Designing Optimal Autonomous Vehicle Sharing and Reservation Systems: A Linear Programming Approach

Jiaqi Ma[a], Xiaopeng Li[b], Fang Zhou[a], Wei Hao[c]
a. Transportation & Financial Solutions,
Leidos, Inc., Reston, VA 20190, USA
b. Department of Civil and Environmental Engineering,
University of South Florida, FL 33620, USA
c. University Transportation Research Center,
City College of New York, New York, NY 10031, United States

Abstract

Autonomous vehicle (AV) technology holds great promise for improving the efficiency of traditional vehicle sharing systems. In this paper, we investigate a new vehicle sharing system using AVs, referred to as autonomous vehicle sharing and reservation (AVSR). In such a system, travelers can request AV trips ahead of time and the AVSR system operator will optimally arrange AV pickup and delivery schedules and AV trip chains based on these requests. A linear programming model is proposed to efficiently solve for optimal solutions for AV trip chains and required fleet size through constructed AVSR networks. Case studies show that AVSR can significantly increase vehicle use rate (VUR) and consequentially reduce vehicle ownership significantly. In the meantime, it is found that the actual vehicle miles traveled (VMT) in AVSR systems is not significantly more than that of conventional taxis, despite inevitable empty hauls for vehicle relocation in AVSR systems. The results imply huge potential benefits from AVSR systems on improving mobility and sustainability of our current transportation systems.

1 Introduction

Privately owned vehicles provide incomparable mobility, flexibility, and freedom to travel. The private auto mode constitutes over 83% of the total passenger trips in the U.S.; however, they pose great challenges to transportation sustainability. Every year in the U.S., private vehicles are a major contributor to approximately 17% of household expenses allocated to transportation, 70% of the total petroleum consumption, and 30% of greenhouse gas emissions (Bureau, 2014). Additionally, private vehicles are left unused for 23 hours a day (Litman, 2007) and the increased parking occupies 25% of urban surfaces (Jakle and Sculle, 2004). Public transit systems have the potential to overcome these difficulties, but they may not have as high service quality and flexibility, e.g., passenger discomfort and difficulty in accessibility (Sinha, 2003).

Vehicle sharing is an alternative to private vehicle ownership. A group of people collectively owns a number of spatially distributed vehicles (Cooper et al., 2000). This mode provides a comfort level similar to that of private vehicles and also reduces ownership significantly. Over the past decade in North America, the number of shared vehicles has increased from under 700 to over 15,000, and the number of people who use this service has grown from 16,000 to over a million (Shaheen and Cohen, 2013). Well-known vehicle sharing services include Zipcar (http://www.zipcar.com/), JustShareIt (http://www.justshareit.com/), Autolib (https://www.autolib.eu/en/) and City Car Club (http://www.citycarclub.co.uk/). Vehicle sharing has become a major transportation mode with high spatial accessibility and holds the promise of a future sustainable transportation system with high vehicle use rates, minimum land occupancy, significant cost savings, and likely environmental and social benefits (Millard-Ball, 2005). Traditional vehicle sharing, however, still faces one major challenge that prevents it from being widely used among the public: nearby vehicle availability. If no vehicles are nearby, a person may be stranded, thus having to wait a long time or walk a great distance. Thus, under such circumstances this person may not continue to use shared vehicles for future travels.

The emerging technologies of mobile communications and autonomous vehicles (AV) have the potential to address the previously mentioned concerns. Through connectivity using certain mobile devices, travelers can request vehicles that are relatively far away before traveling, so that they do not need to walk long distances to available vehicles. Also, different from current on-demand ride-hailing services (excluding ride-sharing), AVs can be fully self-driving and can relocate themselves automatically to any traveler’s location upon request without human operations.

Autonomous vehicle sharing (AVS) has the potential to provide significant environmental and mobility benefits, particularly in reducing vehicle ownership and parking demands. Using an agent-based simulation approach, Fagnant and Kockelman (2014) indicate that each shared AV can replace 11 conventional vehicles though increasing Vehicle Miles Traveled (VMT) by 10%, and the sharing system results in overall benefits in regards to emissions. These benefits are approved using an agent-based simulation with pre-specified agent rules, such as how unoccupied AVs should be relocated to other zones to meet potential future demands and reduce passenger waiting times. Further, if trip demands in the next period (e.g., a day, three hours, or one hour) are known to the AVS system operator, it is possible to optimally plan AV pickup and delivery routes for all travelers ahead of time. The resultant optimal AV trip chains have the potential to provide best system performance.

In this paper, we investigate a new vehicle sharing system, referred to as autonomous vehicle sharing and reservation (AVSR). In such a system, travelers can request AVs for trips ahead of time and the AVSR system operator will optimally arrange the AV pickup and delivery before requested time and design AV trip chains on the basis of the recorded trip demand requests. Particularly, instead of relocating unoccupied AVs heuristically (e.g., to areas with less unoccupied AVs at current time), all AV routes and schedules for the next planning horizon can be optimally planned and determined. Also note that, if optimally designed, AVSR systems can provide upper bound of benefits for different AVS systems, including current on-demand ride-hailing systems (ride sharing excluded). An AVSR system has the following characteristics:

1. A fleet of AVs is distributed and shared by different users over the road network.
2. Each AV serves a passenger or a group of passengers with the same trip at a time. No ride sharing is considered. For a group of passengers with the same trip (i.e., same OD and departure time), if the number of passengers N^P in a group exceeds a vehicle's capacity C, the system will automatically consider this as ceil(N^P/C) subgroups and dispatch one AV for each subgroup to meet these "separate" demands. For implementation, users can specify the number of passengers when making requests, or consider this constraint when specifying the number of vehicles to be reserved. In this paper, we assume that the subgroups have already been generated beforehand by considering the sizes of passenger groups and vehicle capacity.
3. Each trip – a pickup and delivery – is served without any interruption from other pickup and delivery jobs; thus in this paper, AVSR only considers vehicle sharing, and does not consider ride sharing.
4. There is a hard time window specified by users – the latest pickup time. This time can be the latest departure time of the user, or the user will complain if the taxi is late. Predicative technologies are available to AVSR operators to estimate potential travel times using a selected path.
5. Users make requests ahead of time, but with different request horizon options. They can request a vehicle one day before or one hour ahead of the travel. Users need to enter locations for pickup and delivery and preferred pickup time. For requests with short horizons, it is necessary to have an efficient algorithm for near real-time trip chain planning. AVSR planning needs to consider both future trips and trips that are currently being served (due to the availability of AVs at a later time when the current trip is completed).

In addition to the above AVSR characteristics, there are a few more assumptions that are made for this paper. It is assumed that each AV only returns to the depot at the end of the day for basic maintenance and preparation for the next day's service. For modeling simplicity, it is assumed that there are two virtual depots – an origin depot and a destination depot. All AVs are assigned from the origin depot and collected at the destination depot at the end of the day. This assumption does not affect the formulation and is only for modeling simplicity, and can be relaxed easily to include multiple real repots at different locations (e.g., by adding certain depot nodes and corresponding links connecting these depots and relevant trips). This will be further discussed in the model formulation section. It is also assumed that empty AVs will always find a place for temporary parking during the day, either at certain designated locations or the next pickup location, before picking up the next traveler. An AVSR system can even decide where to park with consideration of other constraints (e.g., traffic congestion) to ensure the service quality is guaranteed.

The AVSR systems can be considered as a vehicle sharing or autonomous taxi service, depending on how the service is provided. If the former, the AVs are owned collectively by users/travelers through certain mechanisms and a third-party company will be responsible for vehicle maintenance and dispatching. If the latter, AVs can be owned by a taxi company and AVSR is offered as an advanced service. AVSR, however, is different from an on-demand taxi and ride-hailing service as there is no competition between vehicles and it involves no driver decision on accepting or rejecting requests. AVSR is aimed at optimal system performance while ensuring travelers are picked up before the requested pickup time.

2 Literature Review

To the best of the research team's knowledge, there is no research work investigating the proposed AVSR or similar systems using AVs. However, vehicle sharing, particularly on the policy side, has been extensively studied and much research on autonomous vehicle sharing, autonomous taxis, or other similar services has been seen in recent literature. This section reviews existing research on these topics focusing on quantitative system design and analysis.

Millard-Ball (2005) is one of the early quantitative studies of vehicle sharing systems, but the system only serves a small number of members. Du and Hall (1997) use models from fleet assignment problems and analyze operations of vehicle sharing systems. Simulation models (e.g., Barth and Todd (1999); Uesugi et al. (2007); Ciari et al. (2008)) are used to analyze the sensitivity of system costs and service quality to system parameters such as fleet size and vehicle relocation. George and Xia (2011) proposed a queuing model to determine the optimal fleet size of shared vehicles. Mathematical programming models were developed to determine optimal vehicle relocation, considering stationary demand (Chauvet et al., 1997) and dynamic demand (Fan et al., 2008; Kek et al., 2009). A couple of recent studies took a step further and addressed the location of vehicle sharing stations using discrete integer programming models (Kumar and Bierlaire, 2012) or continuous approximation (Li et al., 2016).

AVS has seldom been investigated until recent literature. Ford (2012) investigates the autonomous taxi service that has a fixed taxi service and allows AVs to operate between stands to pick up passengers. Vehicles are allowed to relocate to more favorable locations for potential next demand when needed. Kornhauser et al. (2013) investigates extending this idea to exploring dynamic ride sharing implications for all person-trips across New Jersey. In these models, one or more passengers boarded at fixed stations, where a taxi waits a given time before departing, and all passengers having similar destinations share a ride. Passengers are expected to relocate themselves to these stands and they do not solve the first-mile, last-mile problem, while one advantage of AVS or autonomous taxis is the capability of driving or relocating themselves to provide door-to-door service.

Some studies address related problems to the one investigated in this paper. Fagnant and Kockelman's (2014) and Chen et al. (2016) adopted agent-based simulation approaches for similar concepts. They consider each AV as an agent and defined rules for AV behavior such as relocation and EV charging, and the resultant system benefits of AV sharing are significant. Heuristic rules are used when relocating AVs and this may result in suboptimal system performance. Mahmoudi and Zhou (2016) study vehicle routing problem with pickup and delivery services with time windows using a dynamic programming approach based on state–space–time network representations. While exact optimality is guaranteed, the computational burden makes it still challenging to solve large-scale instances to the full optimality. Also considering vehicle reservation, Wang et al. (2014) studied a Taxi Dispatch problem with advance reservations. A trip-chaining strategy based on a customized algorithm of the pickup and delivery problem with time windows is proposed using certain heuristic dispatching rules and a Tabu search. This cannot guarantee the optimal solution and the tradeoff between computational burden and optimality is unknown.

None of these papers specifically addresses the unique opportunities of the AVSR concept, which combines three components – AVs, vehicle sharing and service reservation – to achieve more significant system benefits. Many proposed methods in the literature, such as integer programming and dynamic programming, are computationally intensive and are not suitable for large scale AVS/AVSR system design. Therefore, there is a need for an efficient model for optimizing a large number of AV trips over large areas. To bridge these gaps, this study proposes linear programming models that can solve AVSR problems using constructed AVSR space-time networks. The efficiency brought by linear programming models allows AVSR system operators to make optimal AV sharing decisions on large-scale AVSR problems with lowest operational costs and computational burden.

Note that the paper does not intend to propose a competing model to existing integer programming-based models for Pickup and Delivery Problems with Time Windows (PDPTW). We focus on large-scale, real-world problems yet with exact pickup times, suitable for reserved car sharing transportation service, while the literature (e.g., Savelsbergh and Sol, 1998; Ropke and Cordeau, 2009) focus on relatively small- or medium-sized instances (less than 100 request nodes and vehicles) with more complex operational settings, such as flexible time windows and ride sharing. Traditional PDPTW models are computationally intensive and have difficulty in handling problems of the sizes that this paper considers. Therefore with additional but reasonable attributes of the operational settings (e.g., allowing reservations), this paper's main contribution is the introduction of more computationally efficient linear programming models to solve these new problems.

3 Problem Statement

For the convenience of the readers, key symbols used in this study are listed in Table 1.

Table 1 Notation of parameters and variables

Parameters	
$i \in I$	Index of trip demand $i \in I$
$i^- \in I^-$	Pickup location node of trip demand $i \in I$
$i^+ \in I^+$	Delivery location/node of trip demand $i \in I$
I	Set of trip demands in a planning horizon $[0, T]$; $\|I\| = N$
I^-, I^+	Set of pickup and delivery nodes for all trip demands in I
T	AVSR planning horizon $[0, T]$
(i^-, i^+, t_i)	Trip demand i departing from origin i^- to destination i^+ at time t_i
A	Reachable demand set $A^r = \{(i, j) \mid i, j \in I, i\ can\ reach\ j\}$
l_i	Cost of losing demand i
d_{ij}	Driving cost from demand i to demand j, $\forall\, i, j \in A$
p_{ij}	Parking cost between delivery i and pickup j
d_i	Cost to initially dispatch a vehicle to i or collect a vehicle in the end from i
F	Total fleet size (total number of available AVs)
f	Fleet cost per vehicle
m_{ij}	Capacity of link (i, j)
c_{ij}	Cost of using link (i, j)
t_{i^-}, t_{i^+}	Start and end time of trip i
$v(a, b, t)$	Travel speed when a vehicle travel from a to b at time t, where a and b can be any pickup or delivery location for same or different demands
$TD(a, b)$	Actual travel distance from location a to b
θ	AV relocation buffer time
μ	AV relocation buffer distance
T'	Rolling horizon
t^u	Update interval for rolling horizon approach
Decision Variables	
x_{ij}	Linear variable: the number of vehicles using link (i, j); $x_{ij} \geq 0$
y_{ij}	Binary variable: $y_{ij} = 1$ if link (i, j) is used by a vehicle; $y_{ij} = 0$ otherwise

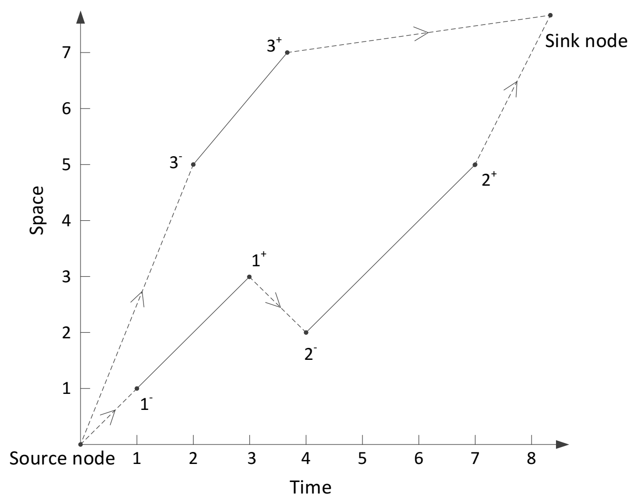


Figure 1 Illustration of trip demands and trip chaining on a space-time diagram

The problem is illustrated in Figure 1 where there are three trip demands: $1^- \to 1^+$, $2^- \to 2^+$ and $3^- \to 3^+$ along a linear corridor. Without loss of generality, our model assumes the existence of a general depot, where vehicles can be dispatched from and collected at it. This depot is represented by using a source node and destination node in Figure 1. These two nodes are virtual and can represent one real depot or multiple depots in large urban areas. The model in the paper can be easily extended to multi-depot models as will be discussed later. Each demand needs to be served before a requested departure time. The research problem is how many AVs are needed and how to dispatch them to serve the three space-time demands. In Figure 1, the three trips are represented on a space-time diagram where vehicles are assumed to travel one unit of distance per unit of time. If not well-coordinated, three AVs may be needed to meet all three demands. However, it is possible to assign one AV to serve demand 1 first and then travel back to serve demand 2 because the start time of demand 2 is greater than the end time of demand 1, resulting in a time difference that is sufficient for a vehicle to be relocated from 1^+ to 2^-. Such patterns in which multiple trips are served by a single AV, such as $1^- \to 1^+ \to 2^- \to 2^+$, are referred to as an *AV trip chain* in this paper.

As discussed above, when all three trip demands are known in advance through certain on-line request systems, an AVS problem becomes an AVSR problem. The reservation period can be as short as 5 minutes (Fagnant et al, 2014) to 30 minutes (e.g., Wang et al [2014]), or as long as a few hours to a day. When there is a large number of trip demands over a long period (e.g., daily peak periods or whole days), an efficient approach is needed to determine the trip chains for each AV, concerning which travelers are to be transported by this AV and the sequence to pick up and deliver each traveler at a required time. In the example in Figure 1, the trip chain includes information on two AVs that are going to be used to meet three demands. The *vehicle use rate (VUR)*, defined as the required number of private vehicles in the base case (which we assume to be identical to the number of trips) divided by that for AVSR, in this case is $\frac{3}{2} = 1.5$. The AVs are initially going to be dispatched to location 1^- and 3^-, and then one AV will serve demand 3 and the other will serve demand 1 and 2 sequentially. The trip chains need to be optimally designed in terms of certain objective functions, such as total AVSR system operation costs, such that the system operator can best manage the vehicle sharing systems.

An important objective of this paper's effort is to automatically determine the required minimum number of AVs, referred to as required *fleet size*, given demand requests (i^-, i^+, t_i), where $i \in I$. The literature used simulation approaches to decide the required fleet size. For example, Fagnant et al. (2014) runs the model 20 times with different trip demand patterns to determine how many vehicles are needed and where they should start (i.e., where to be initially dispatched). In our effort, we hope to simultaneously determine minimum required fleet size and trip chains given a certain demand pattern. AVs will only need to be initially dispatched to the pickup location of the first demand of each AV's trip chain.

Through the mechanism of reservation, AVSR can reduce travelers' waiting times, compared to traditional taxi or ride-hailing services which require travelers to wait for a certain period of time for the vehicles to travel from a different location to the traveler's location after the request. When AVSR is well designed and the fleet size is large enough, the trip chaining designed from the proposed model requires that vehicles arrive before a trip's pickup time given that vehicle on-road travel time is assumed to be known/deterministic. When travel time is stochastic due to traffic congestion, the system can be designed to allow some buffer time θ to ensure that the AVs wait for travelers at requested times and locations instead of the opposite. Spatially, how far away AVs should relocate themselves to pick up the next traveler is another interesting parameter that is referred to as buffer distance μ. While this parameter should not be bounded from the perspective of solution optimality, limiting this parameter can reduce the problem size and thus also computational burden, which will be discussed in the next section of model development.

In optimal planning problems such as AVSR, planning horizon T is another critical consideration. Ideally, travelers who use the service (e.g., 5% of the total travel demand) make requests one day ahead and the AVSR system can plan AV trips to meet all requested demands throughout a day. On one hand, this approach may cause high computational burdens; therefore a highly efficient model is needed. Planning for a shorter horizon and less demand can significantly reduce computational requirements but may lead to suboptimal solutions. Additionally, the quality of solutions from smaller planning horizon needs to be studied. One research question of this paper is to investigate the impact of the length of planning horizon on system performance. On the other hand, some travel requests may only be made a few hours before the actual trip and thus the AVSR system should be able to account for such "temporary" demands. A mixture of the length of planning horizon T for different travelers needs to be addressed by the model.

4 Methodology

4.1 Single-horizon Model

This section develops an optimization model that can efficiently solve the proposed AVSR problem with a defined horizon for all travelers. Consider a set of trip demands, indexed by i, distributed in the studied space. To prepare for model construction, an AVSR network (V, E) is constructed. There are two different types of nodes in the AVSR network. One type is trip nodes – pickup and delivery nodes for each trip demand. We denote pickup nodes with i^- and delivery node with i^+ for each trip demand $i \in I$. Another type is origin and destination nodes where AVs are dispatched and collected. For the network modeling purpose, we set a dummy source node o and a dummy sink node d. All AVs will be dispatched from source node o and finally collected to sink node d. The final node set is $N = \{i^-, i^+ \mid \forall i \in I\} \cup \{o, d\}$. In the meantime, an AVSR network has five different types of links with the final edge set denoted as $A = A_1 \cup A_2 \cup \{(o, i^-),\}_{i \in I} \cup \{(i^+, d)\}_{i \in I} \cup \{(o, d)\}$, where $A_1 = \{(i^-, i^+) \mid \forall i \in I\}$ and $A_2 = \{(i^+, j^-) \mid \forall (i, j) \in A^r\}$. The five types of links are as follows:

- Dispatch link (o, i^-): each AV is dispatched to its first trip demand i through this dispatch link before serving demand i;
- Service link (i^-, i^+): each demand i is served by this service link;
- Relocation link (i^+, j^-): after delivery of demand i at its delivery node i^+, an AV needs to be relocated to pickup node j^- for the next assigned trip demand j;
- Collection link (i^+, d): each AV is collected from its last trip demand i through this collection link after demand i is served;
- Virtual link (o, d): the link is added as a virtual link for modeling purposes for AVs to flow directly from o to d without any cost to the system. This link allows required fleet size to be an implicit variable for optimization, and thus the model can determine required vehicle fleet size while solving for trip chains.

Note that any relocation link (i^+, j^-) should follow the principle of *reachability*, which can be defined as that the required travel time[1] on the relocation link (i^+, j^-) should be less than the difference between the end time of trip i and the start time of trip j, $t_{j^-} - t_{i^+}$, as shown in Eq. 1.

$$t_{j^-} - t_{i^+} \geq \frac{TD(i^+, j^-)}{v(i^+, j^-, t_{i^+})} \tag{1}$$

After defining and identifying various types of links, an AVSR network is built as illustrated in Figure 2 using the three-demand example in Figure 1. Different types of links are illustrated with different types of lines. All links are uni-directional.

[1] If travel time stochasticity is considered, the corresponding travel time can be quantified with certain reliability measures (e.g., certain percentile travel time).

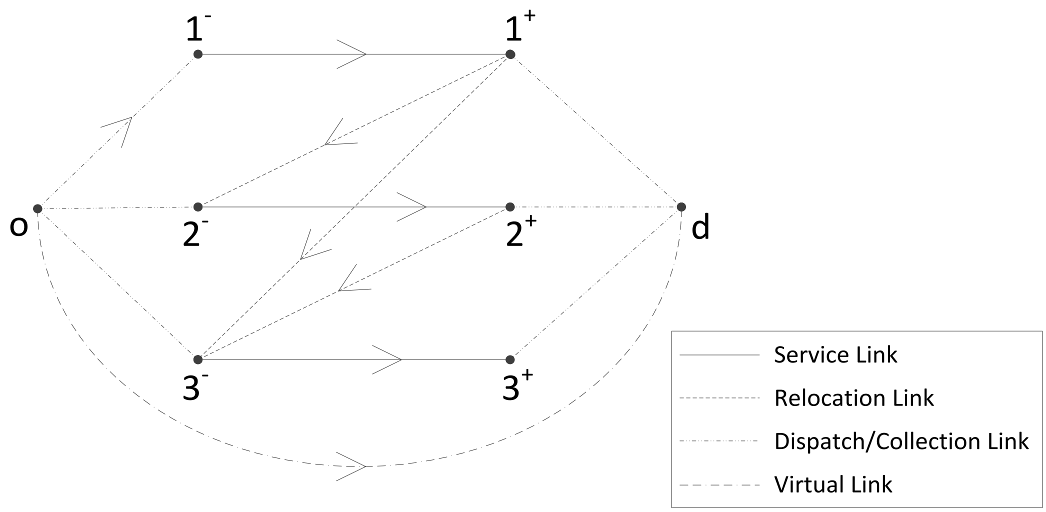


Figure 2 AVSR network for three-demand example

Each link is assigned with multiple link attributes including link distance, travel time, cost, and capacity. Link capacity is assigned according to Eq. 2, which indicates that all links but the virtual link have a capacity of 1, because a trip demand only needs to be served once. The capacity of the virtual link (o, d) is the maximum available number of AVs, i.e., fleet size F. Note that F is not the actual required fleet size output from the model. It reflects the total available AV resources that can be potentially used for an AVSR system.

$$m_{ij} = \begin{cases} 1, & if\ (i,j) \in A \backslash \{(o,d)\} \\ F, & if\ (i,j) = (o,d) \end{cases} \tag{2}$$

Determination of link costs is dependent on the actual objective function of the proposed optimization model. The objective can be minimization of the total VMT of all AVs. The objective can also be minimizing total cost for AVSR system operations, or total system costs including operator and traveler costs such as travel time and toll use. In this paper, we consider an objective function composed of costs of vehicle usage, fleet ownership, vehicle dispatch and collection, parking, and the penalty from unserved trip demands. Also, different types of links will be assigned with different link costs as shown in Eq. 3. It can be conveniently modified when other objectives are considered.

$$c_{ij} = \begin{cases} 0, & if\ i = o, j = d \\ d_j, & if\ i \neq o, j = d \\ f + d_i, & if\ i = o, j \neq d \\ -l_i, & if\ (i,j) \in A_1 \\ d_{ij} + p_{ij}, & if\ (i,j) \in A_2 \end{cases} \tag{3}$$

The virtual link is only for modeling purposes and incurs no costs. A cost d_i or d_j is considered for each dispatch or collection link, respectively. To extend this model to a multi-depot scenario, we can make d_i or d_j represent the travel cost between i^- or j^+ and its closest depot. An extra cost f is added to each dispatch link to consider AV maintenance, refueling, and others. While serving a demand on a service link generates no operation costs, losing a demand because of suboptimal planning of AV trip chains will result in revenue reduction and customer dissatisfaction. Therefore, a link cost of $-l_i$ is assigned to each service link where l_i is the penalty value for losing a demand i. This cost can also be interpreted as the negation of the revenue collected from serving this trip. Lastly, AV relocation may generate costs in travel (e.g., fuel use) and parking, denoted by d_{ij} and p_{ij} respectively.

The network construction process involves other key considerations, particularly during the establishment of relocation links. The following parameters help define key attributes of AVSR.

Buffer time θ

For a relocation link (i^+, j^-), the maximum time difference between the delivery time of demand i and the pickup time of demand j should be greater than actual travel time from i^+ to j^-, as shown in Eq. 4. However, considering that travel time may be unreliable due to congestion during AV relocation, a parameter, buffer time bound θ, is added to account for travel delay from a travel reliability perspective, as shown in Eq. 4. During network construction, only a relocation link (i^+, j^-) that meets the condition in Eq. 4 is allowed.

$$t_{j^-} - t_{i^+} \geq \theta + \frac{TD(i^+, j^-)}{v(i^+, j^-, t_{i^+})} \tag{4}$$

This parameter also implicitly incorporates delays due to other vehicle operations during relocation, e.g., refueling and routine check. Alternatively, *Buffer time θ* can be replaced by θ_i specific to each trip demand i, specified by travelers when making requests. Since this does not significantly affect the model structure, in the following modeling, we only use θ for simplicity. $v(i^+, j^-, t_{i^+})$ indicates the average travel speed from location i^+ to location j^- departing at time t_{i^+}. The value of this variable can be different throughout the day because of the time-varying traffic congestion effect. This value can be obtained from historical data (e.g., speed logs from roadway sensors) or future predictions (e.g., with traffic network flow models).

Buffer bounds for distance μ and time $\overline{\theta}$

Buffer distance bound μ determines an important attribute of AVSR – the AV relocation range. The distance between the delivery and pickup locations of two consecutive demands of an AV is bounded by this parameter. Ideally, this constraint should be relaxed to ensure optimality of the model result. However, using this parameter also significantly reduces the AVSR network size (number of relocation links) and thus enhances model efficiency. In many cases, particularly for relatively large networks with dense space-time demands, AVs will always pick up nearby demands instead of wasting resources during relocation to pick up distant demands.

Another parameter, buffer time bound $\overline{\theta}$, is less relevant to the AVSR strategy but can also help reduce network size. For a relocation link (i^+, j^-), the maximum time difference between the delivery time of demand i, i^+, and the pickup time of demand j, j^-, should not exceed the threshold $\overline{\theta}$. It is usually not a good plan to let an AV wait for many hours to pick up the next demand except for certain special demand patterns. Therefore, such links can be avoided when building the AVSR network without much impact on the optimality of the final result while largely reducing computational burden.

With the constructed AVSR network above, the following integer programming model is established for the AVSR system design problem.

$$\min_{\{y_{ij}\}} \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{5}$$

subject to

$$y_{ij} \le m_{ij}, \forall (i,j) \in A \tag{6}$$

$$\sum_j y_{ji} = \sum_j y_{ij}, \forall i \in I \backslash \{o, d\} \tag{7}$$

$$\sum_{j\in I\backslash\{o\}} y_{ij} = F, i = o \tag{8}$$

$$\sum_{j\in I\backslash\{d\}} y_{ji} = F, i = d \tag{9}$$

$$y_{ij} = 0 \ or \ 1, \forall (i,j) \in A \tag{10}$$

This is very similar to a standard minimum cost flow problem. Eq. (5) shows that the objective function is the total cost for AVSR system operations. Constraints (6) are standard capacity constraints. With all link capacity except virtual link bounded by 1, these constraints indicate these links can only be visited by no more than one AV. Constraints (7) – (9) are flow balance constraints for intermediate nodes $\{i^-, i^+ \mid i \in I\backslash(o, d)\}$, source node o and sink node d, repectively. Eq. (7) and (8) require that all available F AVs are dispatched either to real demands or to the sink node directly (equivalent to not using these AVs). Note that this limit in AV fleet size also indicates that when the optimal solution cannot meet all demand requests, the model will select the best AV trip chains. However, this will only occur when AV resources are limited. In our model, fleet size F can be set as a relatively large number to ensure that all demands can be met, unless the model decides that serving certain demands is too costly. Constraint (10) are binary constraints.

Remark: *Integer Program (5) – (10) is equivalent to Linear Program (11) – (16).*

A general integer program is usually hard to solve when the problem scale is large. Fortunately, the coefficient matrix of constraints (6) – (10) can be proven to be totally unimodular (Schrijver, 1998), and thus the Integer Program (5) – (10) is equivalent to the following Linear Program (11) – (16):

$$\min_{\{x_{ij}\}} \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{11}$$

subject to

$$x_{ij} \le m_{ij}, \forall (i,j) \in A \tag{12}$$

$$\sum_{j} x_{ji} = \sum_{j} x_{ij}, \forall i \in I \backslash \{o, d\} \tag{13}$$

$$\sum_{j \in I \backslash \{o\}} x_{ij} = F, i = o \tag{14}$$

$$\sum_{j \in I \backslash \{d\}} x_{ji} = F, i = d \tag{15}$$

$$x_{ij} \ge 0, \forall (i,j) \in A \tag{16}$$

Proof:

Equality constraints (7) – (9) can be expressed in matrix format $Ay = b$, where $A \in \boldsymbol{Z}^{(2N+2)\times(4N^2+2N+1)}$, $b \in \boldsymbol{Z}^{(4N^2+2N+1)\times 1}$ and N is the total number of demand nodes, i.e., the total number of demand requests. Therefore, the number of i^- will be N and the number of i^+ is also N.

According to Schrijver (1998), Linear Program (11) – (16) is equivalent to Integer Program (5) – (10) if matrix A is totally unimodular. Also, any matrix $A = \{a_{ij}\}$, where a_{ij} is matrix A's element at the i-th row and j-th column, is totally unimodular if it satisfies the following three conditions:

Condition 1: $a_{ij} \in \{0, 1, -1\}$ for all ij.

Condition 2: Every column of A has at most two non-zero entries.

Condition 3: The rows of A can be partitioned into two index sets I_1 and I_2 such that

3(a) If a column has two entries of different signs, then the indices of the rows corresponding to these non-zero entries must be in the same index set.

3(b) If a column has two entries of the same sign, then the indices of the rows corresponding to these non-zero entries must be in different index sets.

Matrix A can be expressed as below:

$$
\begin{array}{c|ccccccccc|ccc|cccc}
 & 1.1 & 1.2 & \ldots & 1.2N & 2.1 & \ldots & 3.1 & \ldots & 2N.2N & o.1 & \ldots & o.2N & 1.d & \ldots & 2N.d & o.d \\
\hline
1 & 0 & -1 & \ldots & -1 & 1 & 0 \ldots & 1 & \ldots & & 0 & & & 0 & & & \\
2 & & & & & \ddots & & & & & & \ddots & & & \ddots & & \\
\vdots & & & & & & \ddots & & & & & & \ddots & & & \ddots & \\
2N & & & & & & & & & & & & 0 & & & & 0 \\
\hline
o & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & 1 & \ldots & 1 & 0 & \ldots & 0 & 1 \\
d & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & 0 & \ldots & 0 & 1 & \ldots & 1 & 1
\end{array}
$$

It is obvious that Condition 1 is met. From Column o.1 to Column o.d, only one entry is nonzero, either at Row o or Row d. From Column 1.1 to Column 2N.2N, only two entries are nonzero, one row has the value of 1 and the other has the value of -1, and both of these non-zero entries are from Row 1 to Row 2N. If matrix A is partitioned into two index sets $I_1: 1 - 2N$ and $I_2: o - d$, it is obvious that Condition 3 is met based on the above analysis. Thus, it can be proven that matrix A is totally unimodular. Therefore, Linear Program (11) – (16) is a linear relaxed form of Integer Program (5) – (10), and all solutions of Linear Program (11) – (16) are integers.

Note that matrix A expressed in the above format is loose because not all columns exist. However, we expressed the loose format when proving unimodularity of matrix A because any compact matrix A obtained by removing certain columns will still be unimodular.

When constructing an AVSR network, in addition to *N* service links (i^-, i^+), the number of relocation links (i^+, j^-) is $N \times (N-1)$. Therefore, the possible number of columns of matrix A is $N + N \times (N-1) + 2N + 1 = N^2 + 2N + 1$. Additionally, because of spatial and time constraints imposed for the AVSR network construction, the actual column dimension is further reduced, but the attribute of unimodularity still holds.

The solution to models (11) – (16) is a complete set of strategies to dispatch AVs to meet demand requests in the planning horizon T, including required AV vehicle fleet size, initial dispatch location (pickup node of first trip demand for each AV), and AV trip chains (the vector of trip demands to be sequentially served by each AV). These results can be easily extracted from the binary values of decision variable x_{ij}.

4.2 Multi-horizon Model

The single-horizon model proposed in Section 4.1 optimizes AVSR system for one service horizon. A service horizon is defined as a period in which all demand requests have been made in the previous period and trip chains are generated before the start of this period. One example of a 24-hour service horizon is that all travelers make the next day's travel requests during the previous day before midnight and the AVSR system will generate dispatch decisions on the AV fleet size and trip chains for each AV before midnight. If the AVSR service provider only allows one service horizon of reservation, the model in Section 4.1 applies. The service provider may also allow travelers to make requests of different service horizons, though the service fee of longer service horizons is likely to be less than that of shorter horizons. Therefore, it is still necessary for the provider to be able to respond to a mixture of requests of different service horizons.

The concept of multi-horizon planning for AVSR is explained in Figure 3, where three different service horizons, 24 hours, 12 hours and 8 hours, are represented in three parallel timelines. Use an 8-hour service horizon as an example. All requests for the next 8-hour service horizon need to be made ε time ahead of the start of this 8-hour period. The time of ε is considered to account for computational time. Because new demand requests are made every 8 hours, it is necessary to update the solution every 8 hours. Note that in the above example, the service horizon is 8 hours. It is also referred to as update interval t^u in the model because the solution needs to be updated every 8 hours. Therefore, with different service horizons under consideration, the shortest service horizon should be used as update interval t^u.

Since some travelers may make requests of longer service horizons than others (e.g., 24 hours or 12 hours in Figure 3), it is necessary to select a rolling horizon T' to account for other service horizons. If $T' = t^u$, then the demands of longer service horizons are considered only when this demand falls under next update interval t^u. Ideally, the rolling horizon T' should be as large as possible to ensure solution optimality. Using a rolling horizon $T' > t^u$ can increase solution quality because parts of later demands are also accounted for and resultant trip chains for current update interval t^u can be closer to an optimal solution. Out of computational consideration, a reasonable rolling horizon T' should be selected. With that being said, in the example of Figure 3, with a service horizon or update interval t^u of 8 hours, a rolling horizon T' equal to or greater than 8 hours can be selected.

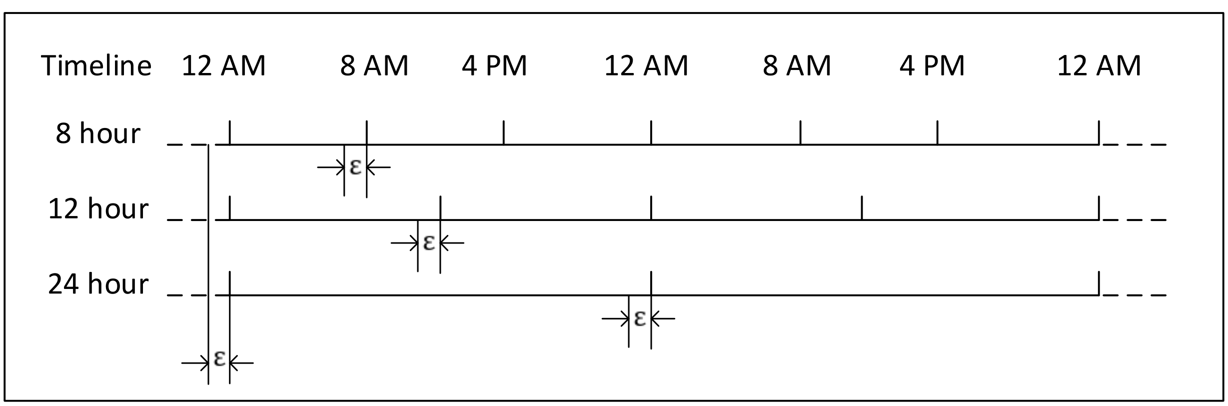


Figure 3 Illustration of multi-horizon planning

Note the model is proposed for reservation services in AVSR for relatively long service horizons t^u. The effectiveness for extremely short request horizon similar to ride-hailing, such as 15 minutes, as compared with other on-demand algorithms, needs to be investigated in future studies.

The multi-horizon model is proposed in this paper by adapting the single-horizon model in Section 4.1, with a rolling horizon T' and an update interval of t^u. In addition to the fact that new demand requests come in every service horizon t^u, unlike "one-shot" scenario in Section 4.1, some AVs may be available during the mid of next planning horizon after they complete current ongoing trip and their trip chains should be re-planned to account for new demands.

In this case, AVs that have been dispatched to serve earlier demands are also assumed to start from the virtual source node o, and we create a new type of dispatch link (o, i^+) for each uncompleted trip during last service horizon that is expected to end during the next rolling horizon T'. These new dispatch links are assigned with negative link costs of $-M$, where M is a large positive number, to ensure that these links will be selected by the linear program. For these completed demands, no service links are constructed and relocation links are constructed from node i^+. With such an AVSR network reconstruction, as shown in Figure 4, the efficient linear programming model (9) – (14) remains applicable. The AVSR network is constructed and the model is run for each update interval t^u.

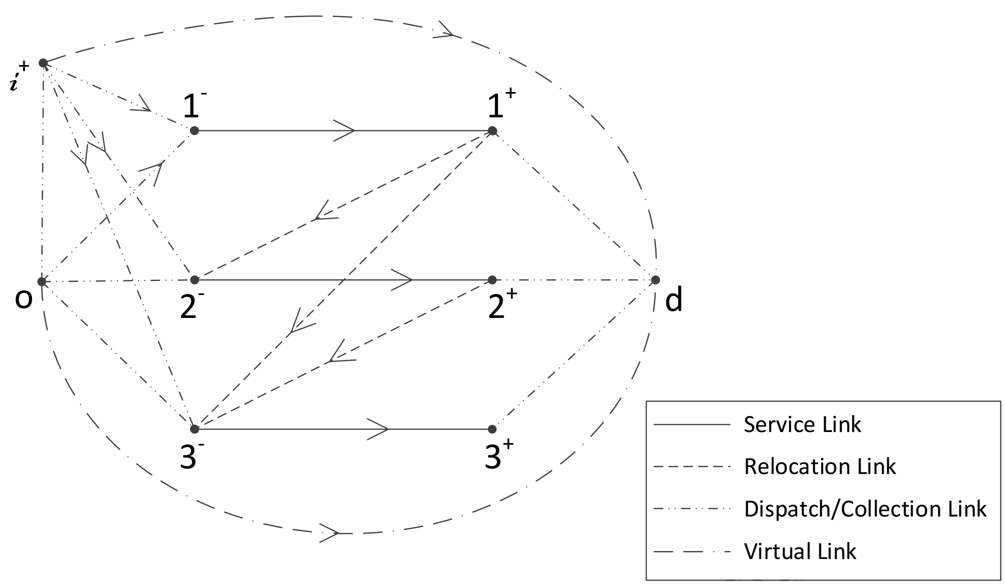


Figure 4 Expanded AVSR network for three-demand example

4.3 Modeling Considerations

The proposed model is to design strategies to dispatch AVs to meet demand requests, including required AV fleet size, initial dispatch location (pickup node of the first trip demand for each AV), and AV trip chains (the vector of trip demands to be sequentially served by each AV). Ride sharing is not considered and one AV will serve one demand request at a time.

This model can also be considered as an adapted special case of PDPTW, but can solve problems of significantly larger scales. The three-index formulation of Cordeau (2006) for the PDPTW in the origin–destination network is presented in Appendix A. Table 2 shows an analogy between Cordeau's model and our model for considerations of similar system attributes. Note that this is a comparison of two models that solve different problems, and this comparison is provided to clarify how the proposed model considers some key factors that are addressed in PDPTW problems.

Table 2 Analogy between Cordeau's model and our model for considerations of similar system attributes

Cordeau's model considerations	AVSR model considerations
(A.1) minimizes the total routing cost.	(11) minimizes the total system operational costs.
(A.2) guarantees that each passenger is definitely picked up.	In (11), c_{ij} uses negative costs $-l_{ij}$ for service links (i^-, i^+) to meet demands; if l_{ij} is a sufficiently large number, all demands will be met (unless vehicle fleet size is too small); by picking a reasonable value of l_{ij}, the model will meet demands to minimize system costs, and this is valuable when the vehicle fleet size is not large enough.
(A.2) and (A.3) ensure that each passenger's origin and destination are visited exactly once by the same vehicle.	(12) serves the same purpose; but the capacity of virtual link (o, d) helps determine the required fleet size.
(A.4) and (A.5) express that each vehicle k starts the trip from the source depot and ends the trip at the sink depot. (A.6) ensures flow balance on each node.	(13) – (15) also ensure flow balance at source, sink and intermediate nodes. However, the number of actual constraints for a constructed AVSR network can be significantly reduced because of pre-imposed spatial and temporal constraints, such as buffer time θ and buffer distance μ. Through the use of an AVSR virtual link (o, d), the minimization of (11) can automatically calculate optimal fleet size.
(A.7) and (A.8) ensure the validity of the time and load variables, and constant M is defined as a sufficiently large number.	Time constraints are already fulfilled during construction of the AVSR network; Load constraints do not apply.
(A.9) ensures that pickup nodes are visited before delivery nodes.	Not applicable because no ride sharing is considered.
(A.10) imposes constraints on time windows.	The time buffer θ requires the vehicle needs to arrive at the pickup location ahead of the ideal pickup time, and the pickup time window can be considered as $[t_i^- - \theta, \theta]$. But this is a simplified consideration and θ also has other

	purposes, such as serving as a buffer time for travel delay.
(A.11) imposes constraints on vehicle capacity.	This types of capacity constraints are not needed because no ride sharing is considered. Vehicle capacity is considered when constructing the AVSR network, by dividing large passenger groups into subgroups.
(A.12) imposes the integrality of variables.	(16) indicates linearity of the model and it ensures efficiency of the proposed model to solve real-world, large-scale problems.

Also, Section 4.2 extends the model to efficiently solve multi-horizon cases, which makes the model applicable to real-world scenarios and implementable in real time. This model focuses on large-scale, real-world problems yet with exact pickup times, suitable for reserved car sharing transportation service. Traditional PDPTW models are computationally intensive and hard to handle problems of the sizes that this paper considers. Therefore with additional but reasonable attributes of the operational settings (e.g., allowing reservations), this paper's main contribution is introduction of more computationally efficient linear programming models to solve these new problems.

5 Case Study

5.1 Standard Grid Dataset

The model is firstly tested on a standard grid dataset that is widely used in the PDPTW literature. The purpose is to understand model efficiency and solution optimality under different demand patterns. Test instances similar to those in Savelsbergh and Sol (1998) are generated with the following approach.

The planning horizon is 600 time units. Multiple demand levels (i.e., number of requests) N are tested: 50, 100 and 500. We first randomly select N pickup times, for each of which, the coordinates of the pickup and delivery locations are randomly chosen according to a uniform distribution over a $m \times m$ square, where m is grid network dimension and $m = 50, 200$ are tested. The distance between every two adjacent points, horizontally or vertically, is 1 distance unit. It costs 1 time unit to travel 1 distance unit. We use $\theta = 0, \mu = 300$ and very small cost rates for parking p_{ij}, driving d_{ij} and daily dispatch/collection d_i. We use very large vehicle fleet cost f and cost of losing a demand l_i, such that the minimum number of AVs will be used. The purpose of this case study is to understand model efficiency and the potential of AV sharing, as measured by Vehicle Use Rate (VUR), defined in Section 3 as the required number of private vehicles/traditional taxis in the base case (which we assume to be identical to the number of trips) divided by that for AVSR.

Since the AVSR formulation generates large-scale linear programming models, the IBM CPLEX solver is used directly and it was proven effective in the case study. Once the model is built up and loaded to the computer memory, it takes only a few seconds to solve the linear programs on a PC with 16 GB memory and a dual-core central processor unit running at 2.70 GHz. One advantage of formulating this linear programming formulation is that even large-scale instances of this problem now can be directly fed into and efficiently solved with available commercial solvers.

The average solution times for instances with N= 50, 100, 500 and 1000 are 3, 16, 41 and 92 milliseconds, respectively, indicating great efficiency of the AVSR model. Figure 5 shows VUR distribution under instances with different demand patterns. These results are obtained from randomly generated 100 instances for each scenario, which is defined by three elements: demand level N (50, 500) – network dimension m (50, 200) – planning horizon T (600, 1200). It can be seen that VUR increases with larger demand levels and longer planning horizons, because more opportunities exist to form trip chains. Given the same planning horizon, a smaller network means shorter travel times and thus increases the possibility for any AV to serve more demands.

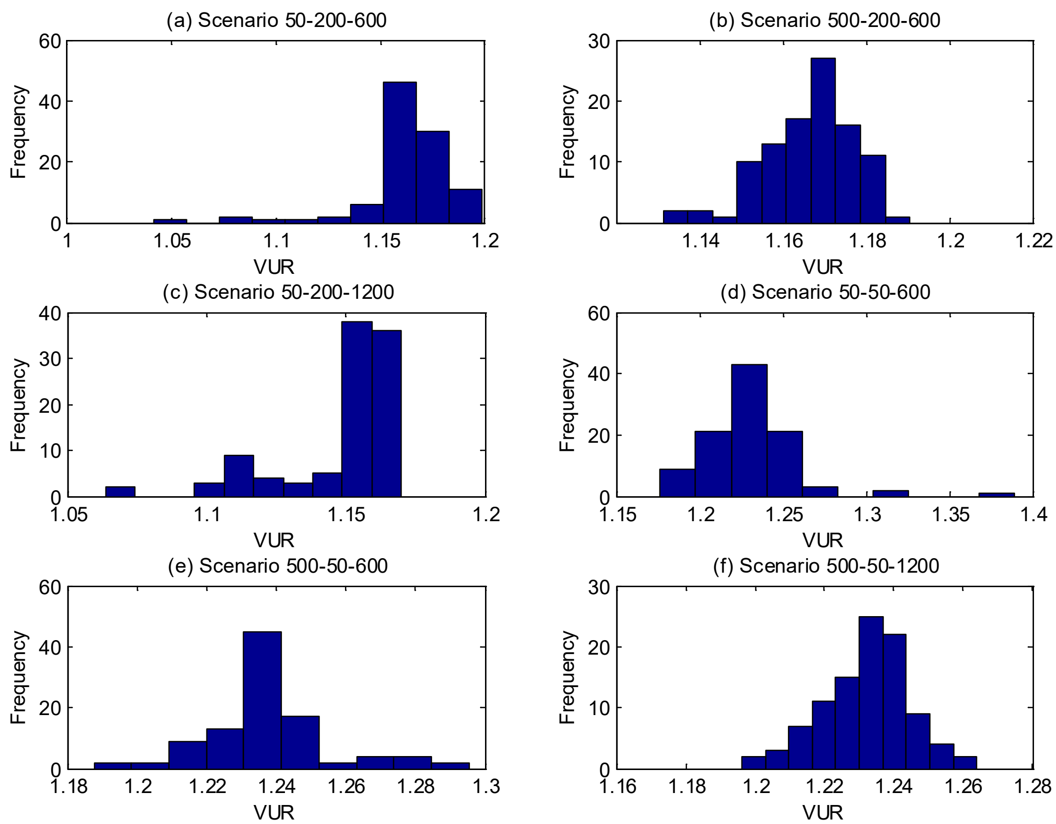


Figure 5 Vehicle use rate (VUR) distribution of 1000 instances with different demand patterns; scenarios are defined by three elements: demand level N (50, 500) – network dimension m (50, 200) – planning horizon T (600, 1200).

Note that in this case study, we only investigate the effects of above three parameters on AV sharing and model performance. As considered in the proposed model, many other factors will also have effects in practice, and sensitivity analysis of practical variables, such as time buffer θ, distance buffer μ, fleet cost f and parking cost p_{ij}, is conducted in the real-world case study in Section 5.2.

5.2 New York Taxi Dataset

The second case study of this paper uses taxi dataset from New York City. This dataset was obtained through a Freedom of Information Law (FOIL) request from the New York City Taxi & Limousine Commission (NYCT&L) (http://www.nyc.gov/html/tlc/html/home/home.shtml). It covers four years of taxi operations in New York City and includes 697,622,444 trips. Each row of the data represents a single taxi trip. Each trip includes information on the vehicle permit, vehicle license, vendor ID, rate code, pickup and delivery time, passenger count, trip time, trip distance, and the latitude and longitude coordinates for pickup and delivery locations. This dataset does not contain trips of unoccupied taxis between serving demands. The dataset also contains a large number of errors as documented in other work (Donovan and Wok, 2015). For example, there are several trips where the reported meter distances are significantly shorter than the straight-line distance, violating Euclidean geometry. Some trips have the same pickup and delivery locations. Additionally, many trips report GPS coordinates of (0, 0), or contain impossible distances, times, or velocities. All of these types of obvious errors account for roughly 10% of all trips and are discarded. Figure 6 shows an example of pickup (red) and delivery (green) locations on an example day of 1/17/2013. The data of this day is also picked for the case studies in the paper because it is a typical workday during the week and there are generally less errors in this dataset. The data will be randomly resampled to generate different traffic demand levels and patterns, and the details will be discussed in this section.

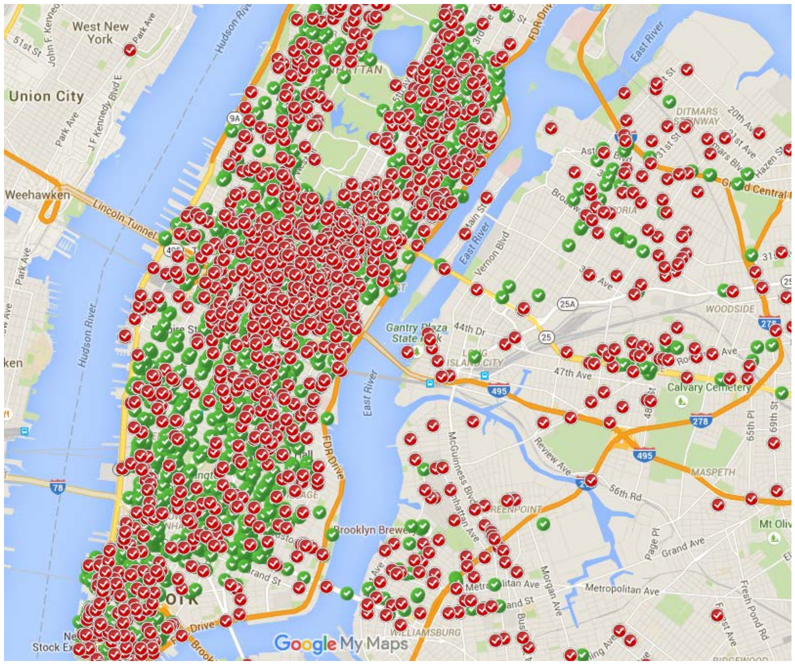


Figure 6 Example taxi data on 1/17/2013

Since the dataset does not provide congestion information, for simplicity of model implementation in the paper, we use a distance expansion factor $\eta(i^+, j^-)$ and congestion expansion factor $\delta(i^+, j^-)$ to calculate actual travel distances and travel times for all relocation links (i^+, j^-). Eqs. (17) and (18) use the average of the expansion factors of trips i and j to approximate the expansion factor of relocation link (i^+, j^-). Future studies can use available traffic data and route selection tools to calculate travel time and distance.

$$\eta(i^+, j^-) = 0.5\left(\frac{TD(i)}{SD(i^-, i^+)} + \frac{TD(j)}{SD(j^-, j^+)}\right) \tag{17}$$

$$\delta(i^+, j^-) = 0.5\left(\frac{TT(i)}{TD(i)} + \frac{TT(j)}{TD(j)}\right) \tag{18}$$

where $TD(i)$ is the travel distance obtained directly from the data, $SD(i^-, i^+)$ is the Euclidian distance calculated based on the latitude and longitude of i^- and i^+, and $TT(i)$ is the travel time obtained directly from the data.

The actual travel distance of relocation link (i^+, j^-) is estimated by $TD(i^+, j^-) = SD(i^+, j^-) \cdot \eta(i^+, j^-)$, and the actual travel time by $TT(i^+, j^-) = TD(i^+, j^-) \cdot \delta(i^+, j^-)$. These results will need to be compared with buffer time bounds $[\theta, \overline{\theta}]$ and buffer distance μ to decide the feasibility of relocation links. Also, the cost of relocation link (i^+, j^-) is $d_{i^+j^-} = \varepsilon_1 \cdot TT(i^+, j^-) + \varepsilon_2 \cdot TP(i^+, j^-)$, where ε_1 is the relocation cost factor that considers fuel use and vehicle mileage, ε_2 is the parking cost, and $TP(i^+, j^-)$ is the time of parking during relocation.

The models proposed in this paper aim to determine the optimal AV fleet size and AV service trip chains that minimize the objective function – the AVSR system operational cost. In addition, two other performance measures, VMT and VUR, are used to evaluate system effectiveness. Some literature (e.g., Fagnant, 2014) reported an increase in VMT of about 10% with autonomous vehicle sharing because of additional miles traveled during relocation. It is necessary to see the changes in VMT and VUR under different parameter settings of the proposed model.

In the case studies, the cost of losing demand i is assumed to be proportionate to the trip distance, $l_i = \pi \cdot TD(i)$, where π is a cost factor that considers trip fare and the penalty of not being able to serve a demand. If π is very large, the model will meet all demands as long as enough AVs are available. For the default setting, we use $\pi = \$100$ per mile as a large value so all demands are required to be met given sufficient AV fleet size. Driving cost from demand i to demand j, or relocation link cost factor, is $\varepsilon_1 = \$30$ per hour and $\varepsilon_2 = \$5$ per hour. The cost to dispatch or collect a vehicle from node i is d_i = \$30 per vehicle. Fleet cost is assumed to be $f = \$30$ per vehicle. The total fleet size is assumed to be relatively large: $F = 10{,}000$. Parking rate is \$5 per hour. In terms of parameters related to AVSR strategies, the buffer time is defaulted to be $\theta = 0$ min. The buffer distance is defaulted to be $\mu = 20$ miles, a large distance for relocation.

Three key performance measures are used to indicate model effectiveness. *VUR* indicates the potential to decrease vehicle ownership. *VMT* indicates the total vehicle miles traveled by all vehicles to serve the demands. Since the dataset does not provide data other than each trip from pickup to delivery locations, we assume that, for other services, each vehicle needs to travel additional three miles when vehicles are empty, including miles traveled to be dispatched to pickup locations and/or collected from delivery locations and to wait for other passengers. With that being said, the total VMT for base case can be calculated as $VMT_{base} = \sum_{i \in I}[TD_i(i^-, i^+) + 3]$, where $TD_i(i^-, i^+)$ is the distance between pickup location i^- and delivery location i^+ for each demand i. With this assumption, in the case study, we compare the AVSR system with regular taxi or ride hailing services for which passengers request a ride when needed through phone calls or smartphone apps. Usually, nearby vehicles need to travel a few miles to the pickup location and they may relocate to places where there are more potential trip demands. Therefore this assumption is relatively conservative. *VMT Ratio* is defined as the average VMT of model results divided by the base case, in which each demand is served without any coordination.

This paper accounts for random trip patterns from two different perspectives. First, we assume two levels of demands in the numerical study: 1% and 2% of the total demands. Second, since only a small percentage of travelers are assumed to use the ASVR service, these subsets of demands are randomly selected from the total demands. With each row in the dataset representing a single demand, 10 sample data subsets are generated for each demand level and the average values of VUR, VMT and VMT Ratio are reported.

The IBM CPLEX solver is used and this case study is tested using the same computer as in Section 5.1. Table 5 shows some basic statistics of model performance. Considering the problem planning horizon, which is significantly longer than the solution time, the efficiency of the model satisfies requirements for practical implementation. Also, it can be seen that a constrained choice of value of time buffer θ and distance buffer μ can significantly decrease the problem size and the corresponding solution time. Later in our case study, we also find that within a certain range of values of θ and μ, the solution quality does not vary significantly. This is another feature that future enhances the efficiency of the proposed model.

Table 5 Basic statistics of model performance

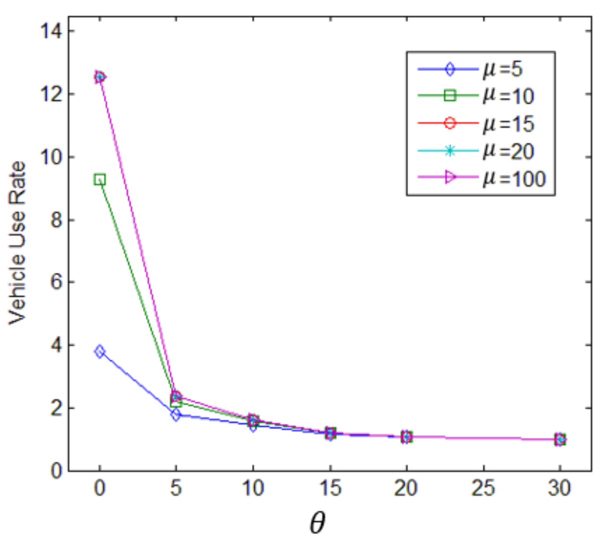
Problem Size		AVSR Network Link #	Memory (GB)	Performance (Solution Time, sec)
Demand Level	# of Trips			
$\theta = 0, \mu = 100$				
1%	1168	546279	1.21	3.86
2%	2336	2184026	2.65	21.07
$\theta = 15, \mu = 20$				
1%	1168	56651	0.18	0.15
2%	2336	223740	0.41	0.61

Single-horizon model

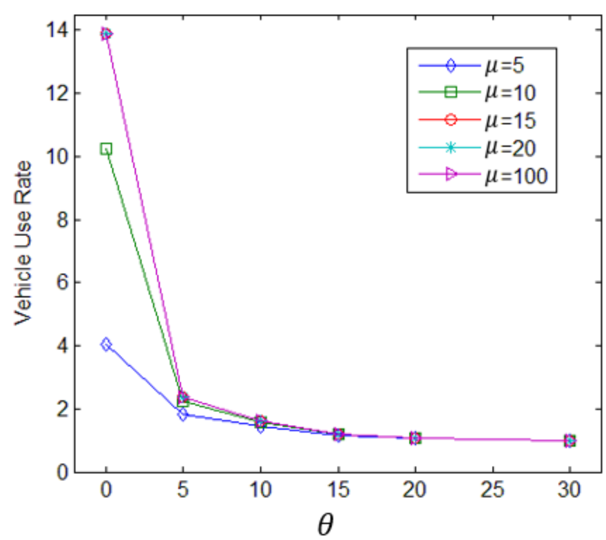
The first set of analysis is to examine the performance of the single-horizon model proposed in Section 4.1 with different buffer time θ and buffer distance μ, as shown in Table 4. It is expected that a larger buffer distance μ and a smaller buffer time θ result in a larger VUR because the model has more options of demands for trip chaining. Figure 7 shows this trend with more data points. It is interesting to notice that the VUR decreases dramatically from $\theta = 0$ to $\theta = 5$ and decreases slowly afterwards. This is partially because if the model allows five minutes between any two demands, this already significantly reduces the number of possible trip demands under consideration over the whole day and therefore, the decrease is less dramatic when $\theta > 5$. When $\theta > 20$, VUR is close to 1 partially because possible trip demands under consideration for chaining are greatly limited and there can be a high cost of parking of vehicle while waiting for next demands. Interestingly, while VUR increases with varying θ and μ, VMT does not increase significantly. It is demonstrated by the results that most VMT ratios are close to 1, indicating comparable VMTs of the model results and the base case. In some cases, taking $\theta = 0, \mu = 100$ under 1% demand as an example, the VUR increases by $\frac{12.56-1}{1} = 1156\%$ (one AV replacing 12.56 regular private vehicles) while VMT only increases by 15% – a much smaller percentage change. Figure 6 also shows that VUR can vary under different demand levels. It is intuitive that more trip demands mean more opportunities of trip chaining. Lastly, it is noted that VUR results are only sensitive to μ when $\theta < 5$ and sensitive to θ when $\theta < 5$ independent of μ.

Table 4 Sensitivity analysis with varying θ and μ under different demand levels

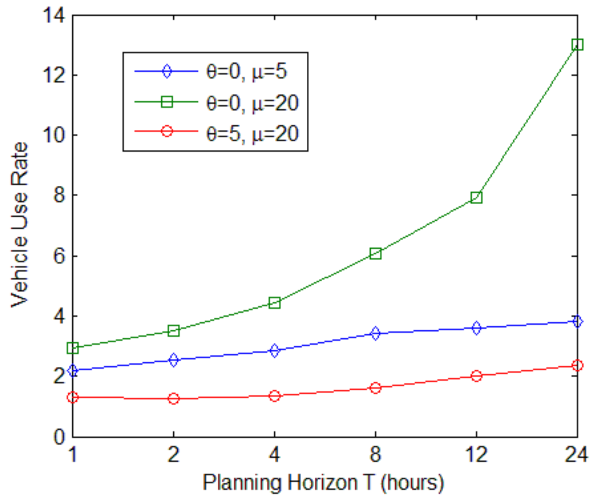
Scenario	VUR			VMT (miles)			VMT Ratio		
1% of Daily Demand									
θ	μ=5	μ=15	μ=100	μ=5	μ=15	μ=100	μ=5	μ=15	μ=100
0	3.80	12.56	12.56	11013	12983	12998	0.97	1.14	1.15
5	1.79	2.35	2.36	11095	12493	12525	0.98	1.10	1.10
10	1.44	1.61	1.61	11199	11973	11973	0.99	1.06	1.06
20	1.08	1.08	1.08	11326	11396	11396	1.00	1.00	1.00
30	1.00	1.00	1.00	11343	11343	11343	1.00	1.00	1.00
2% of Daily Demand									
θ	μ=5	μ=15	μ=100	μ=5	μ=15	μ=100	μ=5	μ=15	μ=100
0	4.03	13.90	13.90	21974	25706	25722	0.97	1.14	1.14
5	1.81	2.39	2.39	22147	24912	24943	0.98	1.10	1.10
10	1.44	1.60	1.61	22335	23778.34	23809.73	0.99	1.05	1.05
20	1.08	1.09	1.09	22597	22735	22735	1.00	1.00	1.00
30	1.00	1.00	1.00	22631	22631	22631	1.00	1.00	1.00



(a) 1% demand



(b) 2% demand



(c) Planning horizon

Figure 7 Results of vehicle use rate with varying θ and μ under different demand levels and planning horizons

Figure 7(c) presents the results of VUR with a different planning horizon T. We tested six different planning horizons for a single day of the AVSR service: 1 hour, 2 hours, 4 hours, 8 hours, 12 hours and 24 hours. Results show that under more "constrained" AVSR strategies – $\theta = 0, \mu = 5$ and $\theta = 5, \mu = 20$, VUR results are not very sensitive to increases in planning horizons. When under favorable strategies, i.e.,$\theta = 0, \mu = 20$, VUR results are sensitive to different planning horizons and a dramatic increase in VUR occurs when $T > 8$ hours. This indicates that ideally a planning horizon of more than 8 hours and a buffer time and distance of $\theta = 0, \mu = 20$ are preferred for operators in this area. When travel time reliability and predictability significantly improved in the future, service quality can still be maintained even when $\theta = 0$.

Sensitivity analyses were also conducted to investigate the effects of fleet cost f, parking cost p_{ij} and dispatch/collection cost d_i on system performance under favorable AVSR strategies. i.e., $\theta = 0, \mu = 100$ and demand level of 1%. Two different values of unit cost of losing demand π (\$30 and \$100 per mile) were also tested. However, system results are not sensitive when $\pi =$ \$100, which is used in the default AVSR setting. It is expected because this value was set very high in the default setting such that all demands are met, and thus all results under different scenarios are the same. Table 5 shows the results when $\pi = \$30$. As fleet cost f and dispatch/collection cost d_i increases, the model selectively ignored some demands to minimize system costs. Although VUR also increases, it is at the cost of serving less customers, and this can significantly reduce the service quality. Unfortunately, VMT also increases even when serving fewer demands. Interestingly, the variation of parking cost p_{ij} does not impact system performance. It is because the trip chaining design from the proposed model results in very limited time of parking for each AV and thus parking cost is only a small part of total system cost. This further proves the effectiveness of the proposed model in designing an AVSR system.

Table 5 Results of Sensitivity Analysis of Cost Parameters

Scenario	VUR	VMT (miles)	Demand Met	Fleet Size
f ($/veh)				
10	12.56	12998	1168	93
30	12.56	12998	1168	93
50	12.68	12996	1167	92
70	13.36	13051	1162	87
90	13.95	13098	1158	83
d_i ($/veh)				
10	12.56	12998	1168	93
30	12.56	12998	1168	93
50	13.36	13051	1162	87
70	14.11	13108	1157	82
90	14.61	13130	1154	79
p_{ij} ($/mile)				
5	12.56	12998	1168	93
20	12.56	12998	1168	93
50	12.56	12998	1168	93

Multi-horizon model

The total planning horizon is $T = 4$ hours. $t^u = 1, 2, 4$ hours indicate the models are run at 1-hour, 2-hour and 4-hour service intervals, respectively. It is also assumed that 10% of the demand requests are made before the start of first service interval t^u, and these demands spread across all four hours. For scenarios with different t^u, before each t^u, all demands during this t^u become known to the operator.

Table 6 shows results from the multi-horizon model proposed for a selected peak period. Results of scenarios of $\theta = 0$ are reported since it generates ideal system performance. The results are similar to that presented in Figure 7(c). The VUR increases as both t^u and T' increases, indicating benefits of having travelers with long service horizons. Again, the multi-horizon model also generates favorable results in VMT Ratio, implying no significant increase in VMT during relocation.

Table 6 Results of multi-horizon model

Scenario	Vehicle Use Rate	VMT Ratio
10% of Peak Period* Demand		

$t^u - T'$	μ=5	μ=15	μ=5	μ=15
1 – 1	1.93	2.08	0.97	1.00
1 – 2	1.99	2.15	0.97	1.00
1 – 4	2.09	2.27	0.97	1.00
2 – 2	2.20	2.40	0.97	1.01
2 – 4	2.22	2.43	0.97	1.01
4 – 4	2.30	2.59	0.97	1.01
20% of Peak Period Demand				
$t^u - T'$	μ=5	μ=15	μ=5	μ=15
1 – 1	1.96	2.11	0.96	0.99
1 – 2	2.00	2.15	0.96	0.99
1 – 4	2.11	2.26	0.96	0.99
2 – 2	2.21	2.42	0.96	1.00
2 – 4	2.23	2.43	0.97	1.00
4 – 4	2.35	2.57	0.97	1.00

* Peak period indicates 6 AM – 10 AM.

6 Conclusions and Future Research

The emerging technologies of autonomous vehicles (AV) have the potential to further improve the efficiency of traditional sharing systems. Through connectivity using certain devices, travelers can request vehicles that are relatively far away before traveling, and AVs can be fully self-driving to relocate themselves automatically to any traveler location upon request without human drivers. In this paper, we examined a new vehicle sharing system, referred to as autonomous vehicle sharing and reservation (AVSR). In such a system, travelers can request AV trips ahead of time and the AVSR system operator will optimally arrange the AV pickup and delivery schedule and trip chaining patterns on the basis of the recorded trip demand requests. A linear programming model is proposed to efficiently solve for optimal solutions for AV trip chains under a single service horizon. It is noted that the optimal solutions to AVSR systems provide an upper bound benefits of similar AVS systems. A multi-horizon model, adapted from the single-horizon model, is also proposed to address scenarios where travelers to make requests of different service horizons, and the service operator needs to respond to a mixture of requests of different lengths of service horizons.

Case study results show that AVSR can significantly increase vehicle use rate (VUR), for example, by replacing more than 13 private vehicles or traditional taxis. In the meantime, it is found that the actual VMT incurred by AVSR systems, though with much more vehicle relocation, is similar to that of private vehicles or taxis. This indicates that AVSR can reduce vehicle ownership significantly and that the increased VMT incurred by relocation of sharing vehicles can be compensated by the reduction in required AV fleet size and optimized AV trip chains. This implies huge potential benefits for improving mobility and sustainability of our current transportation systems.

The effectiveness of AVSR may be different in different cities because of distinct trip patterns, some of which may allow the existence of a large number of trip chains and some may not. The linear programming models proposed in this paper, however, can be applied to any trip pattern and can identify such trip patterns using an AVSR network.

There are areas where this paper can be improved in future studies. First, ongoing research is investigating how the reduction of system operation cost can be used to reduce traveler cost as an incentive to increase market penetration and further reduce system costs. Second, the proposed model in this paper does not explicitly consider other vehicle operations such as refueling. Considering such a process would make the model more robust. This also applies to the idea of using real traffic data (even predicted traffic conditions) as the basis for constructing AVSR networks. We also only applied the model to one site as a case study to demonstrate the model efficiency. While the model can be applied to any network and demand pattern, it is an interesting question to research the efficiency of AVSR systems under different network and demand scenarios.

Another key future work is extending the proposed models to account for ride sharing. If this service is offered as an alternative, it is usually less expensive for travelers. The major difference is construction of a AVSR network that allows ride sharing. If the network is appropriately constructed, it may still be possible to use highly efficient linear programs to optimize over the extended AVSR network. How to construct such networks that also consider complexities and different characteristics of ride sharing is currently being investigated.

Acknowledgments

This research is partially supported by NSF Research Grant CMMI #1638355. Advice from Dr. Jianxun Cui at the Harbin Institute of Technology is much appreciated.

Appendix A

In this appendix, a standard three-index formulation of PDPTW, similar to Cordeau (2006), is introduced. The purpose is to provide a foundation for comparison with the proposed AVSR model. We use slightly different notations from the proposed model to comply with standard practice in PDPTW literature.

Let N be the total number of customer requests. The PDPTW can be represented by a complete graph $G(V, A)$ defined by node set V and arc set A, where $V = \{0, 1, 2, \dots, 2N + 1\}$. Nodes 0 and 2N + 1 represent the source and sink nodes, respectively. Each node $i \in V$ has a demand q_i and a non-negative service time d_i where $q_0 = q_{2N+1} = 0$ and $d_0 = d_{2N+1} = 0$. The subsets $P = \{1, \dots, N\}$ and $D = \{N + 1, \dots, 2N + 1\}$ are pickup and delivery node sets, respectively. Each pickup node i is associated with a delivery node $N + i$, and e_i and l_i represent the earliest and latest time at which service is allowed to start at node i, respectively. Furthermore, a routing cost c_{ij} and a travel time t_{ij} are associated with each arc $(i, j) \in V$. The travel time t_{ij} includes service time d_i at node i and the triangle inequality holds for routing costs and travel times. Also, consider a homogeneous fleet of vehicles $K = \{1, 2, \dots, m\}$ is the set of identical vehicles with capacity C.

A binary variable x_{ijk} is defined for each arc $(i, j) \in V$ and each vehicle $k \in K$, such that, $x_{ijk} = 1$ if and only if vehicle k visits node i and then travels directly to node j. Non-negative continuous variables Q_{ik} and B_{ik} respectively indicate the load of vehicle k after visiting node i and the time that the vehicle k starts servicing node i, for each $i \in V$ and each $k \in K$. The three-index formulation is expressed as a mixed-integer programming: Eqs. A1 – A12.

$$\min_{\{x_{ijk}\}} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} \tag{A.1}$$

subject to

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} \quad \forall i \in P \tag{A.2}$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{N+1,j,k} \quad \forall i \in P;\ k \in K \tag{A.3}$$

$$\sum_{j \in V} x_{0jk} = 1 \quad \forall k \in K \tag{A.4}$$

$$\sum_{i \in V} x_{i,2N+1,k} = 1 \quad \forall k \in K \tag{A.5}$$

$$\sum_{j \in V} x_{jik} = \sum_{j \in V} x_{ijk} \quad \forall i \in P \cup D;\ k \in K \tag{A.6}$$

$$B_{jk} \geq B_{ik} + t_{ij} - M\left(1 - x_{ijk}\right) \quad \forall i \in V;\ j \in V; k \in K \tag{A.7}$$

$$Q_{jk} \geq Q_{ik} + q_j - M\left(1 - x_{ijk}\right) \quad \forall i \in V;\ j \in V; k \in K \tag{A.8}$$

$$B_{ik} + t_{i,n+i} \leq B_{N+i,k} \quad \forall i \in P;\ k \in K \tag{A.9}$$

$$e_i \leq B_{ik} \leq l_i \quad \forall i \in V;\ k \in K \tag{A.10}$$

$$\max\{0, q_i\} \leq Q_{ik} \leq \min\{C, C + q_i\} \quad \forall i \in V;\ k \in K \tag{A.11}$$

$$x_{ijk} \in \{0, 1\} \tag{A.12}$$

The objective function (A.1) minimizes the total routing cost. (A.2) guarantees that each passenger is definitely picked up. (A.2) and (A.3) ensure that each passenger's origin and destination are visited exactly once by the same vehicle. (A.4) and (A.5) expresses that each vehicle k starts the trip from the source depot and ends the trip at the sink depot. (A.6) ensures the flow balance on each node. (A.7) and (A.8) ensure the validity of the time and load variables, and the constant M is defined as a sufficiently large number. (A.9) ensures that pickup nodes are visited before delivery nodes. (A.10) and (A.11) impose constraints on time windows and vehicle capacity, respectively. (A.12) imposes the integrality of variables.

References

Berman, O., Bertsimas, D., Larson, R.C., 1995. Locating discretionary service facilities, ii: maximizing market size, minimizing inconvenience. Operations Research 43 (4), 623–632.

Brian Donovan and Daniel B. Work. "Using coarse GPS data to quantify city-scale transportation system resilience to extreme events." presented at the Transportation Research Board 94th Annual Meeting, January 2015.

Bureau of Transportation Statistics, 2014. Pocket guide to transportation. U.S. Department of Transportation, Bureau of Transportation Statistics, Washington.

Chauvet, F., Proth, J.-M., Sauer, N., et al., 1997. Management of a pool of self-service cars. Journal of Intelligent Manufacturing 8 (5), 459–465.

Chen, T. Donna, Kara M. Kockelman, and Josiah P. Hanna. 2016. Operations of a Shared, Autonomous, Electric Vehicle Fleet: Implications of Vehicle & Charging Infrastructure Decisions. Forthcoming in Transportation Research Part A.

Ciari, F., Balmer, M., Axhausen, K.W., Axhausen, K.W., Axhausen, K.W., 2008. Concepts for a Large Scale Car-sharing System: Modeling and Evaluation with an Agent-based Approach.

Cooper, G., Howe, D.A., Mye, P., 2000. The Missing Link: An Evaluation of Carsharing. Portland Inc., Portland, Oregon.

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

Du, Y., Hall, R., 1997. Fleet sizing and empty equipment redistribution for center-terminal transportation networks. Management Science 43 (2), 145–157.

Fagnant, Daniel J., and Kara M. Kockelman. "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios." Transportation Research Part C: Emerging Technologies 40 (2014): 1-13.

Fan, W.D., Machemehl, R.B., Lownes, N.E., 2008. Carsharing: dynamic decision-making problem for vehicle allocation. Transportation Research Record 2063 (1), 97–104.

Ford, Hillary Jeanette (2012) Shared Autonomous Taxis: Implementing an Efficient Alternative to Automotive Dependency. Bachelors Thesis in Science and Engineering. Princeton University.

George, D.K., Xia, C.H., 2011. Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. European Journal of Operational Research 211 (1), 198–207.

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

Kumar, V. P., Bierlaire, M., 2012. Optimizing locations for a vehicle sharing system. In: Proceedings of Swiss Transport Research Conference. http://www. strc.ch/conferences/2012/Kumar _ Bierlaire.pdf (accessed30.01.16.).

Li, X., Ma, J., Cui, J., Ghiasi, A. and Zhou, F., 2016. Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model. Transportation Research Part B: Methodological, 88, pp.21-45.

Litman, T., 2007. Parking Management: Comprehensive Implementation Guide. VTPI. www.vtpi.org

Mahmoudi, M. and Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. Transportation Research Part B: Methodological, 89, pp.19-42.

Millard-Ball, A., 2005, vol. 108. Car-Sharing: Where and How it Succeeds. Transportation Research Board.

Ropke, S., and Cordeau, J. 2009. Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows. Transportation Science 43(3):267-286. http://dx.doi.org/10.1287/trsc.1090.0272

Savelsbergh, M and Sol, M. 1998. Drive: Dynamic Routing of Independent Vehicles. Operations Research 46(4):474-490. http://dx.doi.org/10.1287/opre.46.4.474

Schrijver, A. 1998. Theory of linear and integer programming. John Wiley & Sons.

Sinha, K.C., 2003. Sustainability and urban public transportation. Journal of Transportation Engineering 129 (4), 331–341.

Uesugi, K., Mukai, N., Watanabe, T., 2007. Optimization of vehicle assignment for car sharing system. In: Knowledge-based Intelligent Information and Engineering Systems. Springer, pp. 1105–1111.

Wang, H., Cheu, R. L., & Lee, D. H. (2014). Intelligent Taxi Dispatch System for Advance Reservations. Journal of Public Transportation, 17(3), 8.