

Research Article

Evaluating the Impacts of Optimization Horizon on the Shared Autonomous Vehicle Reservation Request System

Yizhe Yang ¹, Hongjun Cui ¹, Xinwei Ma ¹, Wei Fan ², Minqing Zhu ³,
and Sheng Yao³

¹School of Civil Engineering and Transportation, Hebei University of Technology, Tianjin, China

²Department of Civil and Environmental Engineering, University of North Carolina at Charlotte, Charlotte, USA

³School of Architecture and Art Design, Hebei University of Technology, Tianjin, China

Correspondence should be addressed to Xinwei Ma; xinweima@hebut.edu.cn

Received 24 October 2021; Revised 16 December 2021; Accepted 22 December 2021; Published 7 January 2022

Academic Editor: Yanyong Guo

Copyright © 2022 Yizhe Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of technology, shared autonomous vehicles may become one of the main traffic modes in the future. Especially, shared autonomous vehicle reservation system, commuting, and other trips with fixed departure time mostly submit their travel requests in advance. Therefore, it is important to reasonably match shared autonomous vehicles and reservation demands. In this paper, reservation requests are divided into short-term and long-term requests by inputting requests in a more realistic way. An integer linear programming model considering operator scheduling cost and system service level is established. A detailed scheme considering rolling horizon continuity and ridesharing is used to improve the dispatching result. Based on traffic data in Delft, the Netherlands, 164 scenarios are tested in which the parking cost, fuel cost, ridesharing effect, service level, and network size are analyzed. The results show that a better relocation and ridesharing matching scheme can be obtained when the rolling horizon is small, while the overall effect is better when the rolling horizon is large. Moreover, the buffer time, distance, and travel time limit for vehicle relocation should be selected according to the request quantity and the calculation time requirement. The result can provide a suggestion for the dispatching of shared autonomous vehicle reservation system with ridesharing.

1. Introduction

With the rapid development of science and technology, autonomous vehicles, which can improve safety, alleviate traffic congestion, and reduce pollutant emissions and energy consumption, have become a hot issue in the automotive field in recent years. The introduction of autonomous vehicles will have a great impact on the future travel mode, especially mass transit [1]. Shared autonomous vehicles (SAVs) apply automatic driving technology to car sharing and taxi services, which is convenient for travelers: travelers do not need to get to the car sharing location before they can use it. So, SAVs can pick up passengers according to their needs and provide “door-to-door” services. Some studies have indicated that an SAV can replace 3 to 13 private vehicles depending on different system settings [2, 3]. In particular, when passengers are willing to share a vehicle with other passengers on similar

routes, the fleet size and vehicle miles travelled (VMT) of the SAV system can be further reduced [4–7]. Therefore, the wide use of SAVs can effectively reduce the number of vehicles to improve the utilization rate of vehicles. At present, SAVs have gradually begun to serve the public. Apollo GO Robotaxi, developed by Baidu, was first used in Beijing on October 11, 2020. People who are 18–60 years old can use the service, with a maximum of 2 passengers. Similar driverless taxis have also been put into trial use in Guangzhou, Dalian, Wuhan, and other cities in China. Some studies [8–11] surveyed the public’s preference for SAVs, and the optimistic situation proved the potential of the SAV traffic system. Although it remains a technical challenge to reach driving automation level 5, the rapid development of driverless technology provides the possibility of new travel modes in the future.

The SAV traffic system allows passengers to submit travel requests in real time or make an appointment in advance.

Work commuting, travel when there are fewer available SAVs, or trips arranged in advance can take advantage of making a reservation in advance to improve and guarantee the service. At present, many car sharing applications provide reservation services. Travelers can book a trip from a few hours to a few days in the future using Didi. Lyft allows customers to book a ride seven days in advance. Uber Reserve was launched in 20 cities in the United States at the end of 2020; it can be used to book a ride from 2 hours to 30 days ahead of the scheduled pickup time. Customers are matched to a driver two hours before departure, and the fare is paid in advance. Previously, Uber did not match passengers and vehicles until at least 15 minutes before the scheduled pickup time. Thus, different demand processing methods will affect the scheduling and matching results. In particular, the demand processing interval is an influential factor. If it is too long, subsequent requests may not be well arranged. But if it is too short, the processing burden of the system will be greatly increased. Therefore, it is necessary to research the matching problem between reservation requests and SAVs.

In this paper, the dispatching scheme of the SAV reservation request is optimized by combining the integer linear programming model and the idea of the rolling horizon. The scheduling costs of operator and the service level of demand are taken into account. The ridesharing scheme is proposed by service-level constraints. Using traffic data from Delft in the Netherlands, the influence of different optimization times and scopes on the matching and scheduling of SAV reservation requests, such as parking costs, fuel costs, and penalties, is tested. The effects of reservation requests, road congestion, and vehicle relocation range are also analyzed. This paper contributes to the literature in two aspects. First, in order to improve the performance of the SAV reservation system, the reservation request is divided into the short-term and long-term request. It is more realistic when requests enter the system, and it can better reflect the relative state between requests and optimization time. Second, a detailed rolling horizon continuity scheme is proposed, and the best system parameter setting under different goals is described by analyzing system scheduling cost, and service-level and operation efficiency.

The structure of the paper is as follows. In the literature review part, the demand matching and scheduling methods of traditional shared vehicle reservation services and shared autonomous vehicle reservation services are summarized, respectively, and the research motivation is clarified. Section 3 introduces the model and method considering rolling horizon continuity and ridesharing. Then, the application and the result of the method are discussed. Finally, the research conclusion and future research direction are put forward.

2. Literature Review

Real-time and reservation requests of SAVs are handled differently due to the difference in submission methods. SAV real-time requests are submitted to the system shortly before departure and need to be arranged as soon as possible. Therefore, SAV real-time request systems mostly use agent-based or activity-based models [12–17] to match vehicles

and requests to better research the dispatching of SAVs from a micro perspective. SimMobility, Transims, MATSim, or other simulation platforms are typically used. In addition, optimization models and other mathematical analysis methods [18–21] can be developed and used in SAV real-time request systems. Reservation requests are submitted to the system several hours or even days in advance; the requests are known and the system has sufficient time to create a schedule. Therefore, optimization model is usually used to deal with the matching and scheduling of the reservation demand.

2.1. Traditional Vehicle Sharing Reservation Services. The traditional one-way shared vehicle service, which is similar to the SAV service, also provides a way to submit travel requests by appointment. After the user submits a travel request in advance, if there are vehicles available at the starting station and there is a place to park at the destination, the system determines the appropriate station, and the traveler needs to travel to the station before using the vehicle. The system achieves a balance between vehicles and demands by dispatching vehicles between stations. The traditional one-way car sharing service is mostly station based, and simulation and optimization methods have been used to solve related problems. This kind of problem is relatively simple: the main focus is the matching of vehicles and requests, and the subsequent routing of vehicles is not explicitly accounted for. Alfian et al. [22] studied a one-way car sharing system including reserved and real-time requests based on discrete event simulation. There were four events for reservation requests, namely, calling, assigning, starting, and ending, and only two, starting and ending, for real-time requests. Hu and Liu [23] proposed a mixed queuing network optimization model, including customer queues, vehicle route travel queues, and vehicle virtual service queues, to study the parking capacity and fleet size of a one-way station-based car sharing system. In addition, it was assumed that there was an aggregated and exclusive route between two stations, and the relationship between travel time and the queue state of the route was an increasing function to consider road network congestion. This method of considering road congestion is simple, but a gap with reality remains. An optimization model and simulation can be used in combination. Boyaci et al. [24] proposed an integrated optimization and simulation framework to study a station-based one-way electric car sharing system. Multi-objective mixed-integer linear programming was used to optimize the service and relocation costs: two objectives were combined into one objective by weight coefficients, and the weight set was determined by the hierarchical method according to the importance of the objective. Vehicle states were constructed as spatial-temporal networks including parking, trip, and relocation. Discrete event simulation was used to judge the feasibility of the results of the operations optimization model and personnel flow model. Some studies have focused on the time and space flexibility of reservation requests. Li and Petering [25] analyzed a one-way reservation car sharing system based on discrete-time simulation. If additional

parking time at the origin and destination station was allowed, the system would be more flexible. Strohle et al. [26] studied the impact of reservation time and space flexibility on a two-way station-based car sharing system. A mixed-integer optimization minimizing the fleet size was used for the offline case. The minimum fleet size obtained in the offline case was used for the online case because of insufficient request information, and the objective of the online case was to maximize the quality of service. The results showed that the influence of spatial flexibility was greater in the latter case. Roca-Riu and Menendez [27] proposed an auction scheme with flexible vehicle reservation time to reduce the fleet size of a two-way station-based car sharing system. Drivers were given individual utility according to their preferences and could choose different travel time flexibilities. The goal of the system was to maximize the driver's utility in which drivers submitted reservation requests through bidding. Integer programming was used to allocate drivers' vehicles and travel time, and the optimization results were used to solve the pricing problem to determine how much drivers needed to pay to reserve a car. To improve the efficiency of vehicles, when to implement the matching scheme was important. Lu et al. [6] proposed a two-stage stochastic integer programming to approximately solve the multistage dynamic model. In the first stage, the number of vehicles and parking lot allocation were determined, and the dispatching decision was made in the second stage. A rolling horizon was used to improve the integer programming to obtain the optimal relocation. Demand uncertainty was considered by a spatial-temporal network and turned to a minimum cost flow problem. Molnar and Correia [28] proposed a relocation-based reservation enforcement method for the vehicle locking and relocation movements of a one-way free-floating car sharing system. That is, customers could reserve vehicles for a long time in advance, but the system would not lock the vehicles immediately. Instead, the system would arrange vehicles at an appropriate time before the demand to improve the efficiency of vehicle use. If there were idle vehicles at the station, the system would lock the vehicles; otherwise, it would consider relocating from other stations. Simulation-based optimization was used to operate the system, and the model was solved by iterated local search metaheuristics. Dou et al. [29] studied the influence of demand uncertainty on the customized reservation bus system by using a mixed-integer linear programming model. A branch-and-price method and a column-generation-based heuristic method were used to solve the model.

2.2. Shared Autonomous Vehicle Reservation Services. Compared with traditional shared vehicles, SAVs have better autonomy, and vehicle scheduling does not require drivers or other human resources. This makes the scheduling of SAVs more like a driverless taxi that completely obeys the instructions of the system. The SAV traffic system has more pickup and drop-off nodes, and they are more scattered. Vehicles are dispatched to locations with greater demand according to the balance between vehicles and demand. The

system not only focuses on vehicle matching but also considers vehicle routing and even ridesharing, so the problem is more complex, and an optimization model is typically used to address reservation-based SAV traffic systems. Wang et al. [30] proposed a trip-chaining strategy for dispatch in a reservation-based intelligent taxi system; two phases were used to solve the pickup and delivery problem. In the construction phase, the request with the pickup point closest to the drop-off point of the previous request was selected, and the trip chain was optimized by the improvement phase. The objective of the optimization model was divided into the operator's perspective and user's perspective. Liang et al. [31] proposed two integer programming models to optimize reservation requests to connect railway stations in SAV systems. One was that the demand can be rejected without punishment according to the maximization of interests, and the other was that all demands in the service area must be served. The objective was to achieve the maximal total profit, and SAV routing was implemented via a spatial-temporal network. The results indicated that the total profit was larger when the system was free to choose which requests to serve. Pimenta et al. [32] proposed integer linear programming to minimize the number of stops of autonomous electric vehicles to ensure the reliability of travel time. The study considered vehicles traveling along a closed circle network, and each parking spot on the network had a loading or unloading area. The time spent by a vehicle in this area was random and related to the number of loading or unloading operations performed in the area. Moreover, a greedy randomized adaptive search procedure heuristic based on an insertion mechanism was used to solve a static version of the problem. This static version could be regarded as a reservation-based problem. Su et al. [33] used a tabu search method combined with K-means or K-medoids clustering algorithms to minimize the total vehicle travel time and customer waiting time of reservation-based SAV systems. The objective function was decomposed into several subproblems by clustering. The results indicated that the K-medoids clustering algorithm was more suitable for large-scale problems. Some studies have studied the impact of road network congestion. Levin [3] proposed a linear program with dial-a-ride service constraints to route SAV systems with requests known in advance and used the link transmission model to reflect traffic congestion; thus, the travel time changed with the vehicle route assignment. The objective was to minimize the total system travel time, including travel time and waiting time. Since each link in the model had only upstream and downstream terminal variables, the number of variables could be reduced, and the number of variables varied with the size of the network in polynomial form. The results indicated that travel time during peak hours increased considerably without considering link congestion. Liu et al. [34] proposed a three-dimensional space-time-state path-based flow-based linear programming model for an offline shared mobility system with ridesharing, and Dantzig-Wolfe decomposition was used to solve the model. The study observed that network efficiency could be overestimated without considering congestion. However, in the process of optimization, when to optimize and determine the scheduling

scheme was subjective. Ma et al. [35] proposed a linear programming model for an SAV reservation system without ridesharing and analyzed the effect of a single-horizon and multi-horizon model on the vehicle use rate. The single-horizon model optimized the SAV system for one service horizon; that is, the system ran the optimization program every fixed period of time, and the service horizon was equal to the update horizon. The multi-horizon model allowed different service horizons, and the update horizon was equal to the shortest service horizon. The results indicated that the longer the system was considered, the higher the vehicle use rate would be. Table 1 provides a summary of the literature review.

In previous studies, reservation requests entered the system step by step, that is, the system only allowed reservations at integer hours in advance, such as 2 or 4 hours in advance. However, in reality, passengers may not submit demand just a few hours ahead of the departure time; they just need to meet the minimum reservation time allowed by the system. So, the time of demand entering the system is continuous. Besides, in the previous study, continuity of the optimization rolling horizon was considered simply. The unfinished link was just given a larger value to ensure the continuity of service, so the analysis of the system is not very in-depth. Thus, in this paper, continuous reservation time of SAV and a more detailed scheme of rolling horizon continuity is considered.

3. Method

3.1. Mathematical Model. The optimization of SAV reservation requests is a pickup and delivery problem. In this paper, only reservation requests are considered, so there is no waiting time for passengers. If the vehicle is not dispatched to the next demand after the service, it is parked on the roadside until the next request. In addition, ridesharing is allowed. If two requests want to realize ridesharing, they need to meet the following requirements:

- (1) The difference between the desired departure times of the two requests is no more than the time limit t_d . In this way, two requests with longer departure time interval will not be matched.
- (2) The vehicle should be able to reach the pickup node of the ridesharing request before the desired departure time.
- (3) The travel time delay caused by ridesharing detours should be within the acceptable range of passengers, which is defined as t_a .

After finding all possible ridesharing schemes, a greedy algorithm is used to find the ridesharing scheme with the minimum detour distance and number of trips, and the result is brought into the model for optimization. Table 2 is notations of the model.

The SAV reservation request matching network is constructed as (G, L) . As shown in Figure 1, there are six types of links in the network.

- (1) Each vehicle departs from the station to pick up requests through departure links (o, i^p)
- (2) Each vehicle finishes the service of a request through a service link (i^p, i^d)
- (3) A vehicle drops the passenger at i^d and then travels to j^p to pick up the next passenger through a relocation link (i^d, j^p)
- (4) If it is not the final optimization, each vehicle travels to a virtual node through waiting link (i^d, vd) after finishing request i
- (5) If it is the final optimization, each vehicle travels back to the station through ending link (i^d, d)
- (6) If a vehicle is not dispatched to serve a request and stays at the initial station, the vehicle is assumed to travel through virtual link (o, vd) or (o, d)

Vehicles can pick up requests if constraints (1) or (2) can be satisfied. Buffer time θ is considered because congestion may lead to travel delay, and different values can be used for peak hours and off-peak hours. Moreover, when searching for the next demand that can be served, one can consider the distance limit μ_d and travel time limit μ_t of the search range to reduce the problem size and the amount of computation.

$$t(o, i^p) + \theta \leq t i^p, \quad (1)$$

$$t i^d + t(i^d, j^p) + \theta \leq t j^p. \quad (2)$$

If a link is not a virtual link, its link capacity is one because a request needs to be served only once. For a virtual link, its capacity is the maximum available fleet size F . F is a value set in advance by the system to ensure that there are sufficient vehicles to service demands, which is not the number of vehicles actually called.

$$l_{kl} = \begin{cases} 1, & \text{if } (k, l) \in A, \\ F, & \text{if } (k, l) = (o, vd) \text{ or } (o, d). \end{cases} \quad (3)$$

According to the practical situation, the costs of different types of links are shown in equation (4). For links belonging to set A_2 , vehicles may arrive at the pickup nodes of next requests ahead of time, so parking costs are considered.

$$C_{kl} = \begin{cases} C_v + C_{fkl} \cdot \text{dis}(k, l), & \text{if } k = o, l = i^p, \\ -C_{li}, & \text{if } (k, l) \in A_1, \\ C_{fkl} \cdot \text{dis}(k, l) + C_{pkl} \cdot tp(k, l), & \text{if } (k, l) \in A_2, \\ C_{fkl} \cdot \text{dis}(k, l), K & \text{if } k = i^p, l = d, \\ 0, & \text{others.} \end{cases} \quad (4)$$

For the optimization system of SAV reservation requests, the objective function of the linear program is to minimize the cost of dispatching.

TABLE 1: Summary of literatures.

Authors	Demand			Model			Algorithm or solver
	Real-time	Reservation	Ridesharing	Vehicle-traveler match	Vehicle routing	Vehicle rebalancing	
Fagnant and Kockelman [2]	✓	×	×	Searching for the nearest vehicle	—	Four strategies based on block balance	Agent-based model simulated in C++
Wang et al. [30]	×	✓	×	Pickup and delivery problem with time Window	—	—	The two-phase solution heuristics
Alfian et al. [22]	×	✓	×	Discrete event simulation	—	—	C++
Liang et al. [31]	×	✓	×	Integer program	—	—	Branch-and-cut in Xpress
Hu and Liu [23]	×	✓	×	Mixed queuing network	Optimization	—	Genetic algorithm
Ma et al. [35]	×	✓	×	Linear program	—	—	Cplex
Levin [3]	×	✓	×	Linear program	—	—	Cplex
Mora et al. [18]	✓	×	✓	Linear program	—	—	Greedy assignment
Liu et al. [7]	✓	×	×	Searching for the nearest vehicle	Activity-based agent-based simulation	—	MATSim
Pimenta et al. [32]	×	✓	×	Integer linear program	—	—	A greedy randomized adaptive search procedure
Boyaci et al. [24]	×	✓	×	Mixed-integer linear programming	—	—	C#
Vazifeh et al. [19]	✓	✓	×	Vehicle-shareability network and path cover algorithm	—	—	Hopcroft–Karp algorithm
Li and Petering [25]	×	✓	×	Discrete event simulation	—	—	C++
Lu et al. [6]	×	✓	×	Two-stage stochastic integer programming	—	—	Branch-and-cut algorithms with mixed-integer rounding-enhanced Benders cuts
Javanshour et al. [12]	×	✓	×	Strategy	Fritzsche model	Linear program	An agent-based traffic simulation tool
Segui-Gasco et al. [14]	✓	×	✓	Agent-based simulation	—	—	MATSim and IMSim
Dandl et al. [13]	✓	×	×	Agent-based simulation	—	—	—
Strohle et al. [26]	×	✓	×	Mixed-integer optimization	—	—	Gurobi
Roca-Riu and Menendez [27]	×	✓	×	Integer programming	—	—	Branch-and-cut with Cplex
Molnar and Correia [28]	×	✓	×	Optimization with a variable quality of service model	—	Simulation-based optimization	Java
Liu et al. [34]	×	✓	×	Integer linear program	—	—	Dantzig-Wolfe decomposition.
Dou et al. [29]	×	✓	✓	Mixed-integer linear programming	—	—	MATLAB
Winter et al. [17]	✓	×	×	Searching for the nearest vehicle	—	Three pro-active relocation strategies	MATSIM
Nahmias-Biran et al. [16]	✓	×	✓	Activity-based accessibility measure	Within-day simulation	—	SimMobility
Liang et al. [21]	✓	×	✓	Integer nonlinear programming model (INLP)	Lagrangian relaxation algorithm with INLP	—	Xpress
Su et al. [33]	×	✓	×	Integer program	—	—	A metaheuristic tabu search method

TABLE 2: Notations.

Sets	
G	Set of vehicles and requests.
L	Set of links.
E^n	Set of the requests optimized in the current optimization, $\forall n \in N$.
$E^{n_{al}}$	Set of the newly arrived long-term reserved requests for the current optimization.
$E^{n_{as}}$	Set of the newly arrived short-term reserved requests for the current optimization.
E^{n_b}	Set of the unfinished reservation requests from the last optimization.
E^{n_c}	Set of the unimplemented reservation requests from the last optimization.
I^p	Set of pickup nodes i^p .
I^d	Set of drop-off nodes i^d .
H	Set of all kinds of nodes.
I	Set of pickup and drop-off nodes, $I^p, I^d \in I$.
O	Set of initial station nodes.
D	Set of final station nodes.
O^p	Set of nodes o^p .
O^d	Set of nodes o^d .
A	Set of links except the virtual link (o, vd) and (o, d) .
A_1	Set of service links (i^p, i^d) .
A_2	Set of relocation links (i^d, j^p) .
Parameters	
N	Optimization horizons in an operation day.
i^p, i^d	Pickup or drop-off node of request $i, i \in E^n$.
o	Initial station node, $o \in O$.
d	Final station node, $d \in D$.
o^p	Initial node comes from i^p node of last optimization.
o^d	Initial node comes from i^d node of last optimization.
vd	Sink node, $vd \in D$.
ti^p, ti^d	Time that a vehicle arrives at node i^p or $i^d, i^p \in I^p, i^d \in I^d$.
to^p, to^d	Time that a vehicle arrives at node o^p or o^d .
$t(k, l)$	The minimum travel time from node k to l .
$dis(k, l)$	The travel distance between node k and l .
$tp(k, l)$	The parking time on the link (k, l) .
T^r	The time of the rolling horizon.
T^h	The time of the considered horizon.
C_{kl}	Cost of using link $(k, l), k, l \in I$.
C_{li}	Cost of losing request i .
C_{fld}	Fuel cost from node k to $l, k, l \in I$.
C_{pkl}	Parking cost from node k to $l, k, l \in I$.
C_v	Cost of vehicle depreciation.
F	Fleet size.
l_{kl}	Link capacity of $(k, l), k, l \in I$.
θ	Buffer time for vehicle relocation.
μ_d	Distance limit for vehicle relocation.
μ_t	Travel time limit for vehicle relocation.
f_{sl}	Proportion of short-term reservation requests to total requests.
Decision variables	
x_{kl}	The number of vehicles using link $(k, l), k, l \in I$.

$$\min \sum_{k, l \in H} C_{kl} x_{kl}, \quad (5)$$

$$x_{kl} \leq l_{kl}, \quad \forall k, l \in H \quad (6)$$

$$\sum_{o \in O} x_{oi^p} + \sum_{j^d \in I^d} x_{j^d i^p} = x_{i^p i^d}, \quad \forall i \in I, i^p \in I^p, i^d \in I^d, \quad (7)$$

$$x_{i^p i^d} = \sum_{m^p \in I^p} x_{i^d m^p} + \sum_{v \in D} x_{i^d v} + \sum_{d \in D} x_{i^d d}, \quad \forall i \in I, i^p \in I^p, i^d \in I^d, \quad (8)$$

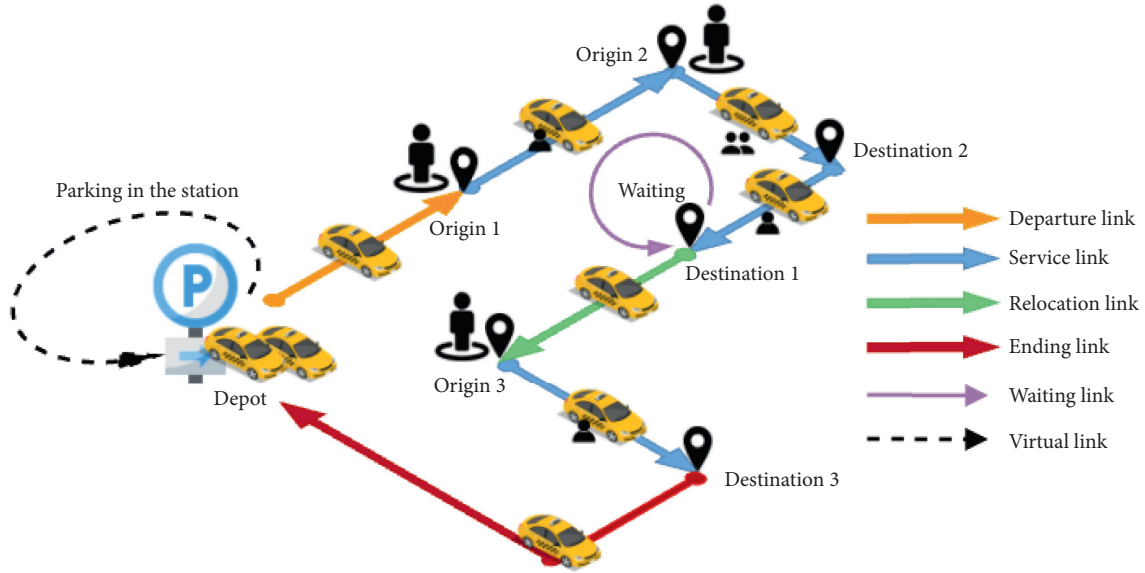


FIGURE 1: Links of the reservation-based SAV traffic system.

$$\sum_{i^p \in I^p} x_{oi^p} + \sum_{v \in D} x_{ovd} + \sum_{d \in D} x_{od} = F, \quad \forall o \in O, \quad (9)$$

$$\sum_{i^d \in I^d} x_{i^d vd} + \sum_{o \in O} x_{ovd} + \sum_{o \in O} x_{od} + \sum_{i^d \in I^d} x_{i^d d} = F, \quad \forall d \in D, \quad (10)$$

$$x_{kl} \geq 0, \quad \forall k, l \in H \quad (11)$$

Constraints (6) are constraints for link capacity. Constraints (7)–(10) are flow conservation constraints. For a pickup node $i^p \in I^p$, its previous node can be the initial station node o or another requests' drop-off node j^d , and its subsequent node must be the corresponding pickup node i^p . For a drop-off node $i^d \in I^d$, its previous node must be the corresponding pickup node i^p , and its subsequent node can be the final station node d , sink node vd , or other requests' pickup node j^p . Moreover, if a vehicle visited a pickup node i^p or a drop-off node i^d , it indicates that request i must be served. For an initial station node o , all of its available vehicles must be dispatched to requests, final station node d , or sink node vd . For a final station node d or a sink node vd , all vehicles must return from drop-off nodes or initial station nodes. Constraints (11) are constraints for variables.

3.2. Rolling Horizon Framework and Continuity. A rolling horizon framework is used to optimize the service of SAVs. The rolling horizon framework divides the operation day into N horizons, and the model described previously in every horizon is optimized. Then, the optimization horizon rolls forward with a specific rolling length and reaches the next optimization horizon. The rolling horizon framework is shown in Figure 2. If the desired departure time of the request is in the considered horizon, the request will be

handled in the current optimization, and the system optimizes requests every T^r time length.

The SAV reservation system in this paper allows passengers to submit reservations at least 2 hours and up to 24 hours in advance. Therefore, short-term reservation requests will not enter the system until 120 minutes after each optimization. In each optimization, there are two types of requests that must be addressed: long-term and short-term reservation requests. Compared with the optimization time, long-term reservation requests submitted earlier are relatively fixed, while short-term reservation requests submitted recently enter the system gradually. Short-term reservation requests are submitted after the start of the system, and the desired pickup time of requests is 2 hours later. In each optimization, only recently submitted requests with the desired departure time in the considered horizon are regarded as short-term reservation requests. Short-term reservation requests with a desired departure time after the considered horizon are considered as long-term reservation requests relative to the optimization time, because by the time such requests are processed, there are already requests submitted earlier than the optimization time. In Figure 3, the shadow part represents short-term reservation requests considered in each optimization when the considered horizon is less than the sum of the rolling horizon and the minimum reservation time. Requests with a departure time in the

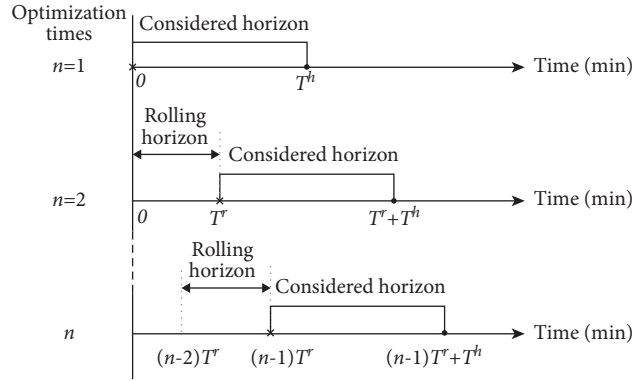


FIGURE 2: The rolling horizon framework.

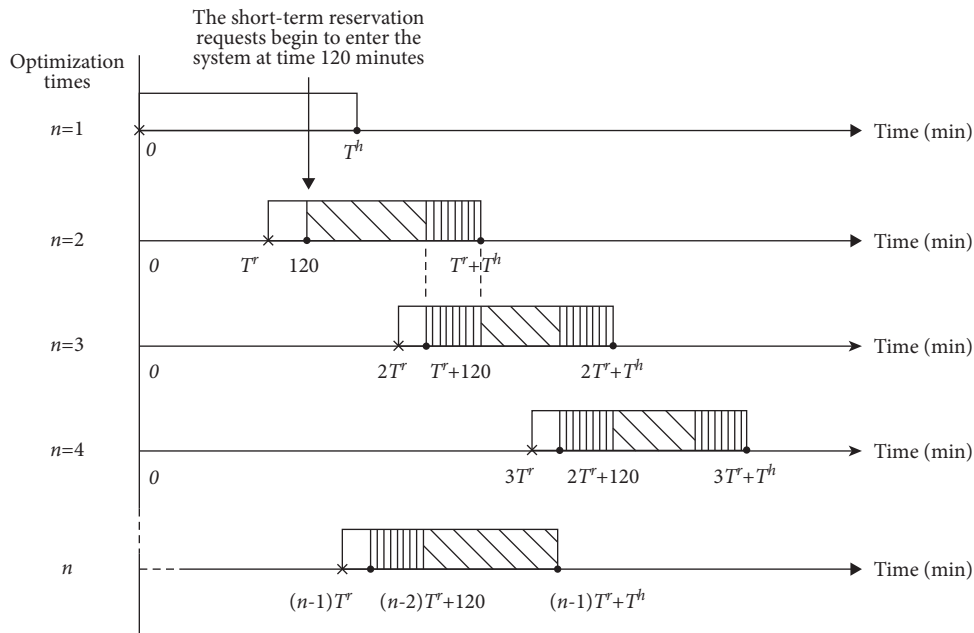


FIGURE 3: The rolling horizon framework of short-term reservation requests.

diagonal shadow area enters the system once, and requests with a departure time in the vertical shadow area are divided into two parts and enter the system in sequence.

The length of the rolling horizon and considered horizon affect the optimization matching results. As shown in Figure 4, the considered horizon is 60 minutes; when $t = 0$, vehicle 1 successively serves request 1 and request 4, vehicle 2 serves request 2, and vehicle 3 serves request 3. When $t = 15$, request 5 reserves a trip with the departure time of 135 minutes. If the rolling horizon is 120 minutes, when the system optimizes at time 120 minutes, vehicle 1 is on the way to the pickup node of request 4. Assume that vehicle 1 does not change the path, the optimization result of vehicle 3 is to serve request 5 after request 3 is completed. If the rolling horizon is 90 minutes, the system performs a second optimization at time 90 minutes. Vehicle 1 is serving request 1, since it has not reached the destination of request 1, and the subsequent path of vehicle 1 can be changed. At this time, the scheduling result is that vehicle 2 picks up request 4 after

serving request 2, and vehicle 1 waits in the same place after serving request 1 and then it picks up request 5. Therefore, different rolling horizons bring about different scheduling results due to the vehicle states. A smaller rolling horizon can ensure a quicker response to short-term reservation requests if the considered horizon is large enough. The optimal schedule might be better if more future requests are included.

At each optimization moment after the first optimization, some links might be in an unfinished state, and vehicles can be traveling between nodes. To ensure the continuity of service, it is necessary to clarify the initial positions of vehicles in the next optimization and the time of arrival at these nodes; that is, the vehicles would start at that time in the next optimization and travel from the initial nodes to complete the subsequent services. Therefore, we created new link types: (o^d, i^p) , (o^p, i^d) , (o^d, vd) , and (o^d, d) . The origins and possible link types for the unfinished links in the n th optimization are shown in Figure 5.

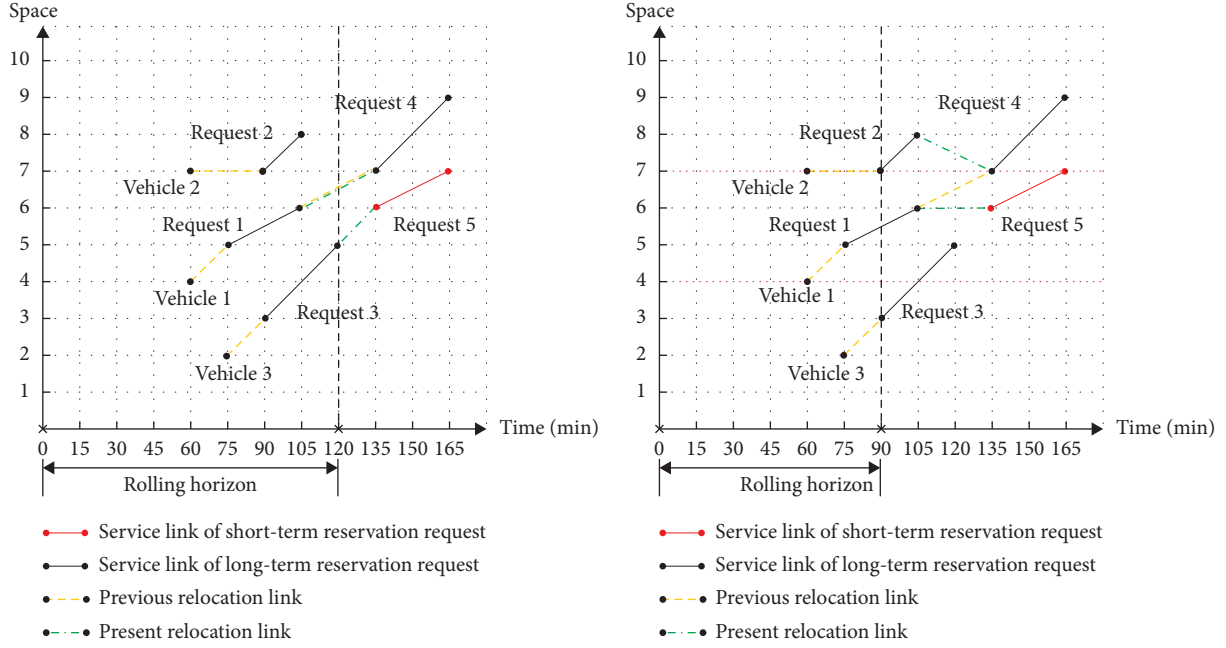


FIGURE 4: Comparison of different rolling horizon scheduling results.

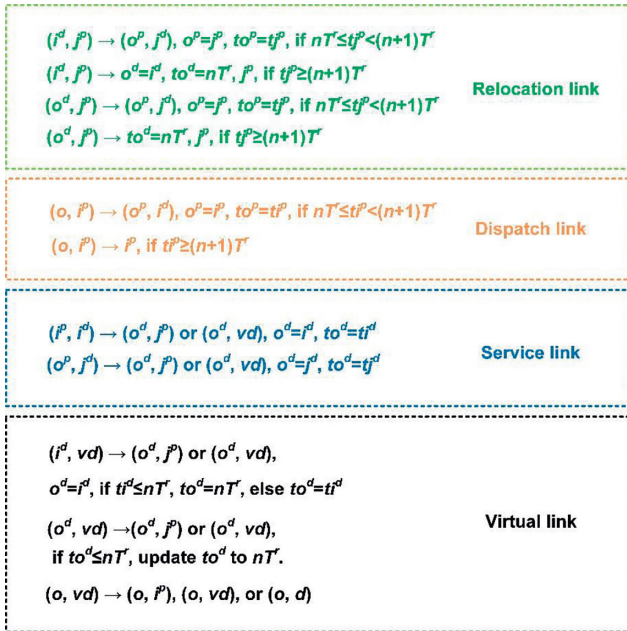


FIGURE 5: Processing scheme for unfinished links.

In a new optimization, o^p and o^d are the initial nodes, which means there are no upstream links to reach them. Link (i^d, vd) or (o^d, vd) can be understood as the vehicle stopping at the same place after dropping off a request until the next service. Links (o, i^p) , (i^d, j^p) , and (o^d, j^p) with pickup times larger than the next optimization time are added to the set of unimplemented reservation requests from the previous optimization E^{n_c} to achieve better optimization results. Links (i^d, d) and (o, d) only exist in the final optimization, so their continuities are not considered. Thus, in the n th optimization, the request that

needs to be optimized is $E^n = E^{n_{al}} \cup E^{n_{as}} \cup E^{n_{b}} \cup E^{n_c}$, where E^{n_b} and E^{n_c} include both short-term and long-term reservation requests.

Based on service continuity, the model is modified as follows:

$$l_{kl} = \begin{cases} 1, & \text{if } (k, l) \in A, \\ F, & \text{if } (k, l) = (o, v d), n = 1, \\ F_o, & \text{if } (k, l) = (o, v d) \text{ or } (o, d), 1 < n \leq N, \end{cases} \quad (12)$$

where F_o is the number of vehicles that pass through link (o, vd) after each optimization; it equals the number of vehicles that stop at the initial stations and are not allocated to travel demands.

Due to the addition of new types of links, the cost is changed to

$$C_{kl} = \begin{cases} C_v + C_{fkl} \cdot \text{dis}(k, l), & \text{if } k = o, l = i^p, \\ -C_{li}, & \text{if } (k, l) \in A_1, \\ C_{fkl} \cdot \text{dis}(k, l) + C_{pkl} \cdot tp(k, l), & \text{if } (k, l) \in A_2, \\ C_{fkl} \cdot \text{dis}(k, l) + C_{pkl} \cdot tp(k, l), & \text{if } k = o^d, l = d, \\ C_{pkl} \cdot \text{dis}(k, l), & \text{if } k = o^d, l = v d, \\ C_{fkl} \cdot \text{dis}(k, l), & \text{if } k = i^p, l = d, \\ 0, & \text{others,} \end{cases} \quad (13)$$

where link (o^p, i^d) is added to A_1 and link (o^d, j^p) is added to A_2 . In the n th optimization, for a link that transforms to (o^d, d) , (o^d, vd) , or (o^d, j^p) , if the drop-off time of its original corresponding node is less than nT^r , the parking cost from ti^d to to^d must be considered.

When $n > 1$, constraints (7) should be modified to

$$\sum_{o \in O} x_{oip} + \sum_{o^d \in O^d} x_{o^d ip} + \sum_{j^d \in J^d} x_{j^d ip} = x_{ipj^d}, \quad \forall i \in I, i^p \in I^p, i^d \in I^d \quad (14)$$

Vehicles that can serve request i come not only from the current optimization but also from the previous optimization. Moreover, when $n > 1$, constraints (15) and (16) must be added. In each optimization, vehicles on node o^d have one of the following three states: traveling to the next request, waiting for the next request, or driving to the station. Vehicles at node o^p must finish this request.

$$\sum_{i^p \in I^p} x_{o^d i^p} + \sum_{v \in D} x_{o^d vd} + \sum_{d \in D} x_{o^d d} = 1, \quad o^d \in O^d, \quad (15)$$

$$\sum_{j^d \in J^d} x_{o^p j^d} = 1, \quad o^p \in O^p. \quad (16)$$

4. Case Study

The model was applied to Delft in the Netherlands. The data came from the Dutch mobility dataset (MON 2007/2008), which was provided by the Dutch government for academic research and could be obtained free of charge on the Internet [36]. By filtering the trips completed by taxis and cars within the surveyed day in Delft city, 1112 available travel demands were obtained from 22,240 data points. The dataset included information on travel requests and traffic networks. Short-term and long-term reservation requests were randomly selected from the data in proportion. As shown in Figure 6, there are 46 points and 66 links in the network. The initial stations of vehicles are node 15, node 16, node 19, node 24, and node 27, which are located in the middle and surroundings of the study area, and the vehicles are evenly distributed at the five stations. According to the time of travel data, the running time of the system is from 6:30 to 24:00, and the time step is 1 minute. The buffer time is related to the link length. The default value of the vehicle relocation search limit is set to a large number to obtain a larger search scope, and ridesharing allows no more than 2 passengers. Table 3 shows the parameter values.

The effect of different parameter values on the system scheduling results was tested. The rolling horizon was varied from 15 to 120 minutes, the considered horizon was varied from 15 to 360 minutes, the test interval was 15 minutes, and the considered horizon of each scene could not be less than the rolling horizon. The maximum rolling horizon was considered to be 120 minutes since travel requests in the SAV reservation request system had to be submitted at least 2 hours in advance. Moreover, the maximum value of the considered horizon was 6 hours. MATLAB and a PC with 128 GB memory and an Intel (R) Xeon (R) Gold 5220 CPU @ 2.20 GHz 2.19 GHz processor were used.

5. Results and Discussion

5.1. Impact of the Optimization Horizon on the Costs. As shown in Table 4, the parking cost of the SAV reservation request system includes four components. The results of

total parking cost and parking costs 1, 2, 3, and 4 are shown in Figure 7. The total parking cost with ridesharing is influenced by both the rolling horizon and the considered horizon; moreover, the total parking cost is greatly affected by parking cost 4. Even though parking costs 1, 2, and 3 are smaller when the rolling horizon is small, parking cost 4 is larger, which makes the total parking cost have a similar trend to that of parking cost 4: decreasing with an increase in the rolling horizon. Furthermore, as the rolling horizon increases further, the change in total parking cost becomes less obvious. When the rolling horizon is equal to the considered horizon, since the demand of the next optimization is not considered completely in each optimization, the total parking cost is small and will change significantly when the considered horizon begins to increase. The total parking cost decreases and tends to become stable as the considered horizon increases. Parking costs 1 and 2 increase sharply when the considered horizon begins to exceed the rolling horizon because the link related to the subsequent request suddenly increases, as does the parking cost. However, as the considered horizon continues to increase, the optimization results of parking costs 1 and 2 gradually stabilize. In addition to the influence of parking time, parking cost 4 is determined mainly by the number of unscheduled vehicles, that is, the number of repeated o^d nodes. Vehicles become more likely to be used by subsequent requests as the considered horizon increases, which leads to a decrease in the number of repeated o^d nodes and gradual flattening, and parking cost 4 also decreases. As the rolling horizon increases, parking cost 4 decreases, and the gap becomes less obvious because o^d nodes are easier to repeat when the optimization interval is small.

Figure 8 shows the fuel cost results, and meanings of fuel cost 1–6 are shown in Table 4. The total fuel cost is affected by the considered horizon and rolling horizon. As the considered horizon increases, the total fuel cost first surges and then stabilizes, while the rolling horizon has a smaller effect. These results are caused by the costs of different links. The sum of fuel costs 1 and 2 in different rolling horizons and considered horizons is basically fixed because they represent the fuel cost generated by serving the demand. So, their images are symmetrical. Fuel cost 1 increases with increasing rolling horizon, whereas fuel cost 2 decreases. Because the optimization interval is large when the rolling horizon is large, the request is more likely to be served in the previous optimization. When the considered horizon is small, requests for the next optimization period are not considered or less considered, there are no or only a small number of o^p nodes; thus, fuel cost 2 is small at the beginning, and the subsequent optimization results are stable, while fuel cost 1 shows the opposite trend. The change in fuel costs 3 and 4 with the considered horizon is also related to the number of o^p nodes. It is easier to generate the corresponding links of fuel costs 3 and 4 at the beginning of considering the request of the subsequent optimization period, and the optimization result is stable when the considered horizon is approximately 60 minutes larger than the rolling horizon. The number of new demands in each optimization decreases and the fuel cost 4 is



FIGURE 6: Network of Delft, the Netherlands.

TABLE 3: Parameter values.

$F = 1000$	$C_{li} = 100$ euro/km
$\theta = 4$ min/km in peak hours	$C_{fkl} = 0.1$ euro/km
$\theta = 2$ min/km in off-peak hours	$C_{pkl} = 0.01$ euro/min
$f_{sl} = 0.5$	$C_v = 20$ euro/vehicle/day
$t_d = 10$ min	$t_a = 10$ min

TABLE 4: Components of parking costs and fuel costs.

Components	Description
Parking cost 1	The vehicle on link (i^d, j^p) arrives at the pickup node of the request in advance and waits for departure.
Parking cost 2	The vehicle on link (o^d, j^p) waits for the next demand to depart.
Parking cost 3	The vehicle finishes the request and parks in place until the next optimization.
Parking cost 4	The idle vehicle parks in place and is not dispatched.
Fuel cost 1	Fuel cost for vehicles traveling on link (i^p, i^d) .
Fuel cost 2	Fuel cost for vehicles traveling on link (o^p, j^d) .
Fuel cost 3	Fuel cost for vehicles traveling on link (i^d, j^p) .
Fuel cost 4	Fuel cost for vehicles traveling on link (o^d, j^p) .
Fuel cost 5	Fuel cost for vehicles traveling on link (o, i^p) .
Fuel cost 6	Fuel cost for vehicles traveling on link (i^d, d) .
Fuel cost 7	Fuel cost for vehicles traveling on link (o^d, d) .

smaller as the rolling horizon increases. The particular trend of the rolling horizon equal to 120 minutes is due to a large number of unsatisfied requests, as detailed in the next paragraph. The rolling and considered horizon have little influence on fuel cost 5, 6, and 7, so they are not plotted.

The fare is calculated based on the travel distance. As shown in Figure 9, the fare is very small when the considered horizon is equal to the rolling horizon. Because the pickup time of some reservation requests is within a short time after the optimization time, especially short-term reservation requests, there is a high probability that no suitable vehicle is

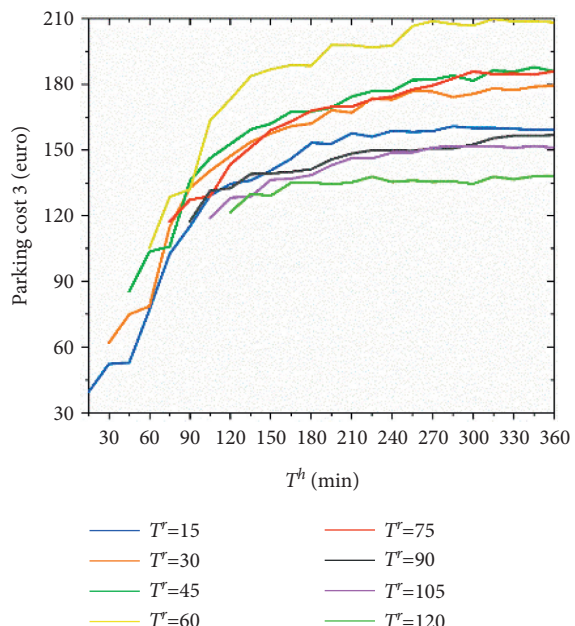
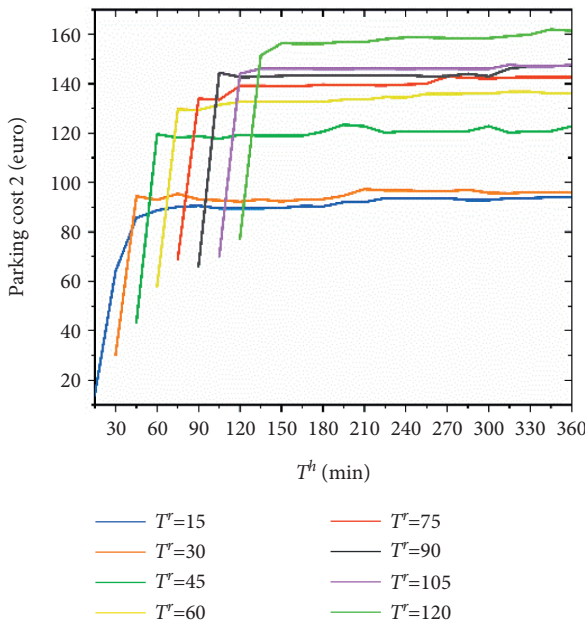
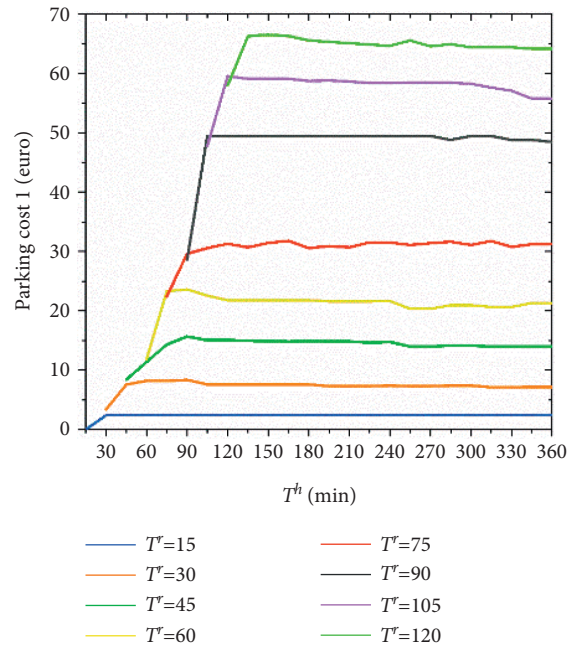
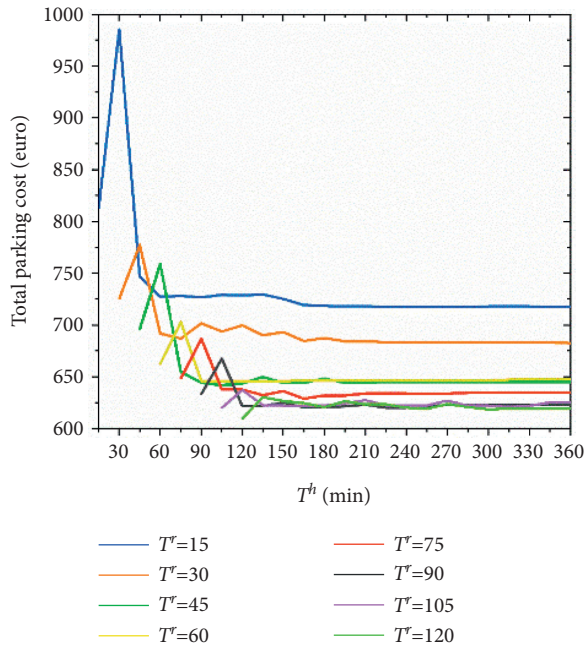


FIGURE 7: Continued.

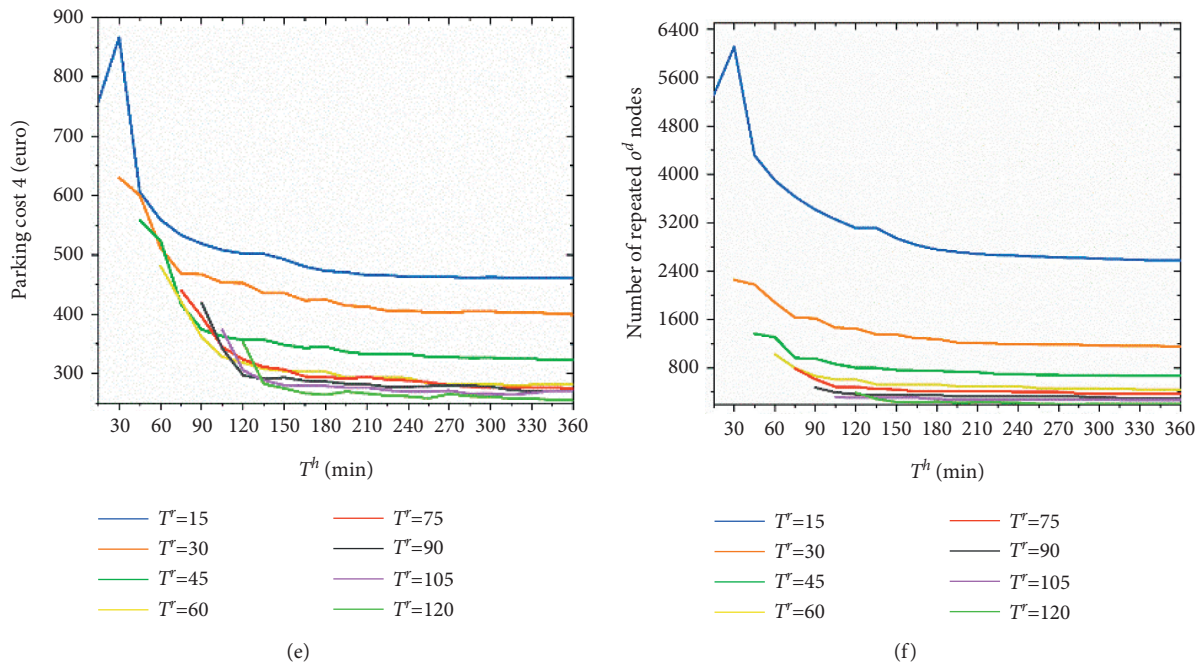
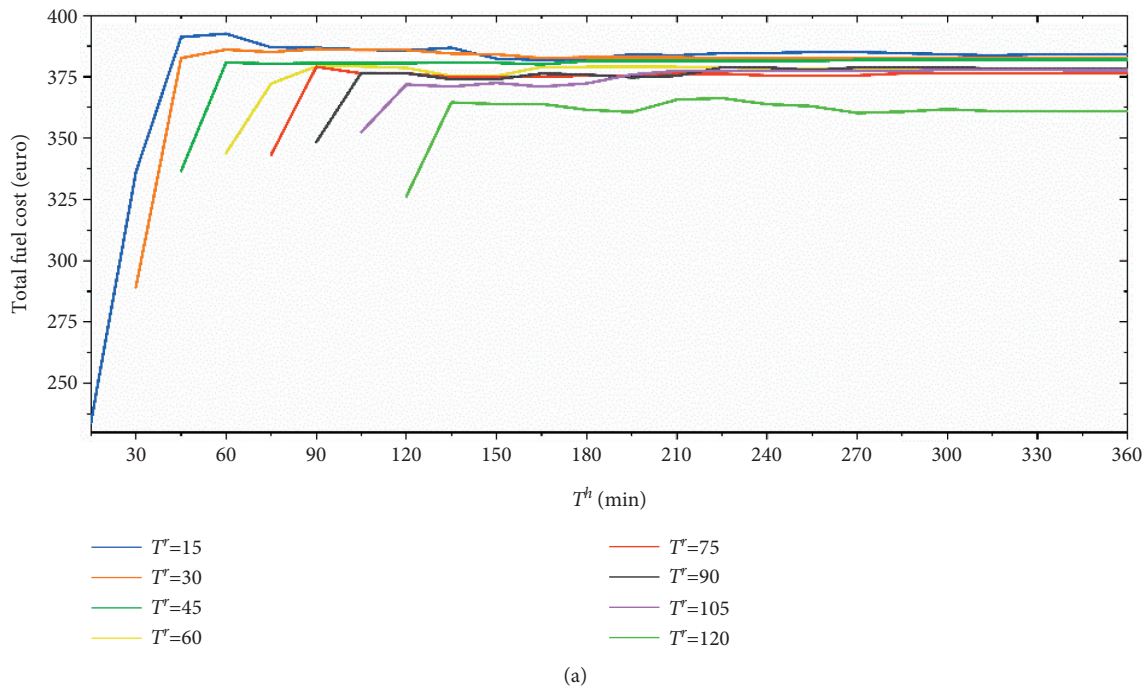


FIGURE 7: Results of parking costs.



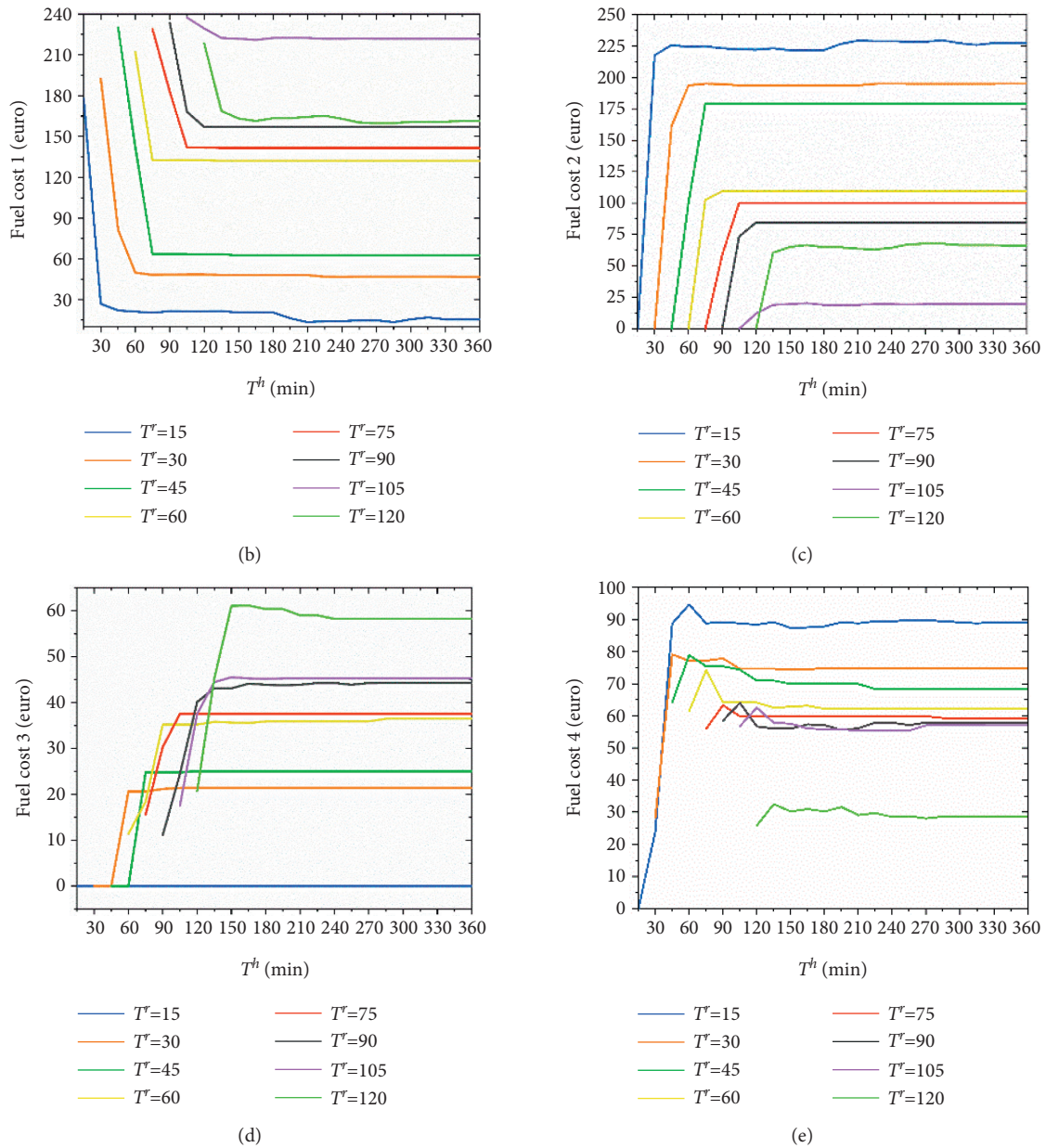


FIGURE 8: Results of fuel costs.

available for service. Fewer requests can be served, so the fare is naturally less. When the rolling horizon is 120 minutes, there are still many requests with a pickup time shortly after the optimization that cannot be served, so the results are much smaller than those of other rolling horizons. The reason is that the rolling horizon is the same as the shortest reservation time, even if the considered horizon is increased. In other rolling horizons, all the submitted requests can be served when the considered horizon increases to a certain value; therefore, the difference between the stable results is very small.

The total cost of system dispatching is the sum of the fuel cost, parking cost, vehicle depreciation cost, and penalty cost. As shown in Figure 10, the total cost decreases with increasing rolling horizon and gradually stabilizes. The total

cost increases due to the high penalty when the rolling horizon is 120 minutes. With an increase in the considered horizon, the penalty cost and parking cost decrease, and the total cost also decreases.

5.2. Impacts of the Optimization Horizon on Ridesharing and Service. As shown in Figure 11, a smaller rolling horizon and larger considered horizon can decrease VMT. Because more subsequent requests can be considered, the optimization scheme can be adjusted rapidly. In addition, as shown in Figure 12, taking a rolling horizon of 60 minutes and a considered horizon of 120 minutes as an example, the system scheduling cost is substantially reduced after considering ridesharing.

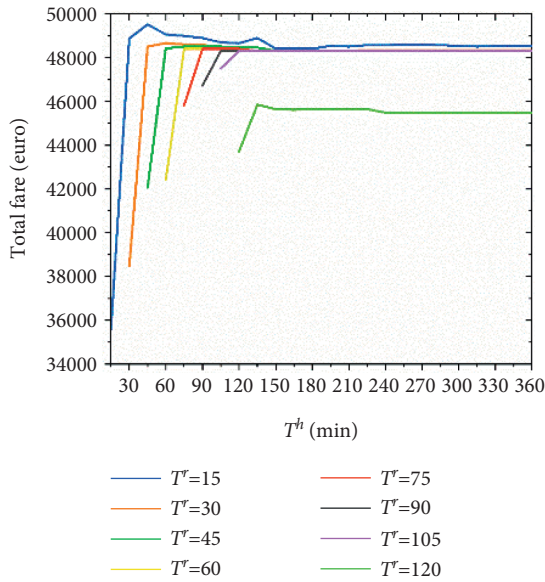


FIGURE 9: Results of fare.

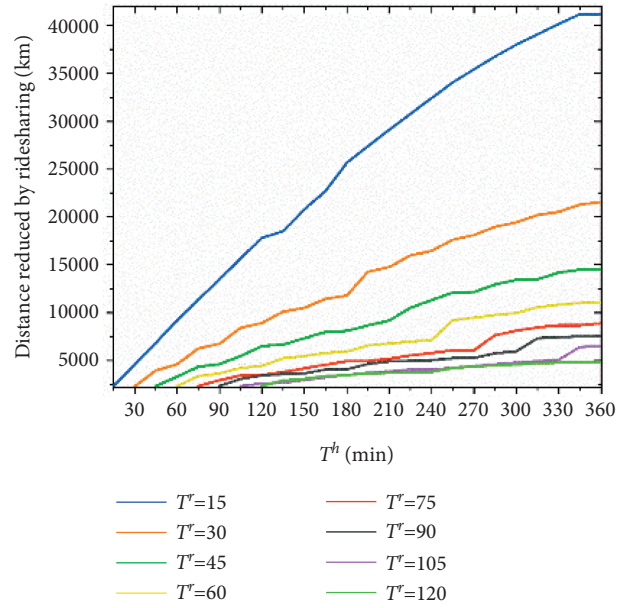


FIGURE 11: Results of distance reduced by ridesharing.

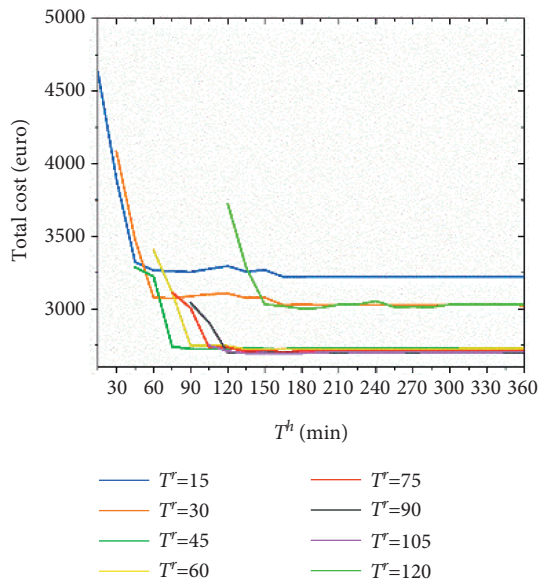


FIGURE 10: Results of total cost.

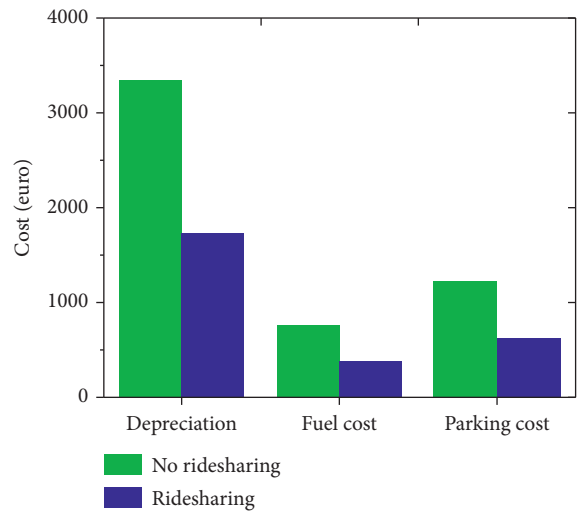


FIGURE 12: Cost comparison between ridesharing and no ridesharing.

The proportion of short-term reservation requests affects the quantity of unserved demand when the rolling horizon is 120 minutes. As shown in Figure 13, with an increase in short-term reservation requests, the amount of unserved demand increases. When the considered horizon is 120 minutes, some reservation requests with a pickup time close to the optimization time cannot find suitable vehicles, so the unserved demand is the same and not affected by the proportion of short-term reservation requests. With an increase in the considered horizon, the long-term reservation requests between the considered horizon and the rolling horizon in each optimization can be serviced, resulting in a considerable difference in the

quantity of unserved demand between the considered horizon of 120 and 135 minutes.

5.3. Impacts of Buffer Time and Vehicle Relocation Search Limit. As shown in Figure 14, when the rolling horizon is 60 minutes, the network size increases with an increase in the considered horizon because more requests and more possible links must be considered. Table 5 shows the specific value of network change, where the network size is reduced compared with when the buffer time is 4 min/km during peak hours. This result indicates that increasing the buffer time could greatly reduce the network size, thereby reducing the system operation time. However, as shown in Figure 15, this approach will increase the overall operation cost of the

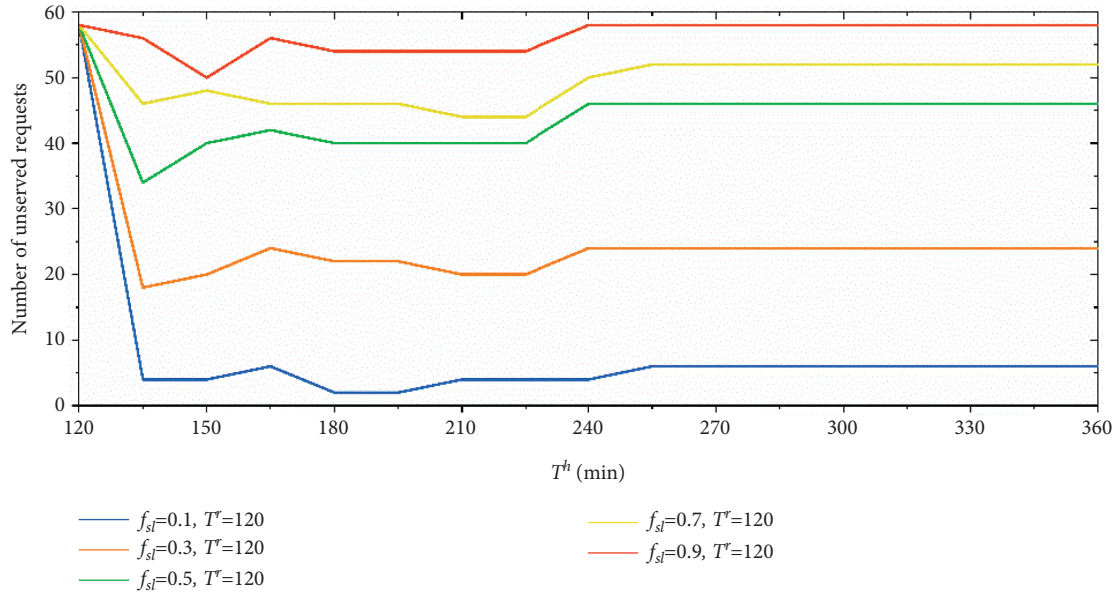


FIGURE 13: Results of the number of unserved requests.

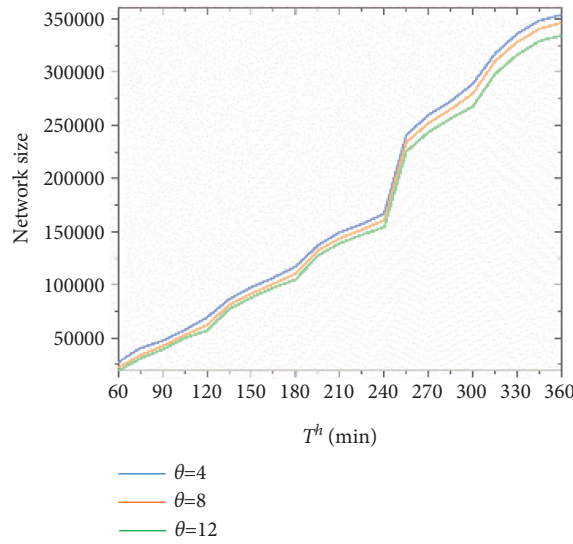


FIGURE 14: Results of network size with different buffer times.

TABLE 5: The influence of buffer time on network size when the rolling horizon is 60 minutes.

θ	$T^r = 60 \text{ min}$										
	$T^h \text{ (min)}$										
min	60	90	120	150	180	210	240	270	300	330	360
8	4913	4908	7066	5954	6559	5571	6013	7644	9108	7731	7407
12	8087	8122	12280	9592	12172	10206	12713	16161	21400	19760	19364

system because a large buffer time makes it more difficult for the vehicle to find an appropriate request when relocating. The fuel cost will decrease, but the vehicle will stop for a longer period of time and the parking cost will increase. Part of the requests also needs vehicles from the station because

vehicles that have completed service cannot satisfy them, thereby increasing the fleet size and the total dispatching cost. Therefore, if the increase in travel time caused by congestion is not considered, a more optimistic cost will be obtained, which deviates from reality.

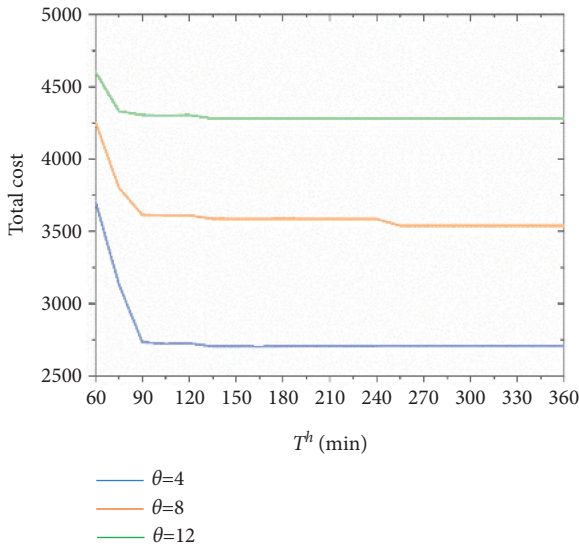


FIGURE 15: Results of the total cost with different buffer times.

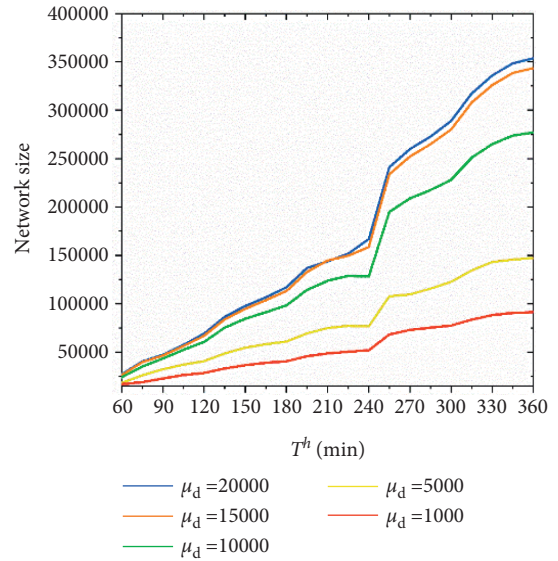


FIGURE 17: Results of network size with different distance limits.

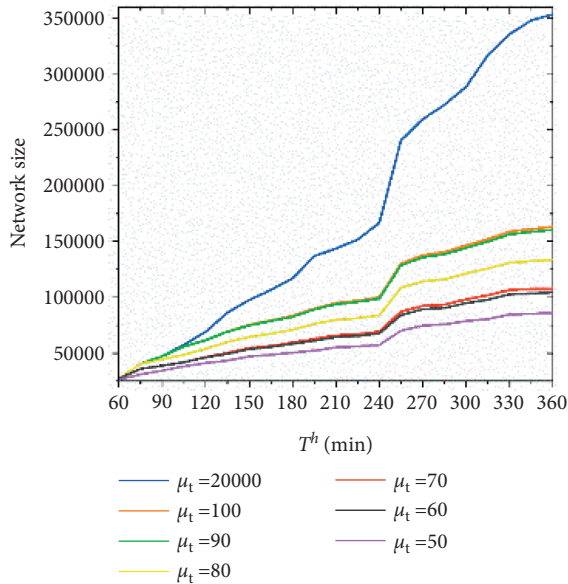


FIGURE 16: Results of network size with different travel time limits.

The relocation travel time limit and distance limit of the basic scenario were set to 50000 minutes and 50000 meters, respectively. A very large travel time limit does not affect the network size, so the relocation was not limited by the travel time or distance limit. Taking a rolling horizon of 60 minutes

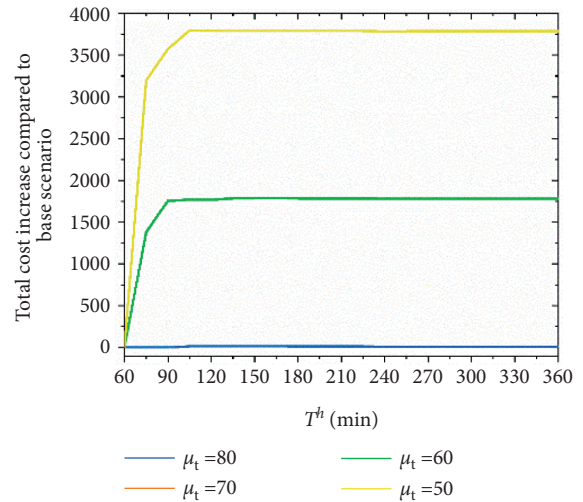


FIGURE 18: The total cost increases with different travel time limits.

as an example, as shown in Figures 16 and 17, setting smaller limits for vehicle relocation can effectively reduce the network size of the system, especially when the considered horizon is large. However, as shown in Figures 18 and 19, the same as when buffer time is considered, the price of reducing the network size is an increase in the total system operation cost. The results of the travel time limit of 60 minutes and 70 minutes in Figure 18 coincide.

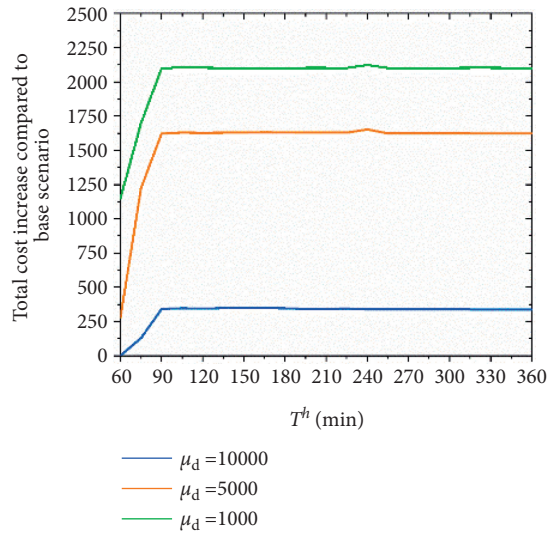


FIGURE 19: The total cost increases with different relocation distance limits.

6. Conclusion

Based on the integer linear programming model of SAV reservation request scheduling and continuity consideration of using a rolling horizon to optimize the request, the influence of different rolling horizons and considered horizons on the system scheduling results was tested in this study. When the considered horizon is just larger than the rolling horizon, it has a considerable impact on the total system costs, which then become essentially stable. A smaller rolling horizon can better link requests, but the overall effect is not satisfactory. A large rolling horizon can reduce the operation cost of the system, and it should be controlled to be less than the minimum reservation time; otherwise, it will cause a large penalty cost. Ridesharing can substantially reduce the operating cost of the system, and a smaller rolling horizon and a larger considered horizon can better reduce VMT. Therefore, compared with previous studies, a smaller rolling horizon does not necessarily indicate a better performance. The goal that needs to be achieved must be considered. The proportion of short-term reservation requests affects the service level when the rolling horizon exceeds the minimum reservation time. Considering the large buffer time and vehicle relocation search limits, the network size can be reduced, and the operation cost increases. Possible matching links can be fully searched to achieve a best scheduling scheme when the number of SAV reservation requests is not large. As the demand of the SAV reservation increases, the search range of relocation can be reduced appropriately to achieve better scheduling timeliness.

This study has several limitations. First, this paper used only the buffer time to express the congestion state of the road network at peak hours and off-peak hours. Second, the matching problem of SAV reservation request is considered as a pickup and delivery problem, and specific vehicle routing is not carried out. To make a more accurate consideration of travel time, future studies can combine the

vehicle routing problem with the real traffic flow model on the premise of calculation timeliness.

Data Availability

Previously reported (travel and network) data were used to support this study and are available at 10.13140/RG.2.2.11097.83043. These prior studies (and datasets) are cited at relevant places within the article as references [36] G. Homem, D. A Correia, and B. V. Are. Trips and network of the case-study city of Delft, 2016. <https://doi.org/10.13140/RG.2.2.11097.83043>.

Conflicts of Interest

No potential conflicts of interest were reported by the authors.

Acknowledgments

This research was part of the project of the National Natural Science Foundation of China (grant no. 52172304); the Youth Project of National Natural Science Foundation of China (grant no. 51908187); and the Bureau of Science and Technology of Hebei Province (grant no. 19970808D).

References

- [1] F. Zhang, W. Liu, G. Lodewijks, and S. T. Waller, "The short-run and long-run equilibria for commuting with autonomous vehicles," *Transportation Business: Transport Dynamics*, vol. 28, pp. 1–28, 2020.
- [2] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 1–13, 2014.
- [3] M. W. Levin, "Congestion-aware system optimal route choice for shared autonomous vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 82, pp. 229–247, 2017.
- [4] A. T. Moreno, A. Michalski, C. Llorca, and R. Moeckel, "Shared autonomous vehicles effect on vehicle-km traveled and average trip duration," *Journal of Advanced Transportation*, vol. 2018, pp. 1–10, 2018.
- [5] Y. Zhao and K. M. Kockelman, "Anticipating the regional impacts of connected and automated vehicle travel in austin, Texas," *Journal of Urban Planning and Development*, vol. 144, no. 4, 2018.
- [6] M. Lu, Z. Chen, and S. Shen, "Optimizing the profitability and quality of service in carshare systems under demand uncertainty," *Manufacturing & Service Operations Management*, vol. 20, no. 2, pp. 162–180, 2018.
- [7] J. Liu, K. M. Kockelman, P. M. Boesch, and F. Ciari, "Tracking a system of shared autonomous vehicles across the Austin, Texas network using agent-based simulation," *Transportation*, vol. 44, no. 6, pp. 1261–1278, 2017.
- [8] T. D. Chen, K. M. Kockelman, and J. P. Hanna, "Operations of a shared, autonomous, electric vehicle fleet: implications of vehicle & charging infrastructure decisions," *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 243–254, 2016.
- [9] P. Bansal, K. M. Kockelman, and A. Singh, "Assessing public opinions of and interest in new vehicle technologies: an

- Austin perspective,” *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 1–14, 2016.
- [10] R. Krueger, T. H. Rashidi, and J. M. Rose, “Preferences for shared autonomous vehicles,” *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 343–355, 2016.
 - [11] Z. Wang, M. Safdar, S. Zhong, J. Liu, and F. Xiao, “Public preferences of shared autonomous vehicles in developing countries: a cross-national study of Pakistan and China,” *Journal of Advanced Transportation*, vol. 2021, pp. 1–19, 2021.
 - [12] F. Javanshour, H. Dia, and G. Duncan, “Exploring the performance of autonomous mobility on-demand systems under demand uncertainty,” *Transportmetrica: Transportation Science*, vol. 15, no. 2, pp. 698–721, 2019.
 - [13] F. Dandl, M. Hyland, K. Bogenberger, and H. S. Mahmassani, “Evaluating the impact of spatio-temporal demand forecast aggregation on the operational performance of shared autonomous mobility fleets,” *Transportation*, vol. 46, no. 6, pp. 1975–1996, 2019.
 - [14] P. Segui-Gasco, H. Ballis, V. Parisi, D. G. Kelsall, R. J. North, and D. Busquets, “Simulating a rich ride-share mobility service using agent-based models,” *Transportation*, vol. 46, no. 6, pp. 2041–2062, 2019.
 - [15] J. Narayan, O. Cats, N. V. Oort, and S. P. Hoogendoorn, “Fleet size determination for a mixed private and pooled on-demand system with elastic demand,” *Transportmetrica: Transportation Science*, vol. 17, no. 4, pp. 897–920, 2021.
 - [16] B. h. Nahmias-Biran, J. B. Oke, N. Kumar, C. Lima Azevedo, M. Ben-Akiva, and M. Ben-Akiva, “Evaluating the impacts of shared automated mobility on-demand services: an activity-based accessibility approach,” *Transportation*, vol. 48, no. 4, pp. 1613–1638, 2020.
 - [17] K. Winter, O. Cats, K. Martens, and B. van Arem, “Relocating shared automated vehicles under parking constraints: assessing the impact of different strategies for on-street parking,” *Transportation*, vol. 48, no. 4, pp. 1931–1965, 2020.
 - [18] J. Mora, S. Samaranyake, A. Wallar, E. Frazzoli, and D. Rus, “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.
 - [19] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, “Addressing the minimum fleet problem in on-demand urban mobility,” *Nature*, vol. 557, no. 7706, pp. 534–538, 2018.
 - [20] Z. Liu, T. Miwa, W. Zeng, M. G. H. Bell, and T. Morikawa, “Shared autonomous taxi system and utilization of collected travel-time information,” *Journal of Advanced Transportation*, vol. 2018, pp. 1–13, 2018.
 - [21] X. Liang, G. H. d. A. Correia, K. An, and B. van Arem, “Automated taxis’ dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times,” *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 260–281, 2020.
 - [22] G. Alfian, J. Rhee, Y. S. Kang, and B. Yoon, “Performance comparison of reservation based and instant access one-way car sharing service through discrete event simulation,” *Sustainability*, vol. 7, no. 9, pp. 12465–12489, 2015.
 - [23] L. Hu and Y. Liu, “Joint design of parking capacities and fleet size for one-way station-based carsharing systems with road congestion constraints,” *Transportation Research Part B: Methodological*, vol. 93, pp. 268–299, 2016.
 - [24] B. Boyacı, K. G. Zografos, and N. Geroliminis, “An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations,” *Transportation Research Part B: Methodological*, vol. 95, pp. 214–237, 2017.
 - [25] L. Li and M. E. H. Petering, “Discrete event simulation analysis of a reservation-based, one-way car-sharing system,” *Journal of Simulation*, vol. 12, no. 1, pp. 1–22, 2018.
 - [26] P. Ströhle, C. M. Flath, and J. Gärttner, “Leveraging customer flexibility for car-sharing fleet optimization,” *Transportation Science*, vol. 53, no. 1, pp. 42–61, 2019.
 - [27] M. Roca-Riu and M. Menendez, “The potential of flexible reservations in a car sharing system with an auction scheme,” *Ieee Access*, vol. 7, pp. 154627–154639, 2019.
 - [28] G. Molnar and G. H. d. A. Correia, “Long-term vehicle reservations in one-way free-floating carsharing systems: a variable quality of service model,” *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 298–322, 2019.
 - [29] X. Dou, Q. Meng, and K. Liu, “Customized bus service design for uncertain commuting travel demand,” *Transportmetrica: Transportation Science*, vol. 17, no. 4, pp. 1405–1430, 2021.
 - [30] H. Wang, R. Cheu, and D.-H. Lee, “Intelligent taxi dispatch system for advance reservations,” *Journal of Public Transportation*, vol. 17, no. 3, pp. 115–128, 2014.
 - [31] X. Liang, G. H. D. A. Correia, and B. van Arem, “Optimizing the service area and trip selection of an electric automated taxi system used for the last mile of train trips,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 93, pp. 115–129, 2016.
 - [32] V. Pimenta, A. Quilliot, H. Toussaint, and D. Vigo, “Models and algorithms for reliability-oriented Dial-a-Ride with autonomous electric vehicles,” *European Journal of Operational Research*, vol. 257, no. 2, pp. 601–613, 2017.
 - [33] S. Su, E. Chaniotakis, S. Narayanan, H. Jiang, and C. Antoniou, “Clustered tabu search optimization for reservation-based shared autonomous vehicles,” *Transportation Letters*, pp. 1–5, 2020.
 - [34] J. Liu, P. Mirchandani, and X. Zhou, “Integrated vehicle assignment and routing for system-optimal shared mobility planning with endogenous road congestion,” *Transportation Research Part C: Emerging Technologies*, vol. 117, Article ID 102675, 2020.
 - [35] J. Ma, X. Li, F. Zhou, and W. Hao, “Designing optimal autonomous vehicle sharing and reservation systems: a linear programming approach,” *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 124–141, 2017.
 - [36] G. Homem, D. A. Correia, and B. V. Arem, “Data from: trips and network of the case-study city of Delft,” 2016.