Electronic Research Archive

http://www.aimspress.com/journal/ERA

ERA, 32(1): 523–550.
DOI: 10.3934/era.2024026
Received: 29 November 2023
Revised: 21 December 2023
Accepted: 26 December 2023
Published: 03 January 2024

Research article

Dynamic coordinated strategy for parking guidance in a mixed driving parking lot involving human-driven and autonomous vehicles

Zhiyuan Wang[1], Chu Zhang[1], Shaopei Xue[1], Yinjie Luo[2], Jun Chen[1,3,*], Wei Wang[1] and Xingchen Yan[4]

[1] School of Transportation, Southeast University, Jiangning District, Nanjing 211189, China
[2] Division of Engineering, New York University Abu Dhabi, Saadiyat Island, Abu Dhabi 129188, United Arab Emirates
[3] Key Laboratory of Transport Industry of Comprehensive Transportation Theory (Nanjing Modern Multimodal Transportation Laboratory), Ministry of Transport of the PRC, No. 56 Baoshansi Road, Jiangning District, Nanjing 211100, China
[4] College of Automobile and Traffic Engineering, Nanjing Forestry University, No. 159 Longpan Road, Nanjing 210037, China

* **Correspondence:** Email: chenjun@seu.edu.cn.

Abstract: The advent of autonomous vehicles (AVs) poses challenges to parking guidance in mixed driving scenarios involving human-driven vehicles (HVs) and AVs. This study introduced a dynamic and coordinated strategy (DCS) to optimize parking space allocation and path guidance within a mixed driving parking lot, aiming to enhance parking-cruising efficiency. DCS considers the distinctive characteristics of HVs and AVs and dynamically formulates parking guiding schemes based on real-time conditions. The strategy encompasses four main steps: Triggering scheme formulation, identifying preoccupied parking spaces, updating the parking lot traffic network and optimizing the vehicle-path-space matching scheme. A programming model was established to minimize the total remaining cruising time, and iterative optimization was conducted with vehicle loading test based on timing. To elevate computational efficiency, the concept of parking-cruising path tree (PCPT) and its updating method were introduced based on the dynamic shortest path tree algorithm. Comparative analysis of cases and simulations demonstrated the efficacy of DCS in mitigating parking-cruising duration of different types of vehicles and minimizing forced delays arising from lane blocking. Notably, the optimization effect is particularly significant for vehicles with extended cruising durations or in parking lots with low AV penetration rates and high saturation, with an achievable optimization

rate reaching up to 18%. This study addressed challenges related to drivers' noncompliance with guidance and lane blocking, thereby improving overall operational efficiency in mixed driving parking lots.

Keywords: mixed driving; parking guidance; dynamic coordinated strategy (DCS); iterative optimization; efficiency; dynamic shortest-path tree

1. Introduction

Parking cruising refers to the process of searching for an available parking space after a traveler has arrived at their destination or its vicinity by vehicle. Extensive research has been conducted by numerous scholars on vehicular parking cruising in urban roadways across various regions and times, with durations typically ranging between 0.5 and 16 minutes. The proportion of cruising traffic can reach up to 74% [1–3]. However, there have been few studies focused on vehicular cruising within parking facilities. Based on our investigation during peak hours at a large-scale parking lot (the same as the one introduced in Section 5.2), it was found that on average, each entering vehicle cruised 49% more distance than the shortest path to reach its final parking space. Some vehicles even needed to travel over 600 meters extra, despite already being inside the parking facility.

Therefore, parking cruising is often assisted by parking guidance and information systems (PGISs) in numerous parking lots [4,5]. These systems serve as bridges between parking lots and vehicles, offering drivers long-range information to assist them in navigating suitable paths toward available parking spaces.

The development of autonomous vehicles (AVs) may bring transformative changes to road traffic, including parking. With the autonomous valet parking function, high-level AVs can independently complete parking-cruising without drivers [6,7]. This change transforms the role of PGIS from advisor to commander.

As it is unlikely that human-driven vehicles (HVs) will disappear entirely in the coming decades, the mixed driving scenario of HVs and AVs is a research topic that combines foresight and practicality [8,9]. Traditional PGISs in parking lots have limited ability to detect real-time vehicle movements, providing only one-way information about available spaces to vehicles [10]. However, in a new mixed driving scenario with both HVs and AVs, the operation and management of parking lots face new situations [11–13]:

1) For HVs, human drivers still can independently choose their paths and parking spaces based on the guidance provided by PGIS. It can use advanced sensing devices to grasp real-time information about HVs and provide conditional data for the optimization of all vehicles' operations.

2) For AVs, PGIS will command AV to operate accordingly through the mutual transmission of information, without the autonomous consciousness of the driver and disobedience to the command.

In the new mixed driving scenario, PGIS should allocate spaces and guide paths based on real-time detection of vehicles, parking spaces and lanes. This enables comprehensive optimization, providing direct decision support for HV drivers and AVs and enhancing parking-cruising efficiency.

The essence of parking guidance is a dynamic resource assignment (DRA) problem [14]. However, existing research and testing systems often treat real-time parking guidance as an independent and one-time task. In other words, their approach involves allocating parking spaces or planning paths for vehicles upon their entry into the parking lot (Figure 1(a)) [15,16]. This approach, called single formulating strategy (SFS), is feasible assuming full compliance with guidance and a static status of the parking facility throughout the parking-cruising process. However, SFS has limitations due to potential guidance violations by drivers [17] and dynamic updates of multiple elements within the parking lot, which may render the original scheme invalid or suboptimal.

As shown in Figure 1(b), HV B enters the parking lot after AV A. However, Vehicle B deviates from its scheduled path manually, indicating a high likelihood of ultimately occupying Parking Space 3. If Vehicle A continues cruising to Space 3 according to the original scheme, it not only forces Vehicle B to detour further in search of an alternative available space, but also blocks B's pathway through the lane near Space 3, significantly increasing the total cruising duration. To avoid such problems, dynamically reallocating Vehicle A to Space 1 and rendering Space 3 available for Vehicle B would present a better solution. Moreover, SFS is unable to promptly utilize newly available spaces resulting from vehicle departures for vehicles that have already started cruising in the parking lot.

In addition, coordination among multiple vehicles is crucial in dynamically changing scenarios. SFS assumes that the guiding scheme for vehicles already in the parking lot remains unchangeable when planning for a newly arriving vehicle, potentially leading to missing the globally optimal solution. As illustrated in Figure 1(c), the leading Vehicle A needs to block the lane to reverse into Parking Space 1. If Vehicle B is allocated to Space 2 by the SFS-based PGIS, it will encounter obstruction while passing through Space 1, resulting in an obvious delay. Even with the consideration of lane blocking, the PGIS employing SFS can only allocate Vehicle B to Space 3. However, if PGIS can coordinate among multiple vehicles, Vehicle A could be reassigned to Space 2 while Space 1 could be reserved for Vehicle B, achieving the globally optimal total parking-cruising duration of both vehicles.

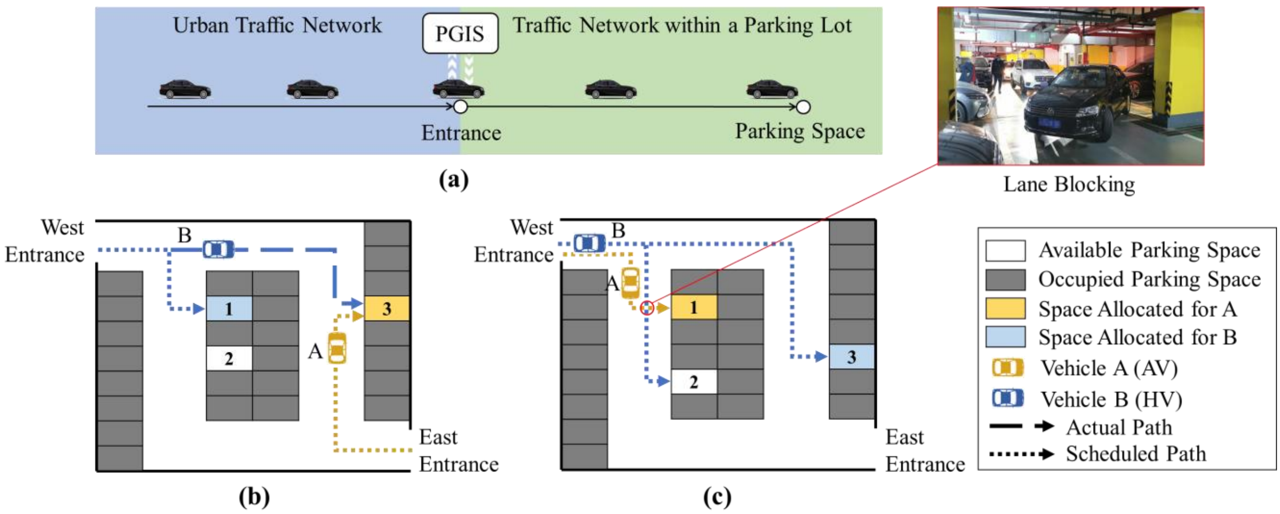


Figure 1. Principle and two scenarios illustrating the shortcomings of SFS.

Therefore, in a mixed driving parking lot, it is necessary to focus on the dynamic and coordinated adjustment of parking space allocation and path guidance schemes for all cruising vehicles that have not yet parked inside the parking lot (Figure 2). This study aims to propose a dynamic and coordinated strategy (DCS) for parking space allocation and path guidance in a parking lot under the possible future scenario of mixed driving. Based on real-time sensing and estimation of the status of multiple entities, including vehicles, lanes and parking spaces, DCS will formulate, optimize and then transmit the parking guidance scheme to all cruising vehicles. It will reduce the overall parking-cruising duration through dynamic coordinated parking guidance, eliminating adverse effects and approaching the globally optimal solution.

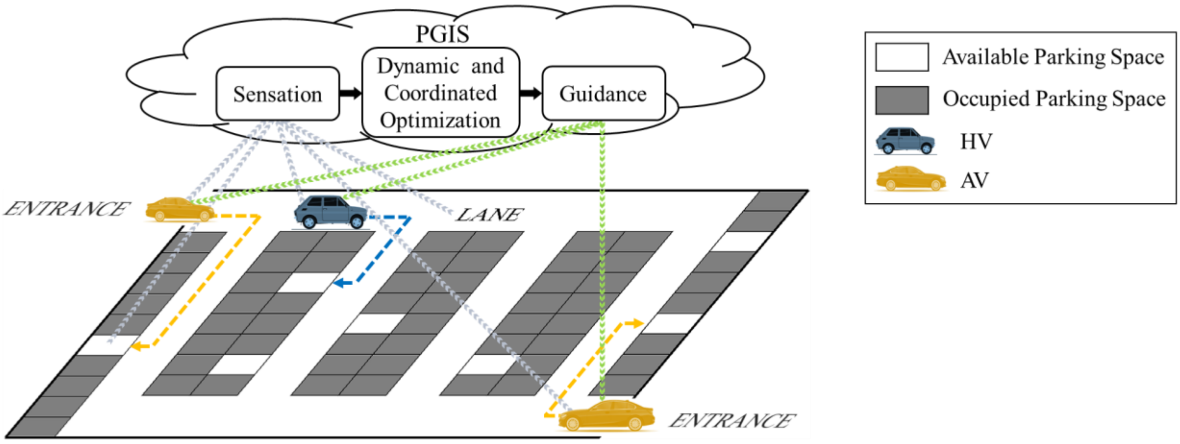


Figure 2. Working framework for PGIS in a mixed driving parking lot.

The remaining sections of this paper are organized as follows: Research on mixed driving and parking guidance strategies is reviewed in Section 2. Section 3 introduces the preparatory work revealing fundamental statements and network construction. Section 4 proposes the dynamic coordinated parking space allocation and path guidance strategy for the mixed driving parking lot. In Section 5, two case studies are conducted with computer simulation to validate the effectiveness of the proposed strategy under various conditions. Finally, conclusions are presented in Section 6.

2. Literature review

In this section, two closely related research streams are reviewed as follows.

2.1. Traffic management and control with mixed driving

Traffic management and control with mixed driving involving HVs and AVs is a realistic and pressing topic to study [18,19]. The penetration of AVs into the traffic system can be divided into four stages: Pure HVs, HV-dominant, AV-dominant, and pure AVs [8]. In the second and third stages, the presence of different types of vehicles leads to unknown and complex interactions posing significant challenges [19,20]. Currently, apart from research on the physical automation of vehicles, investigations on traffic management and control with mixed driving mainly focus on microlevel aspects such as platooning [21–23], intersection [24,25] and lane management [11,26,27], as well as macro-level aspects such as traffic network allocation [28–30]. Few studies specifically address mixed driving within parking lots. However, analysis approaches adopted in existing literature can be valuable references.

An important issue arises from the characteristics of HVs and AVs, specifically regarding the subjective consciousness and agency of vehicle control and the ability to comply with external information. Roughgarden and Tardos [31] highlighted the existence and severe consequences of selfish routing by HV drivers in terms of their chosen driving routes. In one case, it could result in a total delay reaching up to 4/3 times the minimum delay. This highlights the disparity between user equilibrium and system optimality, indicating the greatest distinction in decision-making between HVs and AVs. Human drivers may selfishly make choices based on their styles and preferences [32], while

AVs can adhere to system-optimized instructions.

In light of this, relevant research on mixed driving has considered this. Khattak et al. [26] identified overcoming drivers' noncompliance as a key task for their lane management system. Inspired by Klein's work [33], Guo et al. [30] categorized travelers into two groups for mixed driving path allocation. In their theory, HVs do not sacrifice their interests for the sake of system but make choices with bounded rationality to satisfy their preferences. In contrast, AVs can comply with the guidance from the management system to achieve system optimality. Shi et al. [21] acknowledged that individual HVs were uncontrollable, but the collective behavior of HVs exhibits patterns and could be controlled. Therefore, they utilized the aggregation of HVs in the mixed driving vehicles with AVs to mitigate their randomness in the macroscopic characteristics.

Compared to urban road networks, intersections or public road segments discussed in the aforementioned studies, the dynamic traffic within a parking lot is relatively small. Additionally, in this study, the dynamic coordinated parking space allocation and path guidance schemes are tailored for each individual vehicle in real-time scenarios. In this context, although the group characteristics of HVs and drivers exhibit macroscopic patterns and statistical significance, in the real-time scenarios focused on in this study, probabilities manifest themselves as either not occurring at all or occurring with certainty (zero or one) in a small sample within each time slice. Hence, the macroscopic patterns and conclusions regarding driving behavior are challenging to directly apply to our dynamic strategy design. Specifically, even minor factors within a parking lot can lead drivers to make different choices, such as varying levels of driving proficiency, familiarity with the parking lot, pedestrians or temporary obstacles alongside the lanes. One imaginable scenario is that for a highly desired hot spot parking space, a slight miscalculation by one driver, such as pressing the accelerator a bit harder, could result in the vehicle quickly passing by that space and parking into another less popular space, ultimately causing a breakdown of macroscopic patterns within this particular scenario. Conversely, for the overall parking lot, the stochastic and unpredictable performance exhibited by individual vehicles will have a significant impact on other surrounding vehicles.

Furthermore, given that AVs cruising in the parking lot do not have occupants onboard, this study will prioritize HVs. This approach also represents the customary practice in situations where HVs and AVs encounter conflicts [24]. When human drivers fail to comply with PGIS instructions, their behavior will be treated as a factor for dynamic coordinated decision for other cruising vehicles [25], aiming to seek optimal solutions within a controllable range.

2.2. Parking guidance and optimal operation within a parking lot

Numerous scholars have conducted research on parking guidance and optimal operation methods within a parking lot, which can be broadly categorized based on the objectives into two types.

1) When optimizing for individual parkers, the objective is to save time or costs for each individual parker within the parking lot [34]. Some studies focused on cruising duration as a critical factor and used algorithms like Dijkstra, Floyd, A*, or other improved algorithms for parking space selection and path planning [35–37]. Ata et al. [15] adopted a smart indoor parking system, searching for an available and suitable parking space for the first vehicle to arrive at the entrance in each allocation. Yu [16] and Wang [38] et al. respectively constructed time-varying road network models in HV and AV scenarios, dynamically adjusting edges' weights for shortest path calculations when one vehicle arrives.

In general, these studies focus on providing individual vehicles with optimal available parking spaces upon entry into the parking lot. PGIS functions as a personalized assistant for individual parkers, delivering customized services to each of them.

2) When optimizing the entire parking lot or multiple parkers, the objective is to improve the overall operational efficiency of all vehicles. Gai et al. [39] established a two-layer optimization model considering user and system perspectives, resulting in shorter parking-cruising durations. Chen et al. [40] developed a model for parking space allocation intending to eliminate interweaving and conflicts caused by lane blocking of vehicles entering and exiting parking spaces. Their model benefits both parkers and passer-by drivers by reducing delays.

In such studies, PGIS acts as a manager by coordinating and scheduling multiple vehicles, surpassing the individual parking users.

In summary, various methods have been employed to optimize vehicle parking, with a common objective of reducing individual or overall cruising duration. This objective reflects the convenience for parkers and the operational efficiency of parking facilities, and can further extend to considerations of energy consumption, pollution emissions and other related aspects. In our study, the objective is to minimize the remaining parking cruising duration for the entire field of vehicles, considering mixed driving characteristics and the mutual influence of vehicles on lanes. The strategy for dynamic coordinated parking space allocation and path guidance is designed accordingly.

3. Preparatory work

Before introducing the core strategy, it is essential to explain fundamental settings as preparatory work.

3.1. Fundamental statements and assumptions

Currently, mixed driving scenarios are in a limited testing phase, with many unknown or uncertain details. To standardize and clarify conditions for proposing the strategy subsequently, the following statements and assumptions are provided.

1) HVs receive guidance information provided by PGIS throughout the parking cruise but may deviate from the instructions. AVs strictly follow the PGIS upon entering the parking lot [11–13].

2) In the context of limited lane and parking space, vehicles using Ackermann steering (which is employed by the vast majority of modern cars) typically opt for reverse parking into parking spaces, including both HVs and AVs [41–44]. The duration for HVs to block the lane before reversing into a parking space is an indeterminate positive value influenced by multiple factors [45]. The controllers of AVs typically prescribe a stable reference longitudinal speed for cruising contingent upon specific scenarios [46]. For the sake of simplification in this study, this duration is established as a constant value tr_{AV} .

3) As this is an off-street parking lot, traffic unrelated to parking or vehicles exceeding the capacity will not enter. AV occupants disembark before entering the parking lot [46,47]. HV occupants do not interfere with other vehicles' cruising after disembarking. Lanes are unobstructed, except when a vehicle reversing into a parking space blocks the lane [16].

4) As discussed in Section 2.1, in the event of a conflict between AVs and HVs, HVs have priority [24,25].

3.2. Construction of traffic network within a parking lot

The traffic network within a parking lot is constructed based on the spatial structure, with nodes representing the endpoints of lanes and the connection points of each parking space to the lanes, and edges representing the road segments between adjacent nodes, as shown in Figure 3.

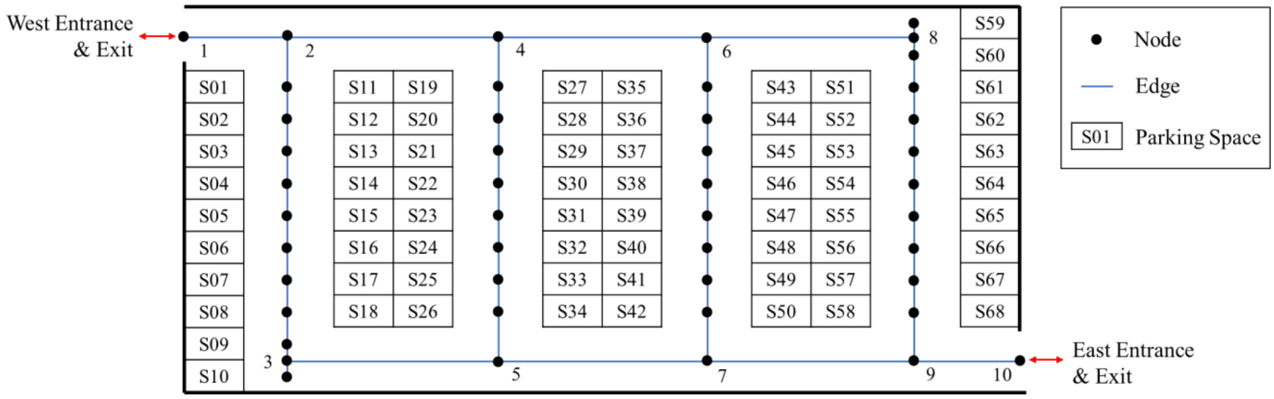


Figure 3. Sample of traffic network in a parking lot.

The nodes of parking spaces are labeled with S followed by a number. In cases of symmetrically arranged spaces on both sides of lanes, some nodes are shared by two spaces. For instance, in Figure 3, parking spaces S01 and S11 share one node on the same road segment. This node can be referred to as either S01 or S11. For constructing this traffic network, the driving velocities of all vehicles are assumed to be v. The weight of each edge is defined as the ratio of the corresponding lane length to v (i.e., the duration required for a vehicle to pass through the corresponding road segment).

4. Description of strategy and algorithm

A feasible process for the parking space allocation and path guidance strategy is proposed in this section. To timely improve parking-cruising efficiency, the scheme of parking space allocation and path guidance will be dynamically updated to meet the latest status. When there are parking-cruising vehicles that have not yet parked, there is a need for parking space allocation and path guidance and the strategy is in an operational state. The process of DCS in the operational state is described as four main steps and several sub-steps (Figure 4).

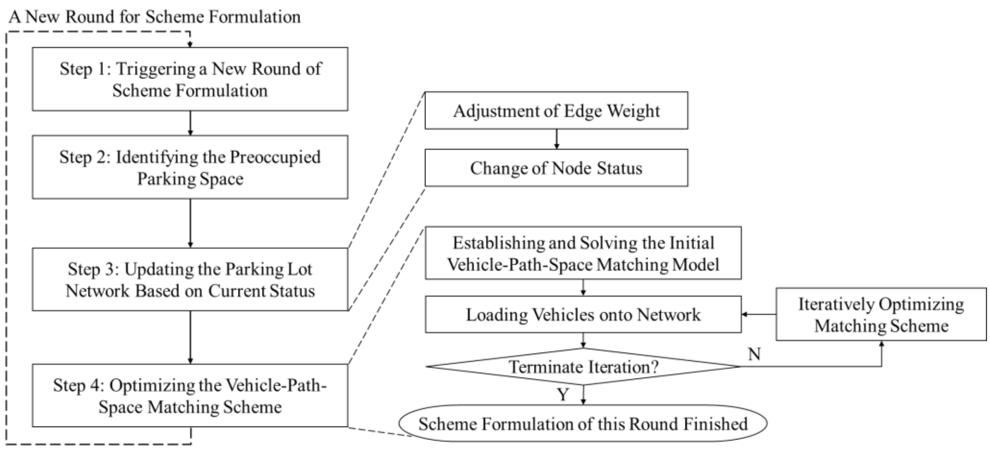


Figure 4. Schematic process of DCS for scheme formulation.

When the necessity arises to update the parking guidance scheme, Step 1 initiates a new round of scheme formulation through conditional judgment. The subsequent three steps are then executed in sequence to derive the current optimal scheme at that moment. Step 2 focuses on the priority of HV parkers, identifying preoccupied parking spaces. Step 3 digitizes the status of various elements within the parking lot, aligning them to the traffic network. Finally, Step 4 generates the guidance scheme for all cruising vehicles through its internal iterative optimization process. From the perspective of time progression, these steps are iteratively performed to ensure that the system continually adapts to the dynamic environment and minimizes any associated losses caused by real-time changes.

Some symbols that will repeatedly appear in later sections are explained in Table 1.

Table 1. Symbol explanation.

Symbol	Meaning
f_{it}	Node where vehicle i happens to be located or the nearest network node running ahead at time t, also denoted as $f(i,t)$
$SPT_{f(i,t),0}$	Initial network SPT with node $f(i,t)$ as its root
$\Delta(G_1,G_2)$	Table of weight-adjusted edges and their change values in network G_2 relative to G_1
$\boldsymbol{\delta}^{+}(G_1,G_2)$	Set of edges with increasing weight in $\Delta(G_1,G_2)$
$\boldsymbol{\delta}^{-}(G_1,G_2)$	Set of edges with decreasing weight in $\Delta(G_1,G_2)$
$SPT_{f(i,t),t}$	Network SPT with node $f(i,t)$ as its root at time t
$PCPT_{f(i,t),t}$	PCPT with node $f(i,t)$ as its real root at time t
$PCPT'_{f(i,t),r}$	PCPT with node $f(i,t)$ as its real root after r iterations at time t
$\boldsymbol{N}(x)$	Set of nodes in network or tree x
Z_r	Overall objective function, representing the total remaining parking-cruising duration of all vehicles in the parking lot
x_{ij}	Binary decision variable, representing whether match vehicle i with parking space j
$\boldsymbol{C}_t$	Set of parking-cruising vehicles at time t in the parking lot
$\boldsymbol{S}_t$	Set of vacant parking spaces at time t in the parking lot
G_t	Traffic network within parking lot at time t, where G_0 is the initial network
G'_r	Traffic network within parking lot after r iterations, where G'_0 is obtained from Step 3
d_{ij}^{r}	The shortest duration from vehicle i to parking space j in network G'_r according to PCPT
R_i	Estimated remaining parking-cruising duration for vehicle i to reach its allocated parking space
i_r	The vehicle that experiences the longest increase in parking-cruising duration due to vehicle loading after r iterations, where $\{i_0\}$ is set to an empty set
$\boldsymbol{A}_r$	Set of vehicles for whose corresponding R_i is no more than R_{i_r}
$\boldsymbol{J}_r$	Set of parking spaces j allocated to the vehicles whose corresponding R_i is greater than R_{i_r} in the previous allocation
Y_h^r	Objective function for optimizing a single vehicle through iterative process, representing the shortest remaining parking-cruising duration for vehicle h after r iterations
$tr_{AV,j}$	AV's reversing duration of blocking lane before entering parking space j

4.1. *Triggering a new round of scheme formulation*

Determining the beginning of a new round of scheme formulation marks Step 1 in the overall strategy. After the last round of parking space allocation and path guidance scheme is formulated, if there are no critical events, there is no need to update the scheme. Critical events include:

1) A vehicle deviating from the scheduled path or entering an unscheduled parking space;

2) A parked vehicle leaving its parking space;

3) A vehicle entering the parking lot.

When PGIS detects such circumstances, it indicates that the previous scheme is likely no longer optimal or effective, and a new round of scheme formulation should be triggered.

4.2. Identifying the preoccupied parking space

Since the choice of parking spaces for HVs is not under control, speculating on the spaces that HV drivers may choose early on can improve the robustness of subsequent planning and reduce the risk of scheme failure. Therefore, if there are cruising HVs in the parking lot, PGIS checks whether they meet the conditions for preoccupying a parking space.

The method for Step 2, identifying preoccupied parking spaces, is as follows: Check each HV and calculate if it is approaching or moving away from each available parking space based on its current location and travel direction. If only one parking space is closer to the HV, the HV will likely occupy that space, marking it as preoccupied. Parking spaces marked as preoccupied are considered unavailable in subsequent optimization of scheme in this round, and the HV matched with a preoccupied space does not require allocation or guidance path.

4.3. Updating the parking lot network based on current status

Traffic networks should be able to accurately abstract and digitize real time traffic status. Based on the latest status within the parking lot, combined with preoccupied spaces, the edge weight and available parking space information in the parking lot traffic network at the current time is updated in Step 3. In addition, during the optimization process in Step 4, the network also needs to be updated based on the results of each iteration to prepare for the next iteration. Therefore, updating the traffic network within the parking lot needs to be repeatedly applied throughout the entire strategy.

The basic idea of updating the traffic network is to adjust edge weights and node statuses based on the given status within the parking lot. This status may be based on the current status (in Step 3) or calculated during the optimization iteration process (in Step 4). The input information for the update includes:

1) Position and remaining duration of each vehicle blocking the lane and reversing into its parking space.

2) Availability of each parking space.

3) Preoccupied parking space information of HV.

Using the above data, some attributes of edges and nodes in the traffic network within the parking lot will change.

4.3.1. Adjustment of edge weight

Since a vehicle reversing into a parking space will block the lane and obstruct the passage of other vehicles, the weight of the network edge corresponding to the road segment should be modified based on the estimated remaining duration of the lane blocking. According to Fundamental Assumption 2, (a) For an AV, the duration it blocks the lane while reversing is constant tr_{AV} , so the remaining duration can be calculated based on the start time. (b) The duration that an HV blocks the lane is uncertain. To prioritize HVs (Fundamental Assumption 4), the remaining duration is set as a large positive number, represented as Big M. (c) An HV preoccupying a parking space is assumed to be reversing into the space with a remaining duration of Big M, even if it has not reached the space yet.

Suppose the remaining duration that vehicle i reversing into parking space j is denoted as tr_{ij}, and the calculation formula for its value at time t is as follows.

$$tr_{ij} = \begin{cases} tr_{AV,j} - (st_{ij} - t) & \text{if } i \text{ is a reversing AV} \\ \text{Big M} & \text{if } i \text{ is an HV reversing into or preoccupying a space} \end{cases} \tag{1}$$

where st_{ij} represents the moment when vehicle i starts reversing into parking space j.

After calculating tr_{ij} using the above method, the weight of either edge adjacent to node Sj is set to the remaining time tr_{ij}.

4.3.2. Change of node status

This status refers to the availability of parking spaces connected to the node, rather than that in the structure of network. It means whether this node can serve as the destination of the vehicle's path and the connectivity of the network will not change due to the change in node status. Specifically, the status of parking space can be occupied, preoccupied, being entered, and available, while the status of a node can be available or unavailable. Therefore, the following criteria need to be considered:

Criterion 1. If all parking spaces connected to a node are either occupied or preoccupied, the node is marked as unavailable.

Criterion 2. Since a vehicle reversing into a parking space will block other vehicles from accessing other available spaces connected to the same node, if at least one space connected to a node is being entered, the node is marked as unavailable.

Nodes that do not meet the above criteria are marked as available.

In addition, a node will not be marked as unavailable to prevent a vehicle from being rejected to enter the parking lot. If a node is connected to more than one parking space and there is only one available space connected to the node, the node will be marked as available regardless of Criterion 2.

In conclusion, Step 3 transforms the status of on-site elements including vehicles, lanes, and parking spaces into the attributes of the traffic network. This transformation enables the convenient digitization of inputs for subsequent establishment and solving of the optimized matching model.

4.4. Optimizing vehicle-path-space matching scheme

Combining the latest traffic network and location of all parking-cruising vehicles, the parking-cruising vehicles, driving paths and available parking spaces (vehicle-path-space) matching for this round is performed in Step 4. This is the most key part of DCS and will also be expounded in detail subsequently with binary programming model and iterative optimization. After this step, an optimal parking space allocation and path guidance scheme can be obtained, which will be fed back to all cruising vehicles by PGIS.

The ultimate objective of this programming model is to allocate parking spaces to cruising vehicles and plan their paths to minimize the total remaining cruising duration of current cruising vehicles. The approach is similar to the assignment problem in operations research. In classical assignment problems, the cost matrix is fixed. However, in this study, the weight of traffic network edges changes dynamically when vehicles are guided to drive on them, which affects each other and changes the cost matrix. Therefore, this study establishes a binary programming model based on the assignment model and uses an iterative approach to search for the optimal solution. The workflow of Step 4 is depicted in the lower right section of Figure 4.

4.4.1. Establishing and solving the initial vehicle-path-space matching model

The purpose of the first sub-step is to obtain the initial matching results of cruising vehicles, paths, and available spaces. Assuming the current time is t and setting $r=0$, the programming model can be represented as:

1) Optimization objective

Since the parking-cruising duration has already been consumed by vehicles and cannot be changed, what PGIS can do is minimizing the total remaining parking-cruising duration. We use the binary decision variable x_{ij} to reflect whether to match the vehicle i to the parking space j. The cruising duration d_{ij}^{r} for vehicle i to space j can be obtained according to the parking lot traffic network. The formula can be expressed as:

$$\min\ Z_0=\sum_{i\in \boldsymbol{C}_t}\sum_{j\in \boldsymbol{S}_t} x_{ij} d_{ij}^{r} \quad . \tag{2}$$

2) Constraints

The goal of the scheme is to match parking cruising vehicles, driving paths, and available parking spaces (Constraints a and b). Each vehicle that is cruising must match an available parking space (Constraint c), and each space can only be allocated to at most one vehicle (Constraint d). Finally, there is a constraint on the values of binary decision variables (Constraint e).

$$\text{s.t.}\begin{cases} \text{a. } i\in \boldsymbol{C}_t \\ \text{b. } j\in \boldsymbol{S}_t \\ \text{c. } \sum_{j\in \boldsymbol{S}_t} x_{ij}=1,\ \forall i\in \boldsymbol{C}_t \\ \text{d. } \sum_{i\in \boldsymbol{C}_t} x_{ij}\le 1,\ \forall j\in \boldsymbol{S}_t \\ \text{e. } x_{ij}\in\{0,\ 1\} \end{cases} \tag{3}$$

Combining Eqs (2) and (3), the initial matching results can be obtained by solving the model. It is possible to have multiple solutions in which the objective Z_0 achieves the same optimal value. In such a case, the solution that minimizes the changes in the vehicle-path-space matching results compared to the previous scheme is considered as the optimal solution.

At this juncture, PGIS is capable of formulating a matching scheme for parking guidance. However, this scheme may not necessarily represent the optimum solution. Therefore, subsequent sub-steps are required to perform optimization and adjustments.

4.4.2. Loading vehicles onto network

Due to the static nature of the above programming model, dynamic interaction between cruising vehicles has not been directly considered. To avoid issues such as lane blocking, it is necessary to load the vehicle on the network for testing. Assuming that each vehicle travels according to the shortest path corresponding to the matching result x obtained from previous solution based on G_t, the remaining cruising duration R_i for each vehicle to reach its parking space is estimated with d_{ij}^{0}. The i corresponding to the sorted R_i in ascending order are loaded onto their respective paths, and the following operations are performed:

1) Set $r = r + 1$ (4)

2) Generate the network G'_r after loading the vehicles according to the update method for the traffic network.

3) Based on network G'_r, calculate $d_{ij}^r \left(i \in \boldsymbol{C_t},\ j \in \boldsymbol{S_t} \right)$ (5)

4) Calculate $Z_r = \sum_{i \in \boldsymbol{C_t}} \sum_{j \in \boldsymbol{S_t}} x_{ij} d_{ij}^r$ (6)

When the iteration time r is not less than two, it is necessary to establish a termination condition after the vehicle loading test, which is vital to avoid oscillation or excessive time expenditure in pursuing negligible effects. If $r \geq 2$ and either of the following two conditions are satisfied, the iteration will terminate, resulting in the final matching result and guidance scheme at current time. Otherwise, it becomes essential to execute a subsequent iteration of the matching scheme optimization in accordance with the next sub-step, aimed at further augmenting the efficacy of scheme formulation.

Condition 1. The iteration time r has reached the preset upper limit;

Condition 2. The decrease rate (DR) in the overall objective function value is nonnegative and less than the preset threshold. The formula of DR is:

$$DR = \frac{Z_{r-1} - Z_r}{Z_{r-1}} \tag{7}$$

If the iteration is judged to terminate, the current x_{ij} will be used as the matching result between vehicles and parking spaces. The path corresponding to d_{ij}^r in network G'_r will be used as vehicle's driving path to complete the matching of vehicles-path-space at the current time. Since Step 4 is the last in the process of formulating the vehicle-parking space allocation and path guidance plan for a round, the above matching result serves as the basis for PGIS to guide or instruct vehicles at time t.

4.4.3. Iteratively optimizing matching scheme

If the iteration has not been terminated previously, it is necessary to perform iterative optimization for the matching scheme that exhibits poor performance after vehicle loading, aiming to obtain improved solutions. Based on the previous results, the following operations are performed to identify vehicles that require rematching in iteration:

1) Calculate $\left(i_r, j_r \right) = argmax_{i,j} \left[x_{ij} \left(d_{ij}^r - d_{ij}^0 \right) \right] \quad \forall i \in \boldsymbol{C_t} - \left\{ i_{r-1} \right\},\ j \in \boldsymbol{S_t}$. (8)

2) Construct the set $\boldsymbol{A_r}$ consisting of vehicles i whose R_i is no more than $R_{i(r)}$.

3) Construct the set $\boldsymbol{J_r}$ consisting of parking spaces j assigned to the vehicles i whose R_i is greater than $R_{i(r)}$ in the previous allocation.

4) Set $x_{i,j} = 0 \quad \forall i \in \boldsymbol{A_r},\ j \in \boldsymbol{S_t}$. (9)

A programming model is constructed to rematch parking spaces with vehicles for iterative optimization. This model needs to be executed for each vehicle with an increased cruising duration due to vehicle loading. Similar to Eq (3), the optimization objective is set to minimize the remaining parking-cruising duration. The constraints are set to establish a one-to-one match between the target vehicle and parking spaces that have not been allocated to other vehicles. Elements from set $\boldsymbol{A_r}$ are selected in descending order and are substituted into h, respectively.

$$\min\ Y_h^r = \sum_{j \in \boldsymbol{S_t}} x_{hj} d_{hj}^r \tag{10}$$

$$\text{s.t.}\begin{cases} \text{a. } j \in \boldsymbol{S_t} - \boldsymbol{J_r} \\ \text{b. } \sum_{i \in \boldsymbol{C}_t} x_{ij} \le 1,\ \forall j \in \boldsymbol{S_t} \\ \text{c. } \sum_{j \in \boldsymbol{S_t}} x_{hj} = 1 \\ \text{d. } x_{hj} \in \{0,\ 1\} \end{cases} \tag{11}$$

It is worth noting that the quantity of elements in $\boldsymbol{A_r}$ equals the times the model is established and solved in the current iteration. Although the model may be computed multiple times, all calculations are part of the r-th iteration. Upon completion of each model solution, the parking space allocated to vehicle h is added to $\boldsymbol{J_r}$.

Upon completing a round of iterative solution, DCS will once again engage in the vehicle loading test and iteration termination judgment as described in the preceding section. This is to determine whether to continue with further iterations or to adopt the current iteration results. If the latter is the case, it signifies the completion of Step 4, whereby real-time parking space allocation and path guidance schemes are formulated and can be transmitted to cruising vehicles through PGIS.

4.5. *Efficient calculation method for shortest path based on the parking-cruising path tree*

Within the process of DCS, the calculation of the shortest path from cruising vehicles to available parking spaces plays a pivotal role in both the identification of preoccupying parking spaces in Step 2, and the optimization of the matching scheme in Step 4. It serves as a crucial data foundation in the proposed strategy. Enhancing the computation speed of the shortest path can significantly augment the overall efficiency of strategy implementation.

Observations of the construction and dynamic changes in the parking lot traffic network reveal that although edge weights and node statuses in the network undergo modifications, the connectivity of the traffic network topology remains unchanged. This provides an opportunity to reduce redundant computation of shortest paths after network updates. In a network, the shortest path tree (SPT) refers to a spanning tree of a graph wherein every tree path constitutes the shortest path. In practical applications, the SPT often requires updates to accommodate changes in the network, with edge weight modifications being one of the most fundamental scenarios [48]. While recalculating the SPT entirely based on the adjusted network is a viable approach, it results in redundant and repetitive computations. This is recognized as the dynamic SPT problem, for which numerous algorithms have been proposed by scholars to provide solutions.

The ball-and-string algorithm, proposed by Narváez et al. [49], is an efficient method for solving dynamic SPT problems based on the Dijkstra algorithm. The algorithm treats network nodes as balls and edges as inelastic strings with lengths matching the edge weights. By lifting the ball representing the root node, the other balls in the network hang below it due to the gravity, connected by the strings. The hanging distance of each ball represents the shortest distance from the root node to the corresponding node, and the straightened strings represent the shortest paths. When there is a change in network edge weights, the algorithm adjusts the string lengths and observes the hanging states of balls to simulate the change in SPT. This algorithm always selects and extends the edge that leads to the minimum increase (or the maximum decrease) in path length, tending to consolidate vertices in the whole branch instead of only one vertex. Consequently, compared to other dynamic SPT algorithms, it involves significantly fewer edge changes to the existing SPT structure, thereby aiding in the maintenance of the tree's structural stability and coherence. This, in turn, culminates in enhanced overall efficiency.

The core steps of the ball-and-string algorithm include node selection, distance update and node search. If we denote the ball-and-string algorithm as a function BnS, its inputs and outputs can be expressed as

$$SPT_{updated} \leftarrow BnS\left(SPT_{original}, G_{original}, \Delta\right) \tag{12}$$

Here, $SPT_{updated}$ represents the SPT after updating the edge weights; $SPT_{original}$ represents the SPT before the edge weight updates; $G_{original}$ denotes a network with edge weights before updating; Δ represents a table containing the edges affected by weight changes and their corresponding change values.

In parking lot networks where edge weights dynamically change, a model can be established that transforms the problem of finding the shortest path from cruising vehicles to available parking spaces into a dynamic SPT problem, thereby reducing redundant computations. This study introduces the concept of parking-cruising path tree (PCPT) and proposes a fast computation method for determining the shortest cruising path for parking vehicles based on the ball-and-string algorithm.

In a parking lot, the starting point of a parking-cruising path is a specific point on the network where a vehicle is located, while the endpoint is a node corresponding to an available parking space. The PCPT refers to a tree structure that is constructed at a given moment, with the location of a cruising vehicle serving as the root and the nodes representing available parking spaces serving as the leaf nodes. These leaf nodes are connected to the root by the shortest paths. When one cruising vehicle is not exactly located on a network node, the position of the vehicle is defined as the "virtual root" of PCPT. The nearest node ahead of the vehicle's movement is the "real root" of the tree. As shown in Figure 5, the tree composed of green lines represents the current PCPT for the blue vehicle, marking the shortest paths to each available parking space.

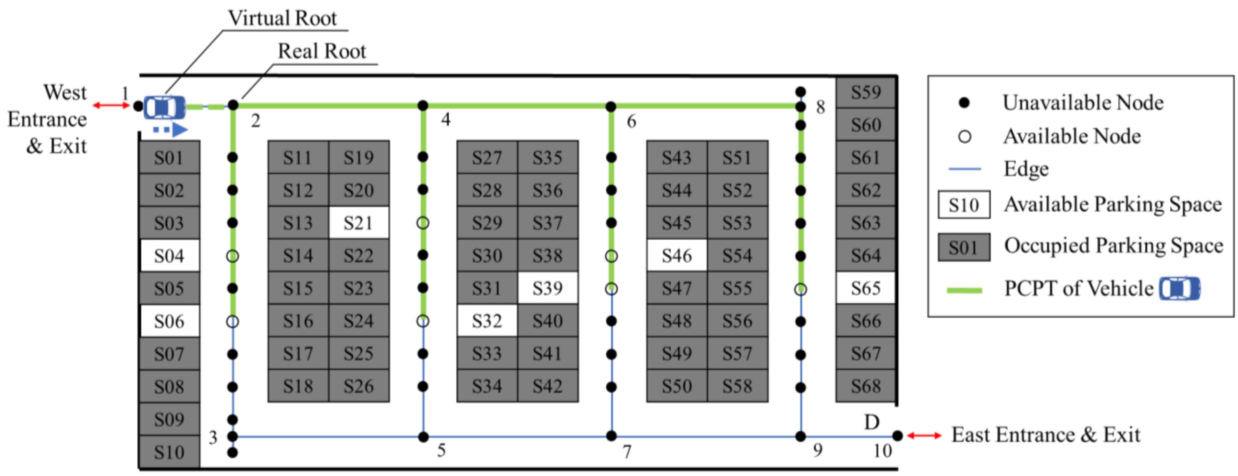


Figure 5. PCPT of a cruising vehicle within a parking lot.

Since in this study the topological connectivity is not altered during the update process, the ball-and-string algorithm can be utilized to efficiently compute the PCPT. According to the definition, PCPT is the tree obtained by removing all sub-trees from the SPT with the same root node that do not contain any available nodes, disregarding incomplete edges between the virtual root and the real root.

Assuming the previous round of scheme generation occurred at time p and the new round of generation occurs at time q, the initial $SPT_{node,0}$ values for each node in the parking lot are precomputed in the vehicle-free network. At time q, the fast computation method for updating PCPT based on ball-and-string is as the pseudocode below.

Algorithm PCPT computation for all parking-cruising vehicles

Input: $\boldsymbol{C}_p$, $\boldsymbol{C}_q$, $\boldsymbol{S}_q$, $PCPT_{f(i,p),p}$, $SPT_{f(i,p),p}\left(i \in \boldsymbol{C}_p\right)$, G_0, G_p, G_q

Output: $PCPT_{f(i,q),q}\left(i \in \boldsymbol{C}_q\right)$

1 Initialize $\Delta\left(G_p, G_q\right)$, $\Delta\left(G_0, G_q\right)$, $\boldsymbol{\delta}^{+}\left(G_p, G_q\right)$, $\boldsymbol{\delta}^{-}\left(G_p, G_q\right)$
2 **for** $i \in \boldsymbol{C}_q$ **do**
3   **if** $i \in \boldsymbol{C}_p$ and $f(i,p) = f(i,q)$ **then**
4   **if** $\boldsymbol{\delta}^{+}\left(G_p, G_q\right) \cap \boldsymbol{N}\left[PCPT_{f(i,p),p}\right] = \varnothing$ and $\boldsymbol{\delta}^{-}\left(G_p, G_q\right) \cap \left\{\boldsymbol{N}\left(G_p\right) - \boldsymbol{N}\left[PCPT_{f(i,p),p}\right]\right\} = \varnothing$ **then**
5             $PCPT_{f(i,q),q} \leftarrow PCPT_{f(i,p),p}$
6   continue to the next i directly
7   **else**
8             $SPT_{f(i,q),q} \leftarrow BnS\left[SPT_{f(i,p),p}, G_p, \Delta\left(G_p, G_q\right)\right]$
9   **end if**
10   **else**
11         $SPT_{f(i,q),q} \leftarrow BnS\left[SPT_{f(i,q),0}, G_0, \Delta\left(G_0, G_q\right)\right]$
12   **end if**
13     $PCPT_{f(i,q),q} \leftarrow SPT_{f(i,q),q}$
14   **for** $j \in \boldsymbol{N}\left[SPT_{f(i,q),q}\right]$ **do**
15   **if** no available node in the subtree of $SPT_{f(i,q),q}$ with root j **then**
16   cut the whole subtree of $SPT_{f(i,q),q}$ with root j from $PCPT_{f(i,q),q}$
17   **end if**
18   **end for**
19 **end for**

In the same way, the $PCPT'_{f(i,t),r}$ after r iterations will also be calculated based on the PCPT of previous iterations at the same time.

According to theoretical derivation by ball-and-string's proposers, this algorithm has a time complexity that, in the worst-case scenario, is no higher than that of classical algorithms such as dynamic Bellman-Ford or Dijkstra. Simulation experiments have indicated that the algorithm possesses significantly lower observed average complexity [49]. In the provided PCPT computation algorithm, Lines 8 and 11 eliminate the redundant steps of recomputing the SPT from scratch by using the ball-and-string algorithm. This reduction in computation time is particularly significant for applications with a larger quantity of cruising vehicles in the parking lot with a larger-scale traffic network. This engineering significance can help improve the efficiency of guidance scheme's formulation.

In summary, with Step 1 the strategy promptly captures changes in the status of vehicles and parking spaces within the parking lot to update the scheme and adapt to the latest dynamics. Step 2 demonstrates the strategy's adaptation to HV characteristics, coordinating HVs and AVs while prioritizing human drivers. Step 3 digitizes changes in physical elements into traffic network adjustments, facilitating efficient PCPT computation. This fast solution of PCPT also provides an efficient data source for subsequent steps. Lastly, Step 4 employs iterative vehicle loading and programming models to obtain the optimal matching scheme for vehicle-path-space. This step implements dynamic allocation and coordination among vehicles, emphasizing efficient resource utilization and coordination. The aforementioned process enables effective guidance for vehicles seeking available parking spaces and optimizes overall parking management.

5. Case studies

Two studies are analyzed in this section: A small parking lot to illustrate the working principles and steps of DCS, and a large parking lot with a complex internal structure to verify the effectiveness of the proposed strategy.

5.1. A simple case for DCS illustration

To provide a more intuitive demonstration of the proposed parking space allocation and path guidance strategy, a parking lot with 68 spaces and a simple spatial structure is used for the first part of the case study. The internal status at time t is depicted in Figure 6.

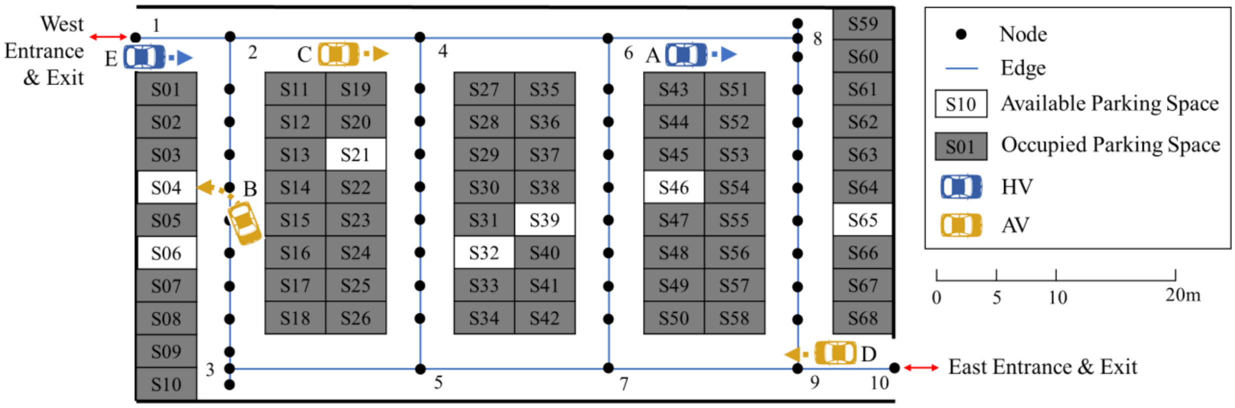


Figure 6. The structure and status within the parking lot.

Prior to the formulation of the parking space allocation and path guidance scheme at time t, five vehicles, encompassing two HVs and three AVs are cruising within the parking lot. Among them, Vehicle B is in the process of reversing into Space S04, consequently obstructing the lane, with an anticipated remaining duration of 40 seconds. Concurrently, Vehicle E has recently entered the parking lot via the western entrance. At this juncture, the triggering criterion 3 in Step 1 is satisfied, prompting the initiation of a new round of scheme formulation.

In Step 2, the identification of preoccupied parking spaces is conducted specifically for vehicles A and E, which are HVs. It is found that Vehicle A is approaching Space S65 and is moving away from all other available spaces. Therefore, Vehicle A meets the condition for preoccupying S65. In contrast, Vehicle E cannot preoccupy any available space.

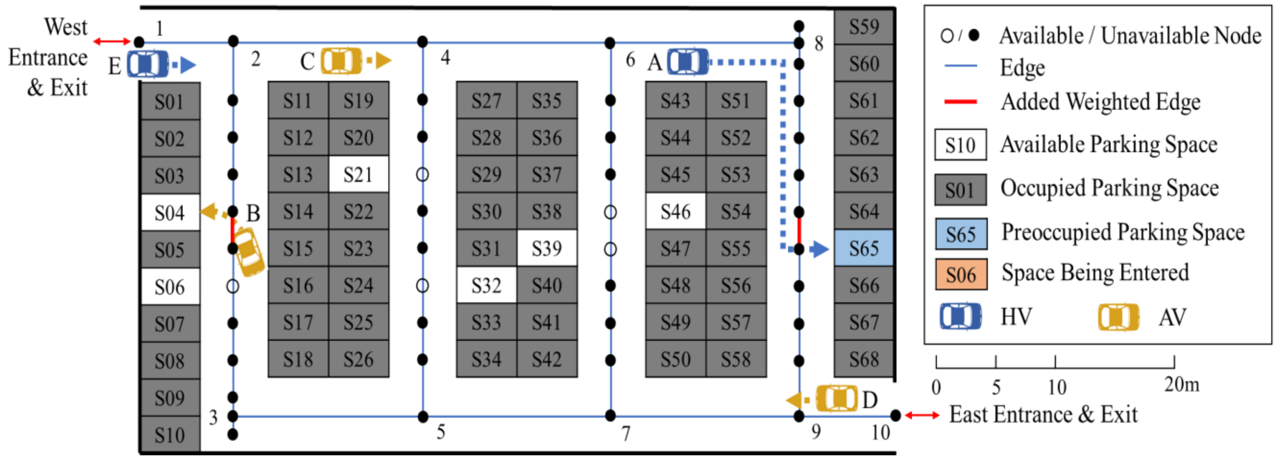


Figure 7. Updated traffic network after Step 3.

Based on the results of previous steps, the traffic network is updated at time t by edge weight adjustments and node state changes in Step 3, as shown in Figure 7. Edge weights of two road segments (marked in red) are increased and some nodes are distinguished as available or unavailable. It should be noted that the unavailability of a node only means that the node cannot be used as the destination of a path guide. The PCPT of each cruising vehicle is computed with the algorithm using ball-and-string for a quick search of the shortest paths between the vehicle and available spaces.

In Step 4, the parking space allocation and path guidance scheme for Vehicles C, D and E, which have not yet found or reserved parking spaces, will be formulated based on the updated traffic network. The initial matching model will be solved in Steps 4.1 and 4.2 and the vehicle loading results are shown in Table 2. The total remaining parking-cruising time after this iteration Z_1 is 89.4 seconds.

Table 2. Results of initial matching and vehicle loading.

Vehicle i	Allocated Parking Space j	Path （Omitting some nodes）	Estimated Remaining Parking-Cruising Duration $R_i = d_{ij}^1$ (seconds)
C	S21	4-S21	9.6
D	S39	9-7-S39	17.4
E	S32	1-2-4-S32	62.4
	Value of Overall Objective Function Z_1		89.4

It can be seen from the table that Vehicle E has an extremely high remaining cruising duration, occupying most of the objective function. Combined with Figure 7, we can observe that part of the paths of Vehicles C and E overlap, while C starts reversing into a parking space on E's path, blocking the lane that E needs to pass through. Therefore, with the highest remaining parking-cruising duration, E is selected as i_1.

Next, the iterative process continues. Vehicles E, D and C are sequentially substituted into model Eq (10) in descending order of R_i, and the matching parking spaces and paths for these three vehicles in the second iteration are solved, as shown in Table 3.

Table 3. Results of matching and vehicle loading after the second iteration.

Vehicle i	Allocated Parking Space j	Path (Omitting some nodes)	Estimated Remaining Parking-Cruising Duration $R_i = d_{ij}^2$ (seconds)
C	S32	4-S32	13.4
D	S39	9-7-S39	17.4
E	S21	1-2-4-S21	18.6
	Value of Overall Objective Function Z_2		49.4

It can be seen that the objective function value Z_2 after the second iteration is significantly lower than the previous Z_2. Although there is a slight increase in the remaining parking-cruising duration for Vehicle C, the reduction in it for Vehicle E is more noteworthy. This is attributable to the absence of vehicles obstructing the lanes of other vehicles in this scheme. Despite spatial overlaps in the paths of Vehicles C and E, C is no longer impacted by lane blocking, as it will traverse this particular location before E reaches Space S21.

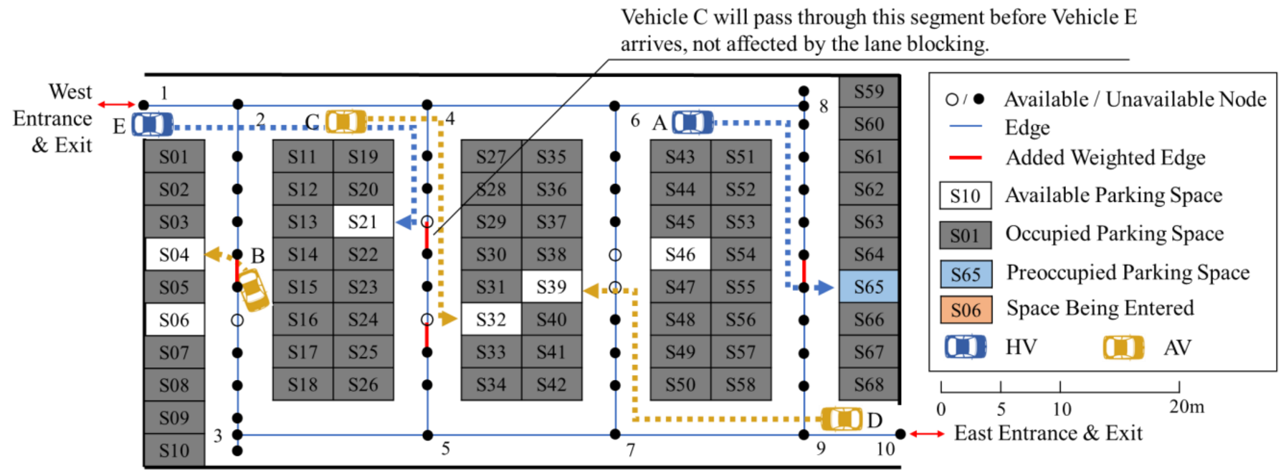


Figure 8. Final parking space allocation and path guidance scheme.

The total objective function value of the final scheme is 49.4 seconds, which is 40 seconds lower than the initial result. As analyzed earlier, the vehicle loading test and iterative process ensures that there are no situations where a front vehicle blocks the passage of a rear vehicle due to reversing into a parking spot and blocking a lane. In accordance with the scheme, each cruising vehicle is capable of reaching its assigned parking space without encountering any forced delays caused by other vehicles. Delays potentially arising from congestion or competition for parking spaces are circumvented through temporal or spatial allocation. This is of great significance for improving the efficiency of vehicle cruising in the parking lot. The complete process of the proposed DCS is illustrated in this section.

5.2. Analysis of DCS based on simulation of a large-scale parking lot

This study also uses a large-scale parking lot as a case to analyze the effectiveness of DCS through simulation.

5.2.1. Case description and data preparation

This case parking lot is derived from reality in Nanjing, China. It is affiliated with a commercial building encompassing various functionalities such as shopping, dining, entertainment and office spaces. There is a significant parking demand during both weekdays and weekends. Compared to the case in Section 5.1, this parking lot has a more complex internal structure and a larger quantity of parking spaces.

As depicted in Figure 9, this parking lot comprises two levels and accommodates a total of 576 parking spaces with two entrances. Only passenger vehicles with a capacity of no more than seven occupants or minivans are permitted to access the parking lot. All available parking spaces can accommodate vehicles of the above types. We assume that this parking lot is equipped with a PGIS integrated with DCS, which can optimize the guidance for mixed driving vehicles.

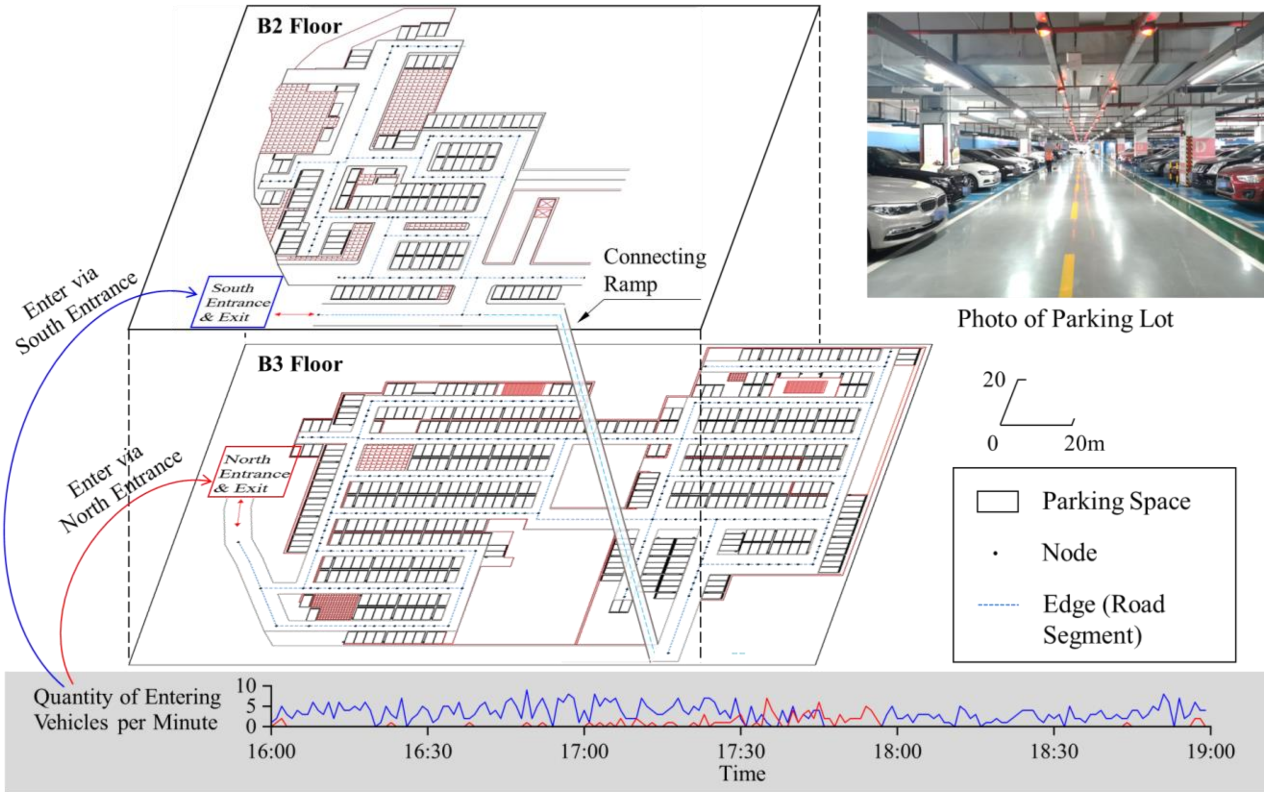


Figure 9. Internal structure and vehicle entry-exit information of the large-scale parking lot.

To compare the optimization effect of the newly proposed DCS with that in existing research, we use SFS as the control experiment group in this study. SFS allocates parking spaces and plans routes for each vehicle based on its status when it enters the parking lot. Once the vehicle enters the parking lot, its allocated parking space and planned path will not be updated due to changes in status within the parking lot.

In addition to using SFS and DCS as controls, two variables will be set to analyze the characteristics of the proposed strategy under different conditions:

1) AV penetration rate, which refers to the proportion of AVs in mixed driving vehicles, is an important attribute of mixed driving scenarios. In the future, it is foreseeable that the proportion of AVs in mixed driving scenarios is expected to continuously increase. Therefore, this study will set the

AV penetration rate in the simulation experiments in increments of 10% within 100%.

2) Initial saturation of the parking lot, which refers to the proportion of available parking spaces before the simulation starts, is adjusted at intervals of 10% within the range of 50 to 90%. The initial occupancy of each parking space in the parking lot will be randomly determined according to the initial saturation.

Additionally, an on-site investigation was conducted at this parking lot during a high-demand period on a weekend afternoon. The investigation encompassed the recording of vehicles entering and exiting the parking lot, as well as some characteristics of vehicular movement within the parking facility. Through tailgating and stationary surveys, data about vehicular cruising speed, paths and lane blocking within the parking lot was acquired. This information will be partially utilized in setting the parameters for the simulation.

During the three-hour simulation period, entry and exit times, as well as specific entrances and exits used by each vehicle were derived from the on-site investigations (the lowermost part of Figure 9). Within the simulation period, 676 vehicles enter and 624 vehicles exit the parking lot, ensuring relatively stable saturation. The type (AV or HV) of vehicles is randomly determined based on the preset AV penetration rate. Considering the possibility of HV drivers violating the guidance scheme provided by PGIS, a random probability between 0 and 100% is assigned to each driver for disregarding the scheme at each intersection or available parking space. Other parameters are also set regarding the on-site investigation of the case parking lot or relevant reference, as shown in Table 4.

Table 4. Some parameter settings in the simulation.

Input parameter	Symbol	Value	Source or reference
Vehicle velocity	v	2 m/s	On-site investigation statistics show that the average cruising speed within the parking lot is 2.34 m/s, and this is determined in conjunction with [16,46].
Estimated duration of HV reversing into parking spaces and blocking lanes	Big M	100 s	On-site investigation indicates that over 85% of human drivers need a duration ranging between 10 to 45 s to block the lane before reversing into a parking space, and the longest possible duration is determined based on [50].
Actual duration for HV reversing into parking spaces and blocking lanes	n/a	Randomly set between 10 and 100 s	
Duration for AV reversing into perpendicular space and blocking lane	$tr_{AV,perp}$	40 s	Determined through comprehensive consideration of sources including literature [16] and evaluation records [51,52].
Duration for AV reversing into parallel space and blocking lane	$tr_{AV,para}$	20 s	

5.2.2. Result analysis

For each combination of conditions, the numerical simulation program runs 10 times to eliminate random effects. First, the efficiency of updating and computing the shortest paths using PCPT was verified according to the algorithm in Section 4.5. The simulations were conducted on a computer equipped with an AMD Ryzen^{TM} 7 3800X processor and 16GB of memory. Using the Dijkstra algorithm directly for path computation instead of PCPT resulted in an average total duration of 64.1 s for optimizing the guidance scheme in each simulation for all entering vehicles. However, when

PCPT was used for fast path solving, the duration decreased to 41.7 s, representing a significant improvement. This improvement is mainly attributed to PCPT, effectively utilizing the existing network calculation results for necessary updates.

Figure 10(a) displays the simulation outcomes of the average parking-cruising duration under various combinations of initial saturation and AV penetration rate. Furthermore, to observe the degree of superiority of the DCS compared to traditional strategy, we calculated the optimization ratio (OR) using the following formula and plotted the results in Figure 10(b).

$$OR_{ij} = \frac{PCD_{ij}^{SFS} - PCD_{ij}^{DCS}}{PCD_{ij}^{SFS}} , \tag{13}$$

where PCD_{ij}^{SFS} and PCD_{ij}^{DCS} represents the average parking-cruising duration under condition of AV penetration rate i and initial saturation j, employing SFS and DCS, respectively.

The calculation results of optimization ratio in Figure 10(b) show that:

1) Compared to the traditional strategy SFS, the proposed DCS can reduce the average parking-cruising duration by up to 18%, indicating a significant optimization effect under some conditions.

2) The OR varies under different conditions. DCS has a higher OR as the AV penetration rate decreases or saturation increases. Moreover, when the AV penetration rate is high, OR shows a strong negative response to it.

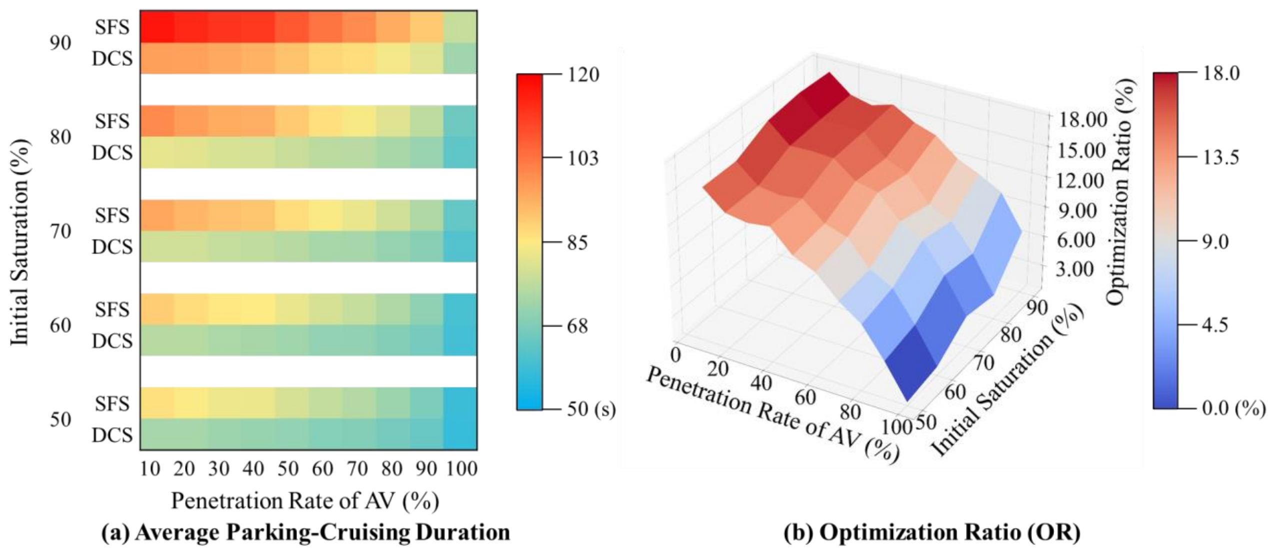
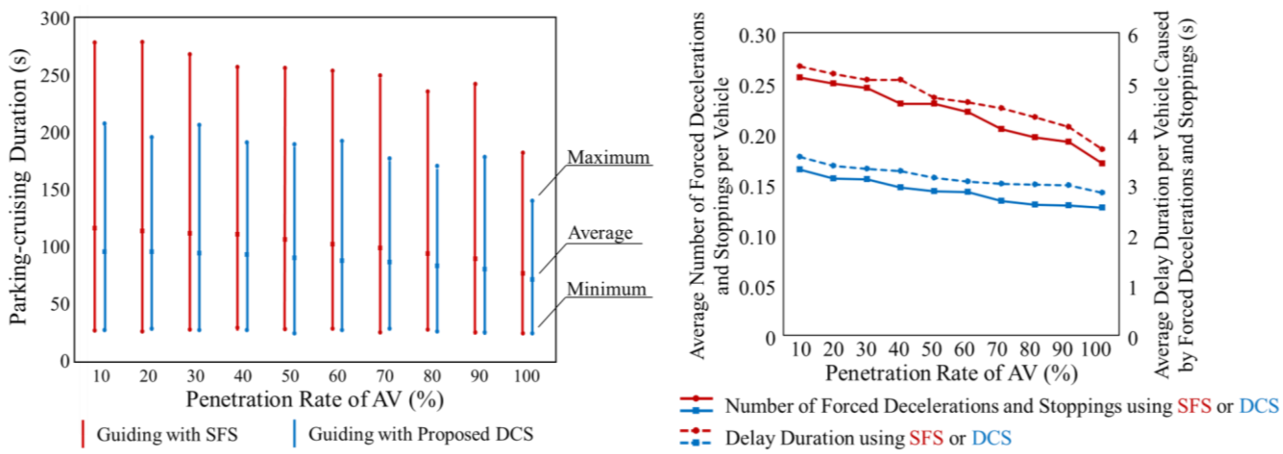


Figure 10. Simulation results under various conditions.

In addition to the average cruising duration, the distribution of vehicle cruising durations can reflect the potential extreme situations caused by different strategies. For the condition where the vehicle cruising time is longer and the proposed DCS performs well, with an initial saturation of 90%, we have collected statistics on the distributing range of vehicle cruising durations (Figure 11(a)). Through each vertical line depicted in this figure, the maximum, minimum and average parking cruising durations under each condition can be observed.

It is evident that under different AV penetration rates, the proposed DCS does not have a significant impact on the minimum cruising duration of vehicles. However, it can significantly reduce the maximum value of cruising duration. That is, the phenomenon of extremely long cruising durations is eliminated.



(a) Range of Cruising Duration **(b) Forced Delay Caused by Lane Blocking**

Figure 11. Statistics on Cruising Duration and Forced Delay.

Similarly, at the initial saturation level of 90%, we compiled statistics on the average quantity of forced decelerations and stoppings per vehicle during cruising, along with the corresponding delay durations (as shown in Figure 11(b)). Please note that red represents the performance of the SFS, while blue denotes that of DCS. Solid lines correspond to the primary vertical axis on the left and dashed lines utilize the secondary vertical axis on the right. DCS is able to significantly reduce the number of forced decelerations and stoppings as well as delays experienced by vehicles. Likewise, the magnitude of reduction is generally inversely correlated with the AV penetration rate.

In connection with the ideas in DCS, the reasons for above results can be deduced.

1) The presence of human drivers in HVs is a key factor contributing to the occurrence of unpredictable events. Compared to the traditional SFS, DCS focuses more on real-time dynamic optimization to avoid the adverse consequences of HV violating the scheme. Furthermore, the concept of preoccupied parking spaces also enables AVs to avoid conflicts with HVs promptly. Therefore, the impact of noncompliant HVs on AVs' parking space allocation (for example, in Figure 1(b)) is weakened, which can highlight the effectiveness of DCS, especially when AV penetration rate is not high enough.

2) When available parking spaces are limited and multiple vehicles are cruising at the same time, DCS can dynamically update the scheme from a coordinated perspective to reduce delays. Specifically, the vehicle loading test in iterative optimization involves coordinated adjustments among multiple vehicles, which helps alleviate the phenomenon of lane blocking and reduces ineffective waiting time.

Overall, the proposed DCS is able to optimize parking space allocation and provide path guidance for mixed driving parking lots, ultimately enhancing the efficiency and performance of vehicles' parking-cruising. Compared to the existing guidance strategy, the method is particularly effective when HVs have a high share and do not fully comply with guidance, or when the parking lot with high saturation represents low fault tolerance.

6. Conclusions

The SFS introduced by previous studies, which only performed when vehicles entered the parking lot through the entrance, struggled to cope with the adverse consequences of HV drivers violating the induction scheme. Furthermore, it could not dynamically update the parking guidance scheme from a cooperative standpoint, thereby failing to mitigate unnecessary cruising duration and forced delay.

To overcome the above limitations and effectively deal with uncontrollable and unpredictable changes as well as multiple mixed driving vehicles cruising simultaneously, this study presented a dynamic and coordinated strategy for parking space allocation and path guidance. The process for formulating guidance schemes for HVs and AVs was designed based on real-time status inside the parking lot, consisting of four steps: Triggering scheme formulation, identifying preoccupied parking space, updating the parking lot network based on current status and optimizing the vehicle-path-space matching scheme. By cyclically executing these steps, the scheme can be dynamically updated to meet real-time status and avoid negative impacts caused by incomplete scheme execution by HVs as well as lane blocking.

Two case studies were conducted to demonstrate the effectiveness of DCS. A simple case of a small parking lot was provided to illustrate its steps. Simulation based on a large parking lot was conducted to verify and analyze the optimization effect of DCS. Through simulations under different conditions, compared to the traditional SFS, the proposed DCS was found to (a) reduce the parking-cruising duration and forced delays of vehicles, especially for those with particularly long cruising durations and (b) perform well in low AV penetration rates or high saturation statuses of parking lots. These results demonstrated DCS's superiority in dynamically formulating schemes under low fault tolerance conditions in a mixed driving parking lot.

Main contributions of this study are the following three points.

1) A process was designed to allocate parking spaces and paths in a mixed driving parking lot and mitigate the adverse effects of HVs' not fully following guidance. This was achieved by updating the scheme based on real-time status and preoccupying spaces promptly.

2) The PCPT model was introduced to effectively reduce redundant computation in shortest-path solutions within the traffic network, thereby improving computational efficiency in engineering practice.

3) An objective optimization model was established based on binary programming to match cruising vehicles, paths and available parking spaces, being able to achieve the objective of minimizing the total remaining parking-cruising duration. An iterative method aligning closely with the actual vehicle operation process was proposed, which considers dynamic changes in edge weights and involves vehicle loading tests to avoid lane blocking and iteratively refine the matching solution.

Some certain limitations existing in this study can be further discussed and improved in future research.

a) The current method for parking space preoccupancy is still relatively rudimentary, leading to the determination being made only when there is one available parking space left that is being approached by an HV. By studying driver behavior, the identification method for preoccupancy can be improved, enabling a more accurate and predictable formulation of guidance schemes for other vehicles. In particular, given the complexity of deciphering the decision-making mechanisms of human drivers, it is feasible to draw on emerging technologies already applied in other fields for analysis, including language models, generation constructive hyper-heuristic and others [53–56].

b) Subjective consciousness and agency of vehicle control are critical, but not the only differences between HVs and AVs. Studying the features of HVs and AVs from economic, energy and vehicle engineering perspectives can be expanded to reflect their differences in social and physical aspects in the establishment of optimization models [23]. This multidimensional approach may enhance the effectiveness of parking lot management.

c) This study employed a binary programming model with PCPT to solve the vehicle-path-space matching scheme. This is a relatively traditional method, advantageous for obtaining precise solutions but tends to consume excessive computational time in large networks, which is not conducive to efficient decision-making. Moreover, as scenarios expand, the quantity of involved elements and variables increases, and the back-end model may not timely assimilate all information for decision-making. The non-immediacy and uncontrollability of problems could diminish the advantages of traditional algorithms. Therefore, in future scenarios involving extremely large-scale parking facilities or urban parking lot groups, advanced optimization algorithms such as adaptive algorithms [57,58], metaheuristics [59,60] and hybrid algorithms [61,62], which have been widely applied in other fields including online learning and data science, could be applied to address distribution and scheduling issues in parking guidance.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (grant No. 71971059 and No. 52131203).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. R. C. Hampshire, D. Shoup, What Share of Traffic is Cruising for Parking?, *J. Transp. Econ. Policy*, **52** (2018), 184–201.
2. G. D. Chiara, A. Goodchild, Do commercial vehicles cruise for parking? Empirical evidence from Seattle, *Transp. Policy*, **97** (2020), 26–36. https://doi.org/10.1016/j.tranpol.2020.06.013
3. H. M. Qin, X. H. Yang, Y. J. Wu, H. Z. Guan, P. F. Wang, N. Shahinpoor, Analysis of parking cruising behaviour and parking location choice, *Transp. Plan. Technol.*, **43** (2020), 717–734. https://doi.org/10.1080/03081060.2020.1805545
4. R. Grbić, B. Koch, Automatic vision-based parking slot detection and occupancy classification, *Expert Syst. Appl.*, **225** (2023), 120147. https://doi.org/10.1016/j.eswa.2023.120147
5. J. Parmar, P. Das, S. M. Dave, Study on demand and characteristics of parking system in urban areas: A review, *J. Traffic Transp. Eng.*, **7** (2020), 111–124. https://doi.org/10.1016/j.jtte.2019.09.003
6. F. Al-Turjman, A. Malekloo, Smart parking in IoT-enabled cities: A survey, *Sust. Cities Soc.*, **49** (2019), 101608. https://doi.org/10.1016/j.scs.2019.101608
7. T. Lin, H. Rivano, F. Le Mouël, A survey of smart parking solutions, *IEEE Trans. Intell. Transp. Syst.*, **18** (2017), 3229–3253. https://doi.org/10.1109/TITS.2017.2685143
8. X. Di, R. Shi, A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning, *Transp. Res. Pt. C-Emerg. Technol.*, **125** (2021), 103008. https://doi.org/10.1016/j.trc.2021.103008
9. Z. Wang, Y. Shi, W. Tong, Z. Gu, Q. Cheng, Car-following models for human-driven vehicles and autonomous vehicles: A systematic review, *J. Transp. Eng., Part A: Syst.*, **149** (2023), 04023075. https://doi.org/10.1061/JTEPBS.TEENG-7836

10. Y. Geng, C. G. Cassandras, Dynamic resource allocation in urban settings: A "smart parking" approach, in *2011 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, IEEE, (2011), 1–6. https://doi.org/10.1109/CACSD.2011.6044566
11. T. K. Chan, C. S. Chin, Review of autonomous intelligent vehicles for urban driving and parking, *Electronics*, **10** (2021), https://doi.org/10.3390/electronics10091021
12. P. S. Perumal, M. Sujasree, S. Chavhan, D. Gupta, V. Mukthineni, S. R. Shimgekar, et al., An insight into crash avoidance and overtaking advice systems for Autonomous Vehicles: A review, challenges and solutions, *Eng. Appl. Artif. Intell.*, **104** (2021), 104406. https://doi.org/10.1016/j.engappai.2021.104406
13. K. Huang, C. Jiang, P. Li, A. Shan, J. Wan, W. Qin, A systematic framework for urban smart transportation towards traffic management and parking, *Electron. Res. Arch.*, **30** (2022), 4191–4208. https://doi.org/10.3934/era.2022212
14. X. Zhang, C. Zhao, F. Liao, X. Li, Y. Du, Online parking assignment in an environment of partially connected vehicles: A multi-agent deep reinforcement learning approach, *Transp. Res. Part C: Emerg. Technol.*, **138** (2022), 103624. https://doi.org/10.1016/j.trc.2022.103624
15. K. Ata, A. C. Soh, A. Ishak, H. Jaafar, N. Khairuddin, Smart indoor parking system based on dijkstra's algorithm, *Int. J. Integr. Eng*, **2** (2019), 13–20.
16. L. Yu, H. Jiang, L. Hua, Anti-congestion route planning scheme based on Dijkstra algorithm for automatic valet parking system, *Appl. Sci.-Basel*, **9** (2019), 5016. https://doi.org/10.3390/app9235016
17. M. Wu, H. Jiang, C. A. Tan, Automated parking space allocation during transition with both human-operated and autonomous vehicles, *Appl. Sci.*, **11** (2021), 855. https://doi.org/10.3390/app11020855
18. C. Lu, C. Liu, Ecological control strategy for cooperative autonomous vehicle in mixed traffic considering linear stability, *J. Intell. Connected Veh.*, **4** (2021), 115–124. https://doi.org/10.1108/JICV-08-2021-0012
19. E. F. Ozioko, J. Kunkel, F. Stahl, Road intersection coordination scheme for mixed traffic (human-driven and driverless vehicles): A systematic review, *J. Adv. Transp.*, **2022** (2022), 2951999. https://doi.org/10.1155/2022/2951999
20. A. Schieben, M. Wilbrink, C. Kettwich, R. Madigan, T. Louw, N. Merat, Designing the interaction of automated vehicles with other traffic participants: design considerations based on human needs and expectations, *Cogn. Technol. Work*, **21** (2019), 69–85. https://doi.org/10.1007/s10111-018-0521-z
21. H. Shi, D. Chen, N. Zheng, X. Wang, Y. Zhou, B. Ran, A deep reinforcement learning based distributed control strategy for connected automated vehicles in mixed traffic platoon, *Transp. Res. Part C: Emerg. Technol.*, **148** (2023), 104019. https://doi.org/10.1016/j.trc.2023.104019
22. S. Gong, L. Du, Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles, *Transp. Res. Part B: Methodol.*, **116** (2018), 25–61. https://doi.org/10.1016/j.trb.2018.07.005
23. Y. He, Y. Liu, L. Yang, X. Qu, Deep adaptive control: Deep reinforcement learning-based adaptive vehicle trajectory control algorithms for different risk levels, *IEEE Trans. Intell. Veh.*, (2023), 1–12. https://doi.org/10.1109/TIV.2023.3303408
24. S. Yan, T. Welschehold, D. Büscher, W. Burgard, Courteous behavior of automated vehicles at unsignalized intersections via rreinforcement learning, *IEEE Robot. Autom. Lett.*, **7** (2022), 191–198. https://doi.org/10.1109/LRA.2021.3121807

25. S. Aoki, R. Rajkumar, Safe intersection management with cooperative perception for mixed traffic of human-driven and autonomous vehicles, *IEEE Open J. Veh. Technol.*, **3** (2022), 251–265. https://doi.org/10.1109/OJVT.2022.3177437
26. Z. H. Khattak, B. L. Smith, M. D. Fontaine, J. Ma, A. J. Khattak, Active lane management and control using connected and automated vehicles in a mixed traffic environment, *Transp. Res. Part C: Emerg. Technol.*, **139** (2022), 103648. https://doi.org/10.1016/j.trc.2022.103648
27. J. Wang, L. Lu, S. Peeta, Z. He, Optimal toll design problems under mixed traffic flow of human-driven vehicles and connected and autonomous vehicles, *Transp. Res. Part C: Emerg. Technol.*, **125** (2021), 102952. https://doi.org/10.1016/j.trc.2020.102952
28. W. Yue, C. Li, S. Wang, N. Xue, J. Wu, Cooperative incident management in mixed traffic of CAVs and human-driven vehicles, *IEEE Trans. Intell. Transp. Syst.*, (2023), 1–15. https://doi.org/10.1109/TITS.2023.3289983
29. S. Bahrami, M. J. Roorda, Optimal traffic management policies for mixed human and automated traffic flows, *Transp. Res. Part A: Policy Pract.*, **135** (2020), 130–143. https://doi.org/10.1016/j.tra.2020.03.007
30. Z. Guo, D. Z. W. Wang, D. Wang, Managing mixed traffic with autonomous vehicles–A day-to-day routing allocation scheme, *Transp. Res. Part C: Emerg. Technol.*, **140** (2022), 103726. https://doi.org/10.1016/j.trc.2022.103726
31. T. Roughgarden, É. Tardos, How bad is selfish routing?, *J. ACM*, **49** (2002), 236–259. https://doi.org/10.1145/506147.506153
32. L. Ma, S. Qu, L. Song, J. Zhang, J. Ren, Human-like car-following modeling based on online driving style recognition, *Electron. Res. Arch.*, **31** (2023), 3264–3290. https://doi.org/10.3934/era.2023165
33. I. Klein, N. Levy, E. Ben-Elia, An agent-based model of the emergence of cooperation and a fair and stable system optimum using ATIS on a simple road network, *Transp. Res. Part C: Emerg. Technol.*, **86** (2018), 183–201. https://doi.org/10.1016/j.trc.2017.11.007
34. Y. Yang, J. Chen, J. Ye, J. Chen, Y. Luo, Joint optimization of facility layout and spatially differential parking pricing for parking lots, *Transp. Res. Record*, **2677** (2023), 241–257. https://doi.org/10.1177/03611981221145139
35. L. Cheng, C. Liu, B. Yan, Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot, in *2014 IEEE International Conference on Information and Automation (ICIA)*, IEEE, (2014), 695–698. https://doi.org/10.1109/ICInfA.2014.6932742
36. X. Wei, Z. Lei, Q. Ze-hua, Z. Yong, Z. Tie-nan, Design of parking lot path guidance system based on multi-objective Point A* Algorithm, *Comput. Modernizat.*, (2020), 40.
37. W. Wang, M. Wang, Y. Yue, Y. Yang, A multi-objective parking space decision-making and path planning method, in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, (2021), 672–677. https://doi.org/10.1109/ICUS52573.2021.9641394
38. G. Wang, T. Nakanishi, A. Fukuda, Time-varying shortest path algorithm with transit time tuning for parking lot navigation, in *TENCON 2015-2015 IEEE Region 10 Conference*, IEEE, (2015), 1–6. https://doi.org/10.1109/TENCON.2015.7373015
39. S. Gai, X. Zeng, X. Yue, Z. Yuan, Parking guidance model based on user and system bi-level optimization algorithm, *J. Jilin Univ. (Eng. Technol. Edit.)*, **52** (2022), 1344–1352. https://doi.org/10.13229/j.cnki.jdxbgxb20210093

40. Y. Chen, T. Wang, X. Yan, C. Wang, An ensemble optimization strategy for dynamic parking-space allocation, *IEEE Intell. Transp. Syst. Mag.*, **15** (2023), 347–362. https://doi.org/10.1109/MITS.2022.3163506
41. D. C. Tseng, T. W. Chao, J. W. Chang, Image-based parking guiding using ackermann steering geometry, *Appl. Mech. Mater.*, **437** (2013), 823–826. https://doi.org/10.4028/www.scientific.net/AMM.437.823
42. C. Löper, C. Brunken, G. Thomaidis, S. Lapoehn, P. P. Fouopi, H. Mosebach, et al., Automated valet parking as part of an integrated travel assistance, in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, (2013), 2341–2348. https://doi.org/10.1109/ITSC.2013.6728577
43. W. Fu, Q. Li, Z. Fan, Research on automatic parking path optimization based on Ackerman model, *Acad. J. Comput. Inf. Sci.*, **5** (2022), 50–57. https://doi.org/10.25236/ajcis.2022.050808
44. Y. Yao, Z. Feng, Z. An, Automatic parking problem based on path planning, in *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, (2023), 110–115. https://doi.org/10.1109/ICCECT57938.2023.10141401
45. L. Yuan, R. Huang, L. Han, M. Zhou, A Parking Guidance Algorithm Based on Time-Optimal Dynamic Sorting for Underground Parking, in *2018 26th International Conference on Geoinformatics*, (2018), 1–6. https://doi.org/10.1109/GEOINFORMATICS.2018.8557079
46. M. Kneissl, A. K. Madhusudhanan, A. Molin, H. Esen, S. Hirche, A multi-vehicle control framework with application to automated valet parking, *IEEE Trans. Intell. Transp. Syst.*, **22** (2021), 5697–5707. https://doi.org/10.1109/TITS.2020.2990294
47. P. Schörner, M. Conzelmann, T. Fleck, M. Zofka, J. M. Zöllner, Park my car! automated valet parking with different vehicle automation levels by v2x connected smart infrastructure, in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, (2021), 836–843. https://doi.org/10.1109/ITSC48978.2021.9565095
48. B. Xiao, H. N. Cao, Z. L. Shao, E. H. M. Sha, An efficient algorithm for dynamic shortest path tree update in network routing, *J. Commun. Networking*, **9** (2007), 499–510. https://doi.org/10.1109/JCN.2007.6182886
49. P. Narváez, K. Siu, H. Tzeng, New dynamic SPT algorithm based on a ball-and-string model, *IEEE-ACM Trans. Networking*, **9** (2001), 706–718. https://doi.org/10.1109/90.974525
50. *Driving Test Contents and Methods*, China: Ministry of Public Security of the People's Republic of China.
51. R. RSEV, Self-Parking cars Tesla v Audi v Ford v BMW - test and comparison-are they any good?, 2022. Available from: https://www.youtube.com/watch?v=nsb2XBAIWyA.
52. R. RSEV, Self-Parking Tesla Vision v Hyundai Ioniq 5 with remote parking v BMW i3, 2022. Available from: https://www.youtube.com/watch?v=-nBnZ7GqPiM.
53. Y. Liu, F. Wu, Z. Liu, K. Wang, F. Wang, X. Qu, Can language models be used for real-world urban-delivery route optimization?, *Innovation*, **4** (2023), 100520. https://doi.org/10.1016/j.xinn.2023.100520
54. X. Qu, H. Lin, Y. Liu, Envisioning the future of transportation: Inspiration of ChatGPT and large models, *Commun. Transp. Res.*, **3** (2023), 100103. https://doi.org/10.1016/j.commtr.2023.100103
55. H. Lin, Y. Liu, S. Li, X. Qu, How generative adversarial networks promote the development of intelligent transportation systems: A survey, *IEEE/CAA J. Autom. Sin.*, **10** (2023), 1781–1796. https://doi.org/10.1109/JAS.2023.123744

56. E. Singh, N. Pillay, A study of ant-based pheromone spaces for generation constructive hyper-heuristics, *Swarm Evol. Comput.*, **72** (2022), 101095. https://doi.org/10.1016/j.swevo.2022.101095
57. M. Chen, Y. Tan, SF-FWA: A Self-Adaptive Fast Fireworks Algorithm for effective large-scale optimization, *Swarm Evol. Comput.*, **80** (2023), 101314. https://doi.org/10.1016/j.swevo.2023.101314
58. M. A. Dulebenets, An Adaptive Polyploid Memetic Algorithm for scheduling trucks at a cross-docking terminal, *Inf. Sci.*, **565** (2021), 390–421. https://doi.org/10.1016/j.ins.2021.02.039
59. J. Pasha, A. L. Nwodu, A. M. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, et al., Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings, *Adv. Eng. Inf.*, **52** (2022), 101623. https://doi.org/10.1016/j.aei.2022.101623
60. P. Singh, J. Pasha, R. Moses, J. Sobanjo, E. E. Ozguven, M. A. Dulebenets, Development of exact and heuristic optimization methods for safety improvement projects at level crossings under conflicting objectives, *Reliab. Eng. Syst. Saf.*, **220** (2022), 108296. https://doi.org/10.1016/j.ress.2021.108296
61. Y. Liu, R. Jia, J. Ye, X. Qu, How machine learning informs ride-hailing services: A survey, *Commun. Transp. Res.*, **2** (2022), 100075. https://doi.org/10.1016/j.commtr.2022.100075
62. M. A. Dulebenets, A Diffused Memetic Optimizer for reactive berth allocation and scheduling at marine container terminals in response to disruptions, *Swarm Evol. Comput.*, **80** (2023), 101334. https://doi.org/10.1016/j.swevo.2023.101334