for each $l \in L'$. If $\bar{\theta}_l < \hat{\theta}_l$, following the integer L-shaped method (see, e.g., Laporte *et al.*, 2002), we propose to add the following cut to the **MP** and re-solve it:

$$\theta_l \geq \hat{\theta}_l \left(\sum_{u,v:\bar{\alpha}_{uv}^l=1} \alpha_{uv}^l - \sum_{u,v} \alpha_{uv}^l + 1 \right). \tag{7}$$

Theorem 1. *Cut (7) is a valid inequality for **MP** and enforces that no same values of $\bar{\alpha}, \bar{\theta}_l$ can be obtained in future iterations.*

Proof. Inequality (7) only takes effect when $\sum_{u,v:\bar{\alpha}_{uv}=1} \alpha_{uv}^l - \sum_{u,v} \alpha_{uv}^l \geq 0$, which is equivalent to $\alpha_{uv}^l = 1$ for all u, v with $\bar{\alpha}_{uv}^l = 1$. Each subproblem $\mathbf{SUBP}_l$ outputs the optimal scheduling for Type 1 driver l and calculates the minimum expected penalty cost, θ_l. The variable representing the recourse penalty cost for the current visiting sequence, should be bounded below by $\hat{\theta}_l$. Therefore, (7) is valid for any optimal (α, θ_l), and the same value of $(\bar{\alpha}, \bar{\theta}_l)$ will not repeat if it is not optimal. □

We summarize the algorithmic steps of the decomposition-based cutting-plane algorithm in Algorithm 1.

Algorithm 1 An integer L-shaped method for solving **SP2**.

1: Initialize **MP** as (5a)–(5b) and set $L(\alpha, \theta_l) \geq 0 = \emptyset$ for all $l \in L'$.
2: Solve **MP** by branch-and-bound to obtain a tentative solution $(\bar{\alpha}, \bar{\theta})$.
3: Recover route sequence $(\sigma^l(1), \sigma^l(2), ..., \sigma^l(n_l))$ for each $l \in L'$ from $\bar{\alpha}$.
4: Solve $\mathbf{SUBP}_l$ for each route sequence of driver $l \in L'$, and let $\hat{\theta}_l$ be the optimal objective for $\mathbf{SUBP}_l$.
5: **if** $\sum_{l \in L'} \bar{\theta}_l < \sum_{l \in L'} \hat{\theta}_l$ **then**
6:     **for** $l \in L'$ **do**
7:         Add Cut (7) to the cut set $L(\alpha, \theta_l) \geq 0$ in **MP**.
8:     **goto** Step 2.
9: **else**
10:     Store the solution $\bar{\gamma}$ from each $\mathbf{SUBP}_l$ as planned service start time for each assigned Type 2 user to driver l.
11:     Return solution $(\bar{\alpha}, \bar{\gamma}, \bar{\theta})$ as an optimal solution to the overall **SP2** problem.

Theorem 2. *Algorithm 1 converges in finite steps.*

Proof. In each iteration, we have $\bar{\theta}_l < \hat{\theta}_l$ only if the current visiting sequence for driver $l \in L'$ has not been explored; otherwise cut (7) enforces $\bar{\theta}_l \geq \hat{\theta}_l$. Since we have a finite number of visiting sequences for a given Type 2 to Type 1 assignment, the algorithm converges in finite steps. □

Remark 1. As an alternative way to solve **SP2**, we can decompose the problem by scenario and construct a relaxed master problem that matches Type 1 drivers to Type 2 users and also determines pick-up routes and schedules. Then the subproblems compute the penalty cost of Type 2 users' waiting time and Type 1 drivers' overtime in each scenario. However, this traditional way of decomposing the problem cannot produce sufficiently good routes and schedules in the master problem under uncertain travel times and service times, which can only be realized in the subproblems. Furthermore, Algorithm 1 avoids "big-M" constraints in both **MP** and $\mathbf{SUBP}_l$, for all $l \in L'$. Due to these two reasons, our decomposition approach significantly improves the computation of **SP2**, which we will demonstrate later through numerical studies.

We may further decompose $\mathbf{SUBP}_l$ by scenario. Since $\mathbf{SUBP}_l$ only contains continuous decision variables and is a linear program that can be quickly solved to obtain cut (7), we directly solve it without further decomposition in our numerical studies.

4.2. Benchmark approaches

To benchmark our proposed algorithm for **SP2**, we compare its computational results against two other approaches for solving **SP2**. The first approach is to solve the deterministic mixed-integer linear programming equivalent of **SP2** using a general commercial optimization solver. We allow the solver to detect possible decomposable structures itself and use any algorithm routines embedded to handle the special structure of the problem. However, we do not specify which algorithms to use when directly solving the deterministic equivalent formulation.

The second approach is to apply the Benders decomposition approach (see Birge and Louveaux, 2011) to solve the sample-based reformulation of **SP2**; it starts with a relaxed master problem containing only first-stage decision variables (made before realizing the values of uncertainties) and iteratively solves subproblems with recourse variables to generate and add cuts to the relaxed master problem. The cuts are used to approximate the value function of the second-stage recourse problem in terms of first-stage decisions.

We detail the Benders decomposition procedures as follows. Let θ be a decision variable to approximate the value function of recourse cost. For **SP2**, we start with the relaxed master problem being:

$$[\textbf{RMP-Benders}] \quad \underset{\alpha,\beta,\gamma,\theta}{\text{maximize}} \sum_{j \in J} r_j^{\text{ride}} \bar{e}_j \beta_j - \theta, \tag{8a}$$

$$\begin{aligned} &\text{subject to } (2\text{b})-(2\text{k}) \\ &B(\alpha, \beta, \gamma, \theta) \geq 0, \end{aligned} \tag{8b}$$

where constraints (8b) contains a set of Benders cuts that will be generated in later iterations. Initially, we only have $\theta \geq 0$. Note that the **RMP-Benders** is a mixed-integer linear program.

Subproblems are linear programs that are use to compute the recourse cost for each sampled scenario, i.e., the penalty cost of the waiting time and overtime given the routing and scheduling decisions from **RMP-Benders**. For each scenario $\omega \in \Omega$, the recourse cost can be computed by (4a)–(4f). Here, we consider subproblems in the dual form of (4a)–(4f) by defining dual variables $\pi(\omega), \mu(\omega), \nu(\omega)$ associated with constraints (4b)–(4d), respectively. We formulate subproblems in their dual form for each scenaraio ω and iteration ν as:

$$\begin{aligned} [\mathbf{SUBP}_B^{\nu}(\omega)] \quad \underset{\pi,\mu,\nu}{\text{minimize}} &\sum_{(o_l, o'_j) \in E} (\gamma_{o_l} + \tilde{c}(\omega)_{o_l, o'_j} - T(1 - \alpha^l_{o_l, o'_j}) \\ &- \gamma_{o'_j}) \pi_{o_l, o'_j}(\omega) \\ + \sum_{(d'_j, o'_{j'}) \in E} &(\gamma_{o'_j} + \tilde{e}_j(\omega) + \tilde{c}_{d'_j o'_{j'}}(\omega) - T(1 - \sum_{l \in L'} \alpha^l_{d'_j o'_{j'}}) - \gamma_{o'_{j'}}) \mu_{d'_j, o'_{j'}}(\omega) \\ + \sum_{(d'_j, d_l) \in E} &(\gamma_{o'_j} + \tilde{e}_j(\omega) + \tilde{c}_{d'_j d_l}(\omega) - T(1 - \alpha^l_{d'_j d_l}) - \gamma_{d_l}) \nu_{d'_j, d_l}(\omega), \end{aligned} \tag{9a}$$

$$\begin{aligned} \text{subject to } &\sum_{l \in L'} \pi_{o_l, o'_j}(\omega) + \sum_{j' \in J} \left(\mu_{d'_{j'}, o'_j}(\omega) - \mu_{d'_j, o'_{j'}}(\omega) \right) \\ &- \sum_{l \in L'} \nu_{d'_j, d_l}(\omega) \leq p^{\text{w}} \quad \forall j \in J, \end{aligned} \tag{9b}$$

$$\sum_{j \in J} \nu_{d'_j, d_l}(\omega) \leq p^{\text{o}} \quad \forall l \in L', \tag{9c}$$

$$\pi(\omega), \mu(\omega), \nu(\omega) \geq 0, \tag{9d}$$

Our problem has complete recourse as every feasible solution from the **RMP-Benders** leads to a finite objective value of each subproblem. Therefore, we only consider the optimality cuts when approximating the recourse costs. Following strong duality, the recourse cost $Q(\alpha, \gamma, \tilde{c}(\omega), \tilde{e}(\omega))$ is equal to the optimal objective value of **SUBP** $^{\nu}_{B}(\omega)$ in each iteration ν and for each scenario ω. Let $(\pi^\nu(\omega), \mu^\nu(\omega), \nu^\nu(\omega))^T$ be an optimal solution to **SUBP** $_B(\omega)$ in the νth iteration of the Benders approach. If we have not closed the optimality gap, we add the following Benders cut into **RMP-Benders**:

$$\theta \geq \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\Bigg(\sum_{(o_l,o'_j)\in E}(\gamma_{o_l}+\tilde{c}(\omega)_{o_l,o'_j}-T(1-\alpha^l_{o_l,o'_j})-\gamma_{o'_j})\pi^\nu_{o_l,o'_j}(\omega)$$
$$+\sum_{(d'_j,o'_{j'})\in E}(\gamma_{o'_j}+\tilde{e}_j(\omega)+\tilde{c}_{d'_jo'_{j'}}(\omega)-T(1-\sum_{l\in L'}\alpha^l_{d'_jo'_{j'}})-\gamma_{o'_{j'}}(\omega))\mu^\nu_{d'_j,o'_{j'}}(\omega)$$
$$+\sum_{(d'_j,d_l)\in E}(\gamma_{o'_j}+\tilde{e}_j(\omega)+\tilde{c}_{d'_jd_l}(\omega)-T(1-\alpha^l_{d'_jd_l})-\gamma_{d_l})\nu^\nu_{d'_j,d_l}(\omega)\Bigg). \tag{10}$$

Algorithm 2 describes detailed algorithmic steps of the Benders approach.

Algorithm 2 The Benders decomposition method for solving **SP2**.

1: Set $\nu = 0$;
2: Initialize **RMP-Benders** with constraints (8b) only containing $\theta \geq 0$; set DONE = False.
3: **while** DONE $\neq$ True **do**
4: $\quad \nu = \nu + 1$;
5: $\quad$ Solve **RMP-Benders** to obtain optimal solutions $\alpha^\nu, \gamma^\nu, \theta^\nu$.
6: $\quad$ Solve **SUBP** $^{\nu}_{B}(\omega)$ for each $\omega \in \Omega$ to obtain optimal solutions $\pi^\nu(\omega), \mu^\nu(\omega), \nu^\nu(\omega)$.
7: $\quad$ **if** $\theta = \theta^\nu$ does not satisfy inequality (10) **then**
8: $\quad\quad$ Add Cut (10) to (8b) and update **RMP-Benders**.
9: $\quad$ **else**
10: $\quad\quad$ Let DONE = True.
return Optimal solutions $\alpha^\nu, \gamma^\nu, \theta^\nu$.

5. Numerical results

We conduct numerical studies and demonstrate the performance of the proposed models using randomly generated instances based on statistical and census data of underserved populations in Washtenaw County, Michigan.

5.1. Experiment design

We generate test instances based on the most updated United States Census data for Washtenaw County in Michigan, collected in April 2010 (see United States Census Bureau, 2010). In the dataset, Washtenaw County is divided into 100 different census tracts, and each contains information on the population and location (i.e., longitude and latitude of the geographical center). We assume that each census tract is represented by its geographical center and construct a corresponding network with 100 nodes. The travel time between any of two nodes can be obtained by accessing Google Map API.[1] In the original dataset, we can get the population for each census tract grouped by different ages. Under safety concerns, we enforce 21–65 years old as the age requirement for Type 1 drivers to match the age requirement of most carsharing service providers (e.g., Zipcar requires drivers to be at least 21 years old).[2] To simulate Type 2 users in underserved populations, we consider the populations with age over 50 and with disability.

- **Problem Size:** To pick the carsharing service zones in set I, we sort the 100 nodes based on the population of Type 1 drivers in each node and select the first $|I|$ nodes. We set $|I| = 5$, and assume that the operational hours of our system is from 7 am to 7 pm, and thus $T = 12$ hours.
- **Type 1 Driver Data:** We simulate the set of Type 1 driver reservations, L, as follows. Consider $|L| = 40$ or 60. For each Type 1 driver reservation $l \in L$, we simulate the pick-up and return locations, o_l and d_l, from the node set I based on the density of populations that are jobless or whose annual incomes are below a certain threshold. For the car reservation time s_l of each Type 1 driver $l \in L$, we sample s_l uniformly from 7 am to 4 pm with a 1-hour time interval. The time of the private use of a car is then uniformly sampled from 0–2 hours with a 1-hour time interval. We simulate Type 1 drivers' different flexibile levels of serving Type 2 users, and consider two types of service-hour distributions: For Case (i), we generate the service hours for Type 1 drivers from $\{1, 2, 3, 4\}$ with equal probability 0.25. For Case (ii), we sample the service hours for Type 1 drivers from $\{1, 2, 3, 4\}$ with probabilities 0.1, 0.2, 0.3, and 0.4, respectively. For the rest of this article, we refer to Case (i) as the case with "regular drivers" and to Case (ii) as the case with "flexible drivers". Except for the tests in Section 5.2.2 and Section 5.3.1, we report the results for the case with "flexible drivers".
- **Type 2 User Data:** We consider instances with $|J| = 40, 60, 80$ for Type 2 ride-hailing demand. For each Type 2 user $j \in J$, we sample the origin node, o'_j, based on the population density of target Type 2 users (age over 50 years old and disabled) and sample the destination node, d'_j, uniformly over the 100 census tracts. At the same time, we avoid $o'_j = d'_j$ in all our sampled instances by re-sampling the destination node if it happens. We uniformly generate g'_j for each Type 2 user over the entire time of operation (i.e., 7 am to 7 pm) and assign a 30-minute time window for each pick-up, i.e., $h'_j = g'_j + 30, \forall j \in J$.
- **Stochastic Travel Time:** In all our test instances, for each Type 2 reservation $j \in J$, we set $e_j = c_{o'_j,d'_j}$ plus a constant loading/unloading time that is the same for all $j \in J$ and thus can be omitted in the model. Following the standard VRP literature, such as Polus (1979) and Laporte *et al.* (1992), we consider a Gamma distribution for sampling the random travel time between pairs of locations. We assume that $\tilde{c}_{ij} = c_{ij}(0.8 + \xi)$ where ξ follows a Gamma distribution with shape parameter α and

Table 1. Summary of key parameters.

Basic Settings			
$\|I\|$	5	$\|L\|$	{40,60}
T	12	$\|J\|$	{40,60,80}
$\|\Omega\|$	500		
Stochastic Travel Time			
c_{ij}	average travel time obtained from Google Map API		
ξ	Follow Gamma Distribution with parameters $\alpha = 0.2$, $\lambda = 1$		
$\tilde{c}_{ij}$	$c_{ij}(0.8 + \xi)$		
Other Parameters			
r^{car}	\$8/hour	c^{idle}	\$1/hour
r^{ride}	\$20/hour	p^{w}	\$0.1/hour
r^{drive}	\$16/hour	p^{o}	\$0.1/hour
c^{ser}	\$1/hour		

Table 2. CPU time (in seconds) for models **P1** and **P2.**

			P1			*P2*		
K	$\|L\|$	$\|J\|$	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*
20	40	40	0.02	0.01	0.01	2.18	1.56	1.91
20	40	60	0.02	0.01	0.01	59.02	5.78	23.05
20	40	80	0.02	0.02	0.02	176.65	18.05	76.65
30	60	40	0.02	0.01	0.02	18.58	5.36	9.65
30	60	60	0.02	0.02	0.02	31.34	12.23	20.51
30	60	80	0.03	0.02	0.03	483.93	33.91	256.95

scale parameter λ, and c_{ij} is the deterministic travel time obtained from Google Map API for arc $(i,j) \in E$. Then following the definition of the Gamma distribution, we have:

$$E(\tilde{c}_{ij}) = (0.8 + \alpha\lambda)c_{ij},$$
$$Var(\tilde{c}_{ij}) = \alpha\lambda^2 c_{ij}$$

We choose parameters $\alpha = 0.2$ and $\lambda = 1$ to generate all the scenarios.

- **Other Parameters:** We set $r^{car} = 8$ per hour and $r^{ride} = 20$ per hour to match the price of current carsharing and ride-hailing service, e.g., Zipcar and Uber, in Washtenaw County. Note that the pricing strategy for a ride-hailing company is composed by two parts, mileage (\$0.95/mile) and service time (\$0.15/minute) in Washtenaw County.[3] Here we combine them together to obtain reasonable price settings. We set $r^{drive} = 16$ per hour to reasonably incentivize Type 1 drivers to provide ride-hailing service. Also, $c^{ser} = 1$ per hour and $c^{idle} = 1$ per hour. The penalty cost parameters are set as $p^{w} = 0.1$ per minute per user and $p^{o} = 0.1$ per minute per car for model **SP2**.

Table 1 summarizes the important parameters used in the numerical studies. For each test instance described above, we generate five replications and report the average statistics unless otherwise noted.

All instances are programmed using Java 10. We call the solver Gurobi 8.0 to optimize all mixed-integer linear programming models. All programs are run on a desktop computer with Microsoft Windows 10 64-bit operating system, an Intel Core i7-6700K Central Processing Unit (CPU) with 4.0 GHz, and 32.0 GB RAM.

5.2. Computational results

For our numerical experiments, we analyze our proposed model from the aspects of computational time, quality of service, revenue and cost, and out-of-sample tests.

5.2.1. Computational time

We present the computational results of the proposed model for various test instances, with five replications for each parameter combination (with the same number of users but different input data, e.g., locations, pick-up time windows). For each instance, we report the maximum, minimum, and average CPU time. The average CPU time is calculated based on solved replications. We choose parameters as described in Section 5.1 and 500 scenarios for the stochastic programming model **SP2**, and set the CPU time limit as 30 minutes for computing each replication.

In Tables 2 and 3, we report the CPU time (in seconds) of models **P1**, **P2**, and **SP2** for solving each test instance with different numbers of available vehicles (K), number of Type 1 reservations ($|L|$), and number of Type 2 reservations ($|J|$). We report the optimality gap for **SP2** when reaching the time limit. For **SP2**, we report results for: (i) directly solving the MILP model (see Columns "SP2 (Direct)"), (ii) applying the Benders decomposition (see Columns "SP2 (Benders)"), and (iii) applying our integer L-shaped method using driver-based decomposition (see Columns "SP2 (L-shaped)").

According to Tables 2 and 3, model **P1** is easy to solve, whereas **P2** and **SP2** are relatively difficult to optimize for both deterministic and stochastic cases. In addition, **SP2** is more challenging to solve compared to **P2**, as four out of six test instance sizes cannot be solved to optimality within time limit when the travel time and service time are both random.

In Table 2, the CPU time for **P2** increases as the number of Type 1 and Type 2 reservations increases. For example, given $|L| = 40$, the average CPU time increases from 1.91 seconds to 76.65 seconds as $|J|$ grows; given $|J| = 60$, the average CPU time increases from 9.65 seconds to 256.95 seconds as $|L|$ grows from 40 to 80. We also demonstrate the effectiveness of the proposed solution approach, as both CPU time and optimality gap have been significantly improved from those given by general-purpose solvers.

Although some instances cannot be solved to optimality when testing **SP2**, the optimality gap can be small when reaching the CPU time limit, i.e., when we set the optimization gap tolerance to 5%, most of the cases can be solved within the time limit. Comparing to the other two solution approaches, our proposed algorithm works well and outputs solutions with the minimum optimality gap. Furthermore, due to the driver-based decomposition, we avoid the out-of-memory issue that appears in the scenario-based Benders decomposition when solving large instances. In later sections, we will see that the number of served trips output by **SP2** is similar to that of **P2** while achieving a much higher quality of service for scheduling. The difficulty of closing the optimality gap for **SP2** is due to the schedule adjustment

Table 3. CPU time (in seconds) or optimality gap for models **SP2** with different methods.

K	|L|	|J|	SP2 (Direct)			SP2 (Benders)			SP2 (L-shaped)		
			Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
20	40	40	38.37%*	0.57%*	10.30%*	2.73%*	1.22%*	1.84%*	0.20%*	113.03	317.51
20	40	60	N/A	N/A	N/A	3.26%*	2.25%*	2.77%*	1.43%*	0.59%*	0.98%*
20	40	80	N/A	N/A	N/A	5.21%*	3.92%*	4.53%*	5.03%*	2.65%*	3.44%*
30	60	40	N/A	1.33%*	37.87%*	1.59%*	1.07%*	1.28%*	0.14%*	120.42	247.97
30	60	60	N/A	N/A	N/A	2.38%*	1.51%*	1.92%*	0.68%*	0.22%*	0.45%*
30	60	80	N/A	N/A	N/A	–	–	–	6.81%*	1.53%*	4.00%*

*: optimality gap is reported; N/A: no feasible solutions found within the 30-minute time limit; –: model is not solvable due to machine memory limit.

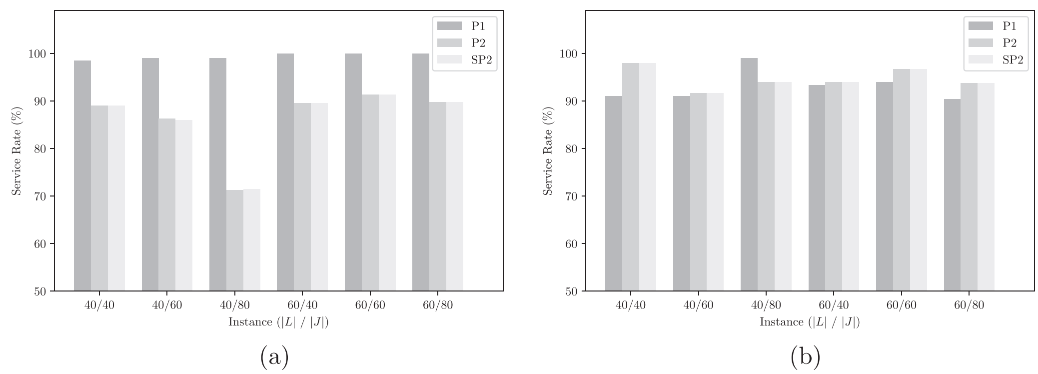

Figure 2. Average demand service rates by proposed models (a) case with "regular drivers", and (b) case with "flexible drivers".

that minimizes the expected penalty cost. Therefore, despite that **SP2** cannot be solved to optimality, we can still use the output solutions of matched Type 1 drivers and Type 2 users, and the sequence of serving Type 2 users to schedule the operations in Phase II. Based on the reasonable instance size of the designed system, our tests show the feasibility and practicability of the proposed models.

5.2.2. Quality of service

Here we show the quality of service (QoS) results of the proposed models based on percentage of approved demands. As mentioned in Section 5.1, we consider two cases of Type 1 drivers: "regular drivers" with service hours generated from $\{1, 2, 3, 4\}$ hours with equal probability and "flexible drivers" with service hours picked from $\{1, 2, 3, 4\}$ hours with probabilities $0.1, 0.2, 0.3, 0.4$, respectively, i.e., the latter drivers are more time flexible and are likely to provide service with extended hours.

Figure 2(a) shows the average demand service rates across different models with "regular drivers" and Figure 2(b) shows the average demand service rates with "flexible drivers". For both cases, the demand service rate for Type 1 drivers is kept at high levels (over 90%). At the same time, both **P2** and **SP2** yield similar demand service rate for Type 2 users. However, for the case of "regular drivers", the demand service rate for Type 2 users varies from 70% to 90%, due to the lack of available drivers/service hours: When the ratio between Type 2 users and Type 1 drivers rises, the demand service rate for Type 2 users drops significantly (see the instance with $|L| = 40, |J| = 80$). On the other hand, the demand service rates for both Type 1 drivers and Type 2 users are very high (over 90%) across all instances for the case with "flexible drivers". Based on these results, the proposed model would be particularly helpful for underserved communities whose residents are more flexible in terms of service hours.

5.3. Separated P1 and P2 versus integrated formulation

To demonstrate the benefits of using the two-phase approach that sequentially solves models in Phases I and II, we compare it with directly solving an integrated model that decides all decisions together (including vehicle allocation, Type 1 and Type 2 users to accept, and the corresponding matching and routing decisions). Specifically, we report the average, maximum, and minimum CPU time comparison and acceptance rates of Type 1 users for each method in Table 4.

In Table 4, the CPU time increases by between 20 and 200% on average when switching from the two-phase method to directly solving the integrated model. For the instance with 60 Type 1 drivers and 80 Type 2 users, the computational time can almost reach 30 minutes. Also, we note that the solution time increases as the average acceptance rate of Type 1 drivers decreases. Moreover, the solutions obtained from the two-phase approach and the integrated model are not significantly different for most of the instances we tested.

5.3.1. Revenue and cost

We first show the revenue and cost composition for a given instance and then extend the discussion across all instances. Figure 3 depicts the revenue and cost composition generated by proposed models for the instance with $|L| = 40, |J| = 40$

Table 4. CPU time (in seconds) and results comparison of the two-phase method and integrated model.

			Time (Two-Phase)			Time (Integrated)			Type 1 Accept.		
K	|*L*|	|*J*|	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max (%)*	*Min (%)*	*Avg (%)*
20	40	40	2.20	1.57	1.92	3.06	1.70	2.50	98	83	92
20	40	60	59.04	5.79	23.06	65.81	7.43	26.63	100	83	91
20	40	80	176.67	18.06	76.67	190.89	15.91	98.87	100	95	99
30	60	40	18.60	5.37	9.67	27.52	6.45	12.49	100	88	93
30	60	60	31.36	12.24	20.53	82.17	14.82	31.73	97	90	94
30	60	80	483.96	33.93	256.98	1606.20	29.36	505.05	97	82	90

Figure 3. Average revenue and cost compositions for instance with $|L| = 40$, $|J| = 40$ by proposed models: (a) case with "regular drivers" and (b) case with "flexible drivers".

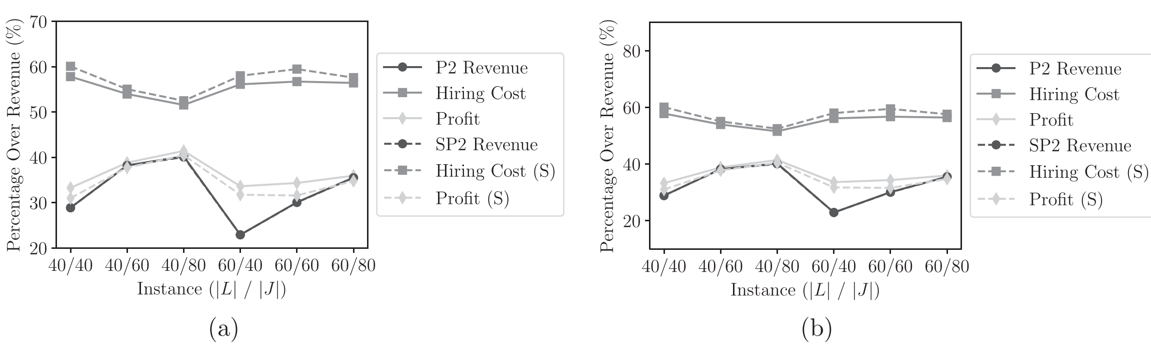

Figure 4. Average revenue and cost percentage for all instances by proposed models: (a) case with "regular drivers" and (b) case with "flexible drivers".

for both "regular drivers" (Figure 3(a)) and "flexible drivers" (Figure 3(b)). In each figure, the top row depicts the case where we only consider deterministic traveling times, and the bottom row demonstrates the results when stochastic traveling times are taken into account. The amounts of revenue generated from serving Type 2 users are similar under **P2** and **SP2**. However, as we consider the variation of traveling and service time, **SP2** utilizes slightly more drivers to provide a reliable service (and correspondingly, the hiring cost proportion increases by 1% and 6%, for the two cases, respectively). Comparing Figure 3(a) with Figure 3(b), the total revenue increases in the latter case from \$1338 to \$1642. At the same time, the hiring cost to serve Type 2 users also increases considerably, whereas the profit proportion drops.

Figure 4 depicts the changes in revenue and cost compositions across all test instances. Both cases show similar effects: Given fixed $|L|$, the proportion for the revenue from serving Type 2 users and also the overall profit increase as $|J|$ increases. However, the proportion of hiring cost for Type 1 drivers decreases as more Type 2 requests appear. Therefore, we infer that our system produces efficient matching and scheduling to accommodate more Type 2 users. Relatively, the proportion of hiring cost is higher for **SP2** than **P2** and the proportion of profit is lower for **SP2** than **P2**. This finding suggests that we need to sacrifice some profit to compensate for a higher quality of scheduling service (which will be further shown in Section 5.3.2). Overall, the proposed system is in good financial health.

5.3.2. Out-of-sample tests

We conduct out-of-sample tests to evaluate the proposed models by measuring Type 2 users' waiting and Type 1 drivers' overtime. We generate the set of out-of-sample test scenarios Ω' following the same distribution of stochastic travel time as discussed in Section 5.1 and use $|\Omega'| = 1000$.

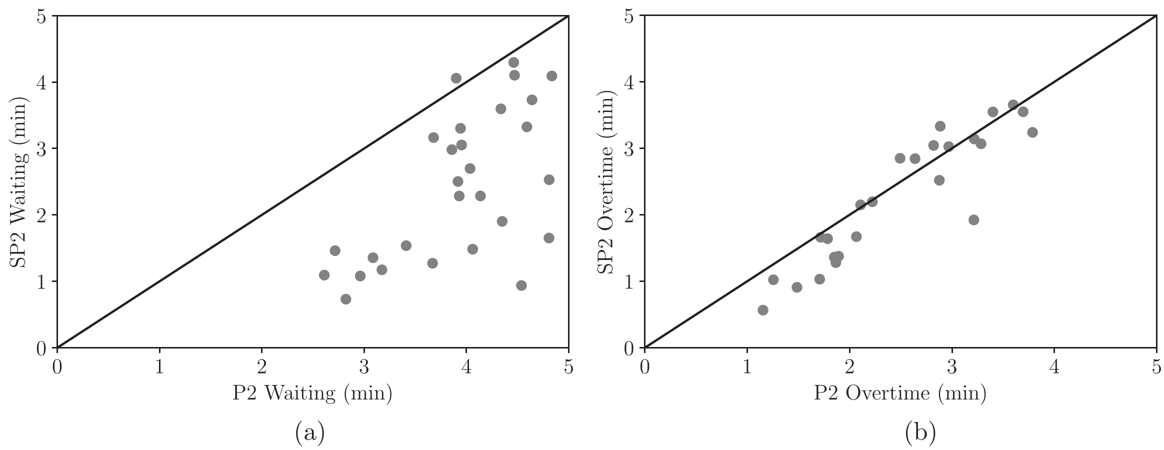

Figure 5. (a) Average waiting time and (b) average overtime per user in the proposed system.

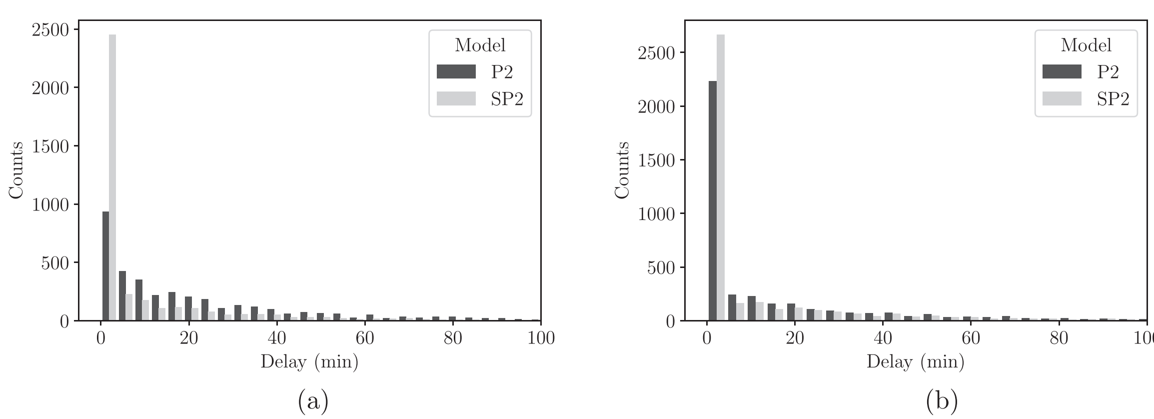

Figure 6. 90%-Tail distributions of average waiting time and overtime per user for instance with $|L| = 40$ and $|J| = 60$.

After performing out-of-sample testing, let $s(l)$ be the scheduled time to return the car by Type 1 driver $l \in L$ and $s'(j)$ be the scheduled starting service time for Type 2 user $j \in J$. Values $s(l), l \in L$ and $s'(j), j \in J$ can be obtained from the optimal solutions of α-variables in **P2** and **SP2**. Let $t(l, \omega)$ be the actual time of returning the car by Type 1 driver $l \in L$ in $\omega \in \Omega'$ and $t'(j, \omega)$ be the actual service starting time for Type 2 user $j \in J$ in $\omega \in \Omega'$, which will be computed based on the routing sequence and random travel and service time in scenario $\omega \in \Omega'$. Then for each test scenario ω, we calculate and report:

- $\text{wait}(j, \omega) = \max\{0, t(j, \omega) - s(j)\}$ for $j \in J$.
- $\text{overtime}(l, \omega) = \max\{0, t'(l, \omega) - s'(l)\}$ for $l \in L$.

Figures 5(a) and 5(b) demonstrate the out-of-sample test results on average waiting time per Type 2 user and average overtime per Type 1 driver by the proposed models. We report the results for all test instances and replications. In Figure 5(a), for **P2**, the average waiting time ranges from 2.5 to 5 minutes per customer, and reduces to 1 to 4 minutes for **SP2**. For most of the cases, the average waiting time decreases significantly as the dots appear far below the 45° line. However, the performance of the models for average overtime per Type 1 driver is mixed. In Figure 5(b), the average overtime per driver ranges from 1 to 4 minutes for both models and the dots are lying around the 45° line, which indicates a slightly better performance than the one of **SP2**. The reason that we do not see an obvious advantage of **SP2** in the overtime performance is that we set the same penalty coefficients for waiting time and overtime. This leads the model to treat waiting time and overtime indifferently and therefore focus on minimizing the overall penalties. If stakeholders weigh more on the overtime, one can adjust the unit penalty coefficients accordingly.

We also demonstrate the detailed out-of-sample performance for one specific instance. Figure 6 shows partial distributions of the average waiting and overtime per user in one test instance with 40 Type 1 drivers and 60 Type 2 users. Due to the significant amount of zero waiting and overtime per user (in more than 90% of all the out-of-sample scenarios), we report the tail distributions of the highest 10% waiting and overtime outcomes. Both Figure 6(a) and Figure 6(b) show long-tail effect that the waiting and overtime are concentrated on small values for most cases. Comparing the performance of the **P2** and **SP2**, **SP2** yields relatively shorter waiting and overtime than **P2**. Similar observations can also be drawn in other instances.

6. Conclusions

In this article, we designed a new shared-mobility system to serve the transportation needs of underserved populations. We integrated both carsharing and ride-hailing, and developed a two-phase approach to design and operate such a system. We evaluated the models on various instances based on synthetic data in Washtenaw County, Michigan, focusing on serving jobless, elderly, and disabled populations. Numerical results indicated the computational efficiency of our proposed solution approaches. Furthermore, the QoS of the system was maintained at high levels.

We further extended the basic model to a two-stage stochastic programming model to capture the randomness of vehicle travel time and service time. Numerical comparisons with a deterministic counterpart using expected values of the random parameters showed the advantages of our models. Both in-sample and out-of-sample test results demonstrated the effectiveness of matching and scheduling using our approach, where the risk of waiting and overtime both decreased.

For future research, our models and results can be extended as follows. First, in this article, we minimize the expected cost of overtime and waiting time. Instead, a robust optimization model that focuses on the worst-case analysis can be used for ensuring reliable operations. Second, we will collaborate with policy makers and social workers for real-world deployment of the CRS system. We aim to make more transportation data of underserved communities available to the public through further investigation of this topic.

Notes

1. https://developers.google.com/maps/documentation/distance-matrix/
2. https://support.zipcar.com/hc/en-us/articles/220333808-Eligibility-Requirements
3. https://www.uber.com/fare-estimate/

Notes on contributors

Miao Yu received a B.S. in mathematics and B.S. in statistics from the University of Minnesota, Twin Cities in 2014. He is currently a Ph.D. candidate in the Department of Industrial and Operations Engineering at the University of Michigan, Ann Arbor. His research interests include stochastic programming and combinatorial optimization with applications in routing and shared mobility problems.

Siqian Shen received a B.S. degree in industrial engineering from Tsinghua University, China, in 2007, and M.S. and Ph.D. degrees in industrial and systems engineering from the University of Florida, USA, in 2009 and 2011, respectively. She is an associate professor in the Department of Industrial and Operations Engineering, University of Michigan at Ann Arbor, and also an Associate Director for the Michigan Institute for Computational Discovery & Engineering. Her research interests include stochastic programming, network optimization, and integer programming. Applications of her work include healthcare, transportation, and energy.

Funding

The authors are grateful for the support from the National Science Foundation under grant CMMI-1636876 and CMMI-1727618.

ORCID

Miao Yu http://orcid.org/0000-0002-7625-6315
Siqian Shen http://orcid.org/0000-0002-2854-163X

References

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

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E. and Rus, D. (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, **114**(3), 462–467.

Birge, J.R. and Louveaux, F. (2011) *Introduction to Stochastic Programming*. Springer, Springer-Verlag New York.

Brandstatter, G., Leitner, M. and Ljubic, I. (2016) Locations of charging stations in electric car sharing systems. Technical report, Department of Statistics and Operations Research, University of Vienna, Vienna, Austria.

Brook, D. (2004) Carsharing–start up issues and new operational models, in *Proceedings of the Transportation Research Board 83rd Annual Meeting*, Washington D.C.

Chan, N.D. and Shaheen, S.A. (2012) Ridesharing in North America: Past, present, and future. *Transport Reviews*, **32**(1), 93–112.

Clewlow, R.R. and Mishra, G.S. (2017) *Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the United States*. Technical Report UCD-ITS-RR-17-07, University of California, Davis, Institute of Transportation Studies, Davis, CA, USA.

De Almeida Correia, G.H. and Antunes, A.P. (2012) Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, **48**(1), 233–247.

Fan, W.D. (2014) Optimizing strategic allocation of vehicles for one-way car-sharing systems under demand uncertainty. *Journal of the Transportation Research Forum*, **53**(3), 7–20.

Firth, S. (2016) Q&A with Detroit's new director of public health. https://www.medpagetoday.com/publichealthpolicy/publichealth/55883. Accessed July 16, 2019.

Gross, M.B., Hogarth, J.M., Schmeiser, M.D. (2012) Use of financial services by the unbanked and underbanked and the potential for mobile financial services adoption. *Federal Reserve Bulletin*, **9**(4), 1–20.

He, L., Mak, H.Y., Rong, Y. and Shen, Z.J.M. (2016) Service region design for urban electric vehicle sharing systems. *Manufacturing & Service Operations Management*, **19**(2), 309–327.

Kek, A.G.H., Cheu, R.L., Meng, Q. and Fung, C.H. (2009) A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, **45**(1), 149–158.

Kleywegt, A.J., Shapiro, A. and Homem-de Mello, T. (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, **12**(2), 479–502.

Laporte, G., Louveaux, F. and Mercure, H. (1992) The vehicle routing problem with stochastic travel times. *Transportation Science*, **26**(3), 161–170.

Laporte, G. and Louveaux, F.V. (1993) The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, **13**(3), 133–142.

Laporte, G., Louveaux, F.V. and van Hamme, L. (2002) An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, **50**(3), 415–423.

Laporte, G., Meunier, F. and Calvo, R.W. (2015) Shared mobility systems. *4OR*, **13**(4), 341–360.

Lu, M., Chen, Z. and Shen, S. (2018) Optimizing the profitability and quality of service in carshare systems under demand uncertainty. *Manufacturing and Service Operations Management*, **20**(2), 162–180.

Martin, C.J. (2016) The sharing economy: A pathway to sustainability or a nightmarish form of neoliberal capitalism? *Ecological Economics*, **121**,149–159.

McLaughlin, K. (2015) Detroit has become such an 'economic desert' that residents now commute up to four-hours each day just to find work. http://www.dailymail.co.uk/news/article-2975237/Dearth-jobs-barrier-post-bankruptcy-Detroits-growth.html. Accessed July 16, 2019.

Millard-Ball, A. (2005) *Car-Sharing: Where and How it Succeeds*. Transportation Research Board, Washington D.C.

Muheim, P. and Reinhardt, E. (1999) Carsharing: The key to combined mobility. *World Transport Policy & Practice*, **5**(3), 58–71.

Nair, R. and Miller-Hooks, E. (2014) Equilibrium network design of shared-vehicle systems. *European Journal of Operational Research*, **235**(1), 47–61.

Nourinejad, M. and Roorda, M.J. (2014) A dynamic carsharing decision support system. *Transportation Research Part E: Logistics and Transportation Review*, **66**, 36–50.

Polus, A. (1979) A study of travel time and reliability on arterial routes. *Transportation*, **8**(2), 141–151.

Shaheen, S., Chan, N., Bansal, A. and Cohen, A. (2015). *Shared mobility: A sustainability & technologies workshop: Definitions, industry developments, and early understanding*. Technical report, Innovative Mobility Research, Berkeley, CA.

Smith, A. (2016) Shared, collaborative and on demand: The new digital economy. Pew Research Center, 19. https://www.pewinternet.org/2016/05/19/the-new-digital-economy/. Accessed July 16, 2019.

Taş, D., Dellaert, N., van Woensel, T. and de Kok, T. (2013) Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, **40**(1), 214–224.

Thebault-Spieker, J., Terveen, L.G. and Hecht, B. (2015) Avoiding the south side and the suburbs: The geography of mobile crowdsourcing markets, in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, Vancouver, BC, Canada, pp. 265–275.

Toth, P. and Vigo, D. (2014) *Vehicle Routing: Problems, Methods, and Applications*. SIAM, Philadelphia, PA.

United States Census Bureau (2010) National census tracts gazetteer. https://www.census.gov/geo/maps-data/data/gazetteer2010.html. Accessed July 16, 2019.

Zhang, Y., Lu, M. and Shen, S. (2018) *On the values of vehicle-to-grid electricity selling in electric vehicle sharing*. Working paper. https://ssrn.com/abstract=3172116. Accessed July 16, 2019.