Please cite the paper as:

Wu W, Zhou H and Liu R (2024) Prediction-failure-risk-aware online dial-a-ride scheduling considering spatial demand correlation via approximate dynamic programming and scenario approach. **Transportation Research Part C**. 169, 104801. https://doi.org/10.1016/j.trc.2024.104801

Prediction-failure-risk-aware online dial-a-ride scheduling considering spatial demand correlation via approximate dynamic programming and scenario approach

Weitiao Wu[a1], Honghui Zou[a], Ronghui Liu[b]

a. School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510641, China

b. Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom

Abstract: The dial-a-ride (DAR) service is a precursor to emerging shared mobility. Service providers expect efficient management of fleet resources to improve service quality without degrading economic viability. Most existing studies overlook possible future demands that could yield better matching opportunities and scheduling benefits, and therefore have short-sighted limitations. Moreover, the effects of correlated demand and potential prediction errors were ignored. To address these gaps, this paper investigates prediction-failure-risk-aware online DAR scheduling with spatial demand correlation. Request selection and cancellation are explicitly considered. We formulate the problem as a Markov decision process (MDP) and solve it by approximate dynamic programming (ADP). We further develop a demand prediction model that can capture the characteristics of DAR travel demand (uncertainty, sparsity, and spatial correlation). Deep quantile regression is adopted to estimate the marginal distribution of each OD pair. These marginals are combined into a joint demand distribution by constructing a Gaussian Copula to capture the spatial demand correlation. A prediction error correction mechanism is proposed to eliminate prediction errors and rectify policies promptly. Based on the model properties, several families of customized pruning strategies are devised to improve the computational efficiency and solution quality of ADP. We solve policies over time in the dynamic environment mixed with actual and stochastic future demands via the ADP algorithm and scenario approach. We propose the value function rolling method and multi-scenario exploration method, to address the deviation of the value function and identify the optimal policy from multiple future demand scenarios. Numerical results demonstrate the importance and benefits of incorporating demand forecasting and spatial correlation into the DAR operation. The improvement due to prediction is significant even when the prediction is imperfect, while the demand prediction can hedge against the negative effects of request cancellation. The real-world application result shows that compared to state-of-the-practice, the overall delivery efficiency can be substantially improved, along with better service quality and fleet size savings.

Keywords: Dial-a-ride service; OD demand prediction; Spatial demand correlation; Approximate dynamic programming; Risk-aware decision-making

[1] Corresponding author, E-mail: ctwtwu@scut.edu.cn

1. Introduction

Public transportation is essential to urban mobility and development. Its operation usually compromises between service quality and operation cost. Conventional fixed-route service can be operated in high-demand areas but cannot be affordable in low-density demand areas. Taxis offer a good door-to-door, however, expensive service. The advancement in information and communication technologies creates both opportunities and challenges for new transportation services. Dial-a-ride (DAR), as an emerging public transport and ridesharing service, has been suggested as a system capable of fulfilling the dynamic and sparse demand, which is becoming increasingly important in both practice and the academy.

Although the DAR service can better cope with the ever-changing demand for transportation, it is often over three times more expensive than traditional fixed-route systems in that more fleet and crew resources are required to meet the unbalanced spatiotemporally distributed demand (Anderson et al., 2014). Crew costs usually take up to as high as 60% of the overall operation cost (Perumal et al., 2019). The operation cost and service quality are great concerns for transit service providers because they impact their profitability and long-term survival. In this highly competitive, growing market segment of on-demand mobility services, the service providers are often in a challenging dilemma of whether to maintain the economic viability of the system or maintain good services for passengers, which can trigger the 'vicious cycle' effect if the system is not well designed. Inexpensive solutions that do not involve new fleet resources are the most desired.

Traditionally, dynamic DAR problems are usually addressed by the repetitive one-period static model and myopic insertion heuristics, while having not fully taken advantage of historical demand data. However, traditional methods are generally sub-optimal because requests are matched independently between periods, and they fail to consider the future dynamics and measures of the whole system, making it difficult to adapt to a time-varying stochastic environment. Instead, on one hand, more informed decision-making could be achieved by joint planning for requests across multiple periods through more refined dynamic schedule adjustment strategies, such as request insertions and trip adjustments. On the other hand, an anticipating solution based on look-ahead information and foresighted prior fleet allocation can potentially enable more efficient and effective deployment of vehicles to travel demand. A possible way to exploit the vehicle capacity is to forecast future demand and then distribute the scarce fleet resources. The increasing availability of travel demand data from reservation platforms provides opportunities for moving beyond DAR services into a proactive operation mode. The historical travel demand data from the day-to-day operation can be used to predict future demand and dispatch vehicles preemptively, which contributes to improving service quality and system efficiency.

In this paper, we address the dynamic and stochastic problem for DAR services with an accurate demand-anticipatory predictor. We consider a fleet of vehicles with limited capacity operating in a general multi-depot context. Our problem determines simultaneously vehicle routing, departure times, and request selection. Seeking effective solutions for this problem is difficult due to the following complications. First, such DAR problems often deal with travel demand that reveals over time and requires decision-making under uncertainty of not only the appearance of future users and cancellations of users but also the operations concerning users who have already appeared. In the highly dynamic DAR system, a delayed arrival at a single station may affect the operational efficiency of subsequent routes. Accordingly, any joint optimization action can affect not only the immediate performance but also the future performance (i.e., the rewards of the following states). Thus, an efficient approach that can maximize long-run rewards and that can make dynamic decisions is highly desired for the real-time application of DAR services.

Second, the decision-making should be amenable to the unique characteristics of travel demand. Thus far, DAR problems have relied on the assumption that all customer presences are independent from each other. Unlike traditional bus services with fixed itineraries, the DAR service usually operates in lower-density regions subject to volatile demand (Chandakas, 2020). In the transit catchment area, the spatial-temporal activities in area-based OD pairs present latent features following some latent spatiotemporal distribution (Guo and Karimi, 2017). In theory, the OD movement between locations presents possibly spatial dependencies mainly attributed to spatial spillover effect or unobserved factors shared by adjacent zonal units. In practice, close neighborhoods might share a similar spatial and temporal mobility pattern. For example, mobility tends to be more active in the vicinity of the areas with activities of high intensity. When the travel demand is aggregated such as work and shopping movement, people move densely from a large number of dispersed origins to a specific functional area; when travel demand is evacuated such as home-bound movement, people from the same origins move widely. Meanwhile, a few recreation-based travel demands and short-distance traffic modes can induce movement in adjacent areas. These facts indicate the presence of possible simultaneous movements between certain OD pairs. The dynamic decision-making process will be affected by spatially correlated demand forecasts. For instance, a vehicle with only a single empty seat would not be selected to pick up a passenger if more passengers heading in the same direction are expected to arrive shortly at the nearby locations of the first passenger. Instead, a vehicle with a higher remaining capacity will be employed to serve the passenger. As such, overlooking spatial demand correlations can fail in demand fulfilment when optimizing and anticipating decisions.

Another challenge is that, even after making optimal decisions, inevitable prediction errors and passenger trip cancellations can lead to scheduling failure. This failure risk not only wastes the driver's work efforts, but also exacerbates the fleet resource scarcity in execution due to unfulfilled tasks, among other disappointing consequences. Thus, the service provider faces the challenge of how to design the correction mechanism for prediction errors and request cancellation to mitigate such negative effects.

Putting together the above challenges, this paper investigates a prediction-failure-risk-aware online DAR scheduling problem. Request selection and cancellation are also considered. Under the rolling horizon framework, the research problem is formulated as a Markov decision process (MDP), which aims to find the optimal policy that minimizes the expected cost over the planning horizon in each optimization period. To overcome the curse of dimensionality, we resort to approximate dynamic programming (ADP) to solve it. Our approach has the potential to act as a generic methodology for solving a wide range of similarly structured dynamic and stochastic problems in other application fields. Our model and algorithm are applied to a real-world case study in Guangzhou, China. Managerial insights are also provided.

Overall, our study makes contributions to the DAR problem as well as general methodologies of ADP:

First, we introduce a foresighted online DAR scheduling problem allowing for request selection and cancellation. Distinct from prior research, the correction mechanisms for prediction errors and cancelled requests are devised and embedded into the optimization framework to proactively mitigate the negative effects.

Second, we develop an OD demand prediction model to generate multiple future demand scenarios through sampling, and embed them into the MDP model using the scenario approach. The prediction model captures the unique characteristics of DAR travel demand (i.e., uncertainty, sparsity, and spatial demand correlation) by comprehensively using deep quantile regression and Copula function joint distribution.

Third, we combine the ADP algorithm and scenario approach to solve policies over time in the dynamic environment mixed with actual and stochastic future demands. The value function rolling method and multi-scenario exploration method are proposed to enhance the ADP algorithm, to tackle the challenge of deviation of the value function in iterations between adjacent periods and identify the optimal policy from multiple future demand scenarios.

Fourth, based on the model properties, we propose several families of customized pruning strategies for the dynamic programming formulation to derive a compact state space representation of provably near-optimal dispatching decisions. Experiments show that the pruning strategies can improve both solution quality and computation efficiency for ADP.

The rest of this paper is organized as follows. In Section 2, we review the related literature. In Section 4, the model is presented, followed by solution methodologies in Section 5. In Section 6, a numerical example and a real-world application are conducted and managerial insights are provided. Finally, the concluding remarks are provided.

2. Literature review

In recent years, there has been a resurgence of interest in shared mobility systems. In what follows, we begin by reviewing the on-demand ride services, proceed to review DAR problems, and finally solution approaches.

2.1. On-demand ride services

The emergence of mobile internet technologies has reshaped personal mobility. The successful deployment of ridesharing requires a high-quality balance of demand and supply (Wang and Yang, 2019; Qin et al., 2022). There has been considerable research on on-demand ride service, including matching (Qin et al., 2021), the vehicle routing/rebalancing problem (Guo et al., 2021), pricing (Chen et al., 2021), and equilibrium analysis for ride-sourcing markets (Ke et al., 2020; Ke et al., 2021; Zhou et al., 2022).

Ridesharing matching is an online bipartite matching problem in which both supply and demand are dynamic. Matching can be conducted continuously in a streaming manner or at fixed intervals (i.e., batching). Özkan and Ward (2020) presented parameter-based matching policies using a continuous linear program, considering different arrival rates and the amount of time that customers and drivers are willing to wait. Qin et al. (2021) proposed a multi-party matching problem with bundled option services. The authors developed an integer linear programming model with multiple objectives for on-line matching. Zhou et al. (2022) developed a matching optimization model for coexisting ride-hailing and ridesharing services accounting for traveler's mode choices. Feng et al. (2022) proposed a block-matching mechanism for the ride-sourcing system. The impacts of block size on key market measurements were discussed. Weidinger et al. (2023) studied the free-floating car-sharing system using instantaneous matching and batch matching, and they showed that the batch-matching mechanism greatly improves service performance.

The vehicle routing/rebalancing problem is to dynamically guide idle vehicles over the road network to alleviate the supply-demand imbalance. The decision is generally to determine the optimal rebalancing flows from one region to another. Braverman et al. (2019) modelled the ride-sharing network with a closed queueing network, and optimized empty-car routing based on a fluid model associated with the closed queueing network. Chen and Levin (2019) developed a linear programming model to optimize the car flows between regions, and they used an agent-based simulator to test the rebalancing policy. Guo et al. (2021) presented a robust optimization model for on-demand ride services comprised of a fleet of both autonomous vehicles and human-driven vehicles. Later, Guo et al. (2023) developed a data-driven robust optimization for contextual vehicle rebalancing for on-demand ride services considering uncertain demand. The model was formulated as mixed integer second-order cone programming solvable by commercial solvers. Wang et al. (2019) developed routing and scheduling approaches for the last-mile transportation system where passengers arrive in batches at the metro station, which was formulated as a mixed integer model and addressed by several computationally feasible heuristics. Later, Shehadeh et al. (2021) investigated the fleet sizing and allocation problem for the on-demand last-mile transportation system. They developed a stochastic programming model and a distributionally robust optimization model to solve the problem.

Request cancellation behavior has a considerable impact on the operational performance of ride-sourcing platforms, which is reflected by wasted platform and driver efforts to pick up passengers and reduced platform vehicle supply (He et al., 2018). Xu et al. (2022) discussed the effect of time-dependent cancellation behavior on the maximum number of completed trips, and they showed that relaxing request cancellation penalties can improve the utilization rate of taxis and the profits of service providers. Wang et al. (2019) investigated the relationship between pickup and delivery distance and passenger cancellation decisions, which showed that the platform's profitability can be enhanced by a well-designed penalty and compensation strategy.

2.2. The DAR problem

The DAR problems have drawn much research attention in the field of public transportation. The first trial of on-demand flexible buses, called dial-a-ride services, was provided to the elderly and disabled in the USA in 1970 and was first formulated by Psaraftis in 1980 (Psaraftis, 1980). Later, its application was extended to other areas, such as patient transportation (Lim et al., 2017; Berg and Essen, 2019) and demand-responsive feeder services (Ren et al., 2022; Galarza Montenegro et al., 2021). For an overview of research on DAR problems, see Ho et al. (2018) and Vansteenwegen et al. (2022).

The DAR problem can be viewed as a variant of the vehicle routing problem with pickup and delivery, which is generally to determine rider-vehicle matching and the associated routes. There are static and dynamic versions of DAR problems. In the static version, all riding requests are known in advance, and decisions are made a priori. In the dynamic case, the existing schedule might be modified during the time horizon in response to new requests. Early studies on DAR feature homogeneous vehicles (Psaraftis, 1980; Desrosiers et al., 1986). Subsequent research has been extended to considering more realistic and complex factors, such as heterogeneous vehicles characterized by differentiated equipment and capacities (Molenbruch et al., 2017; Detti et al., 2017), passenger transfers (Masson et al., 2013; Masson et al., 2014; Posada et al., 2017), and manpower requirements (Braekers and Kovacs, 2016). Qu and Bard (2013) developed a heterogeneous pickup and delivery problem, in which the vehicle capacity can be reconfigured by modifying the vehicle's inner configuration to comply with different customer demands. Later, Tellez et al. (2018) extended this work by including both heterogeneous fleets of vehicles and users in a DAR problem, considering different combinations of equipment required (e.g., wheelchairs and stretchers). The fundamental question of the DAR problem with transfers is how to synchronize passenger transfers between vehicles in space and time. Masson et al. (2014) addressed a DAR problem in which passengers can transfer from one vehicle to another at intermediate points. Kim and Schonfeld (2014) integrated conventional and flexible bus services considering uncertain arrival times. The objective is to optimize the service type selection, vehicle size, headways, slack times, and the number of zones for different types of services. The DAR problems with manpower requirements, which are motivated by the preference of companion staff members on the vehicle, address jointly the traditional DAR problem and staff scheduling. Braekers and Kovacs (2016) presented a multi-period dial-a-ride problem considering driver consistency. Lim et al. (2017) investigated a joint multi-trip pickup and delivery and staff scheduling problem for real-life healthcare.

With the advent of transportation electrification and automatic driving, the DAR problems in this context have attracted increasing attention. Pimenta et al. (2017) proposed a reliability-based DAR problem with autonomous electric vehicles, to minimize the number of stops. Masmoudi et al. (2016) investigated the DAR problem considering electric vehicles and battery-swapping stations. Bongiovanni et al. (2019) presented an electric autonomous DAR problem and developed two mixed-integer linear program formulations. Li et al. (2022) considered eco-routing technology in the DAR service and developed a mix-integer model to determine eco-oriented routes and schedules.

As for the dynamic DAR problem, Huang et al. (2020) investigated the DAR service network design problem with both static and dynamic demands. Azadeh et al. (2022) proposed a dynamic DAR problem integrating passengers' choice behavior using discrete choice models and assortment optimization. There are also a handful of works that incorporate pricing in decision-making (Sayarshad and Chow, 2015; Santo and Xavier, 2015). To deal with the dynamism of the DAR problem, most research works employ the rolling horizon method (Huang et al., 2020), in which the whole period is decomposed into several static sub-problems, and adjustment is made at each decision epoch. Another category makes discrete event-based decisions each time a new riding request pops up (Wong et al., 2014; Engelen et al., 2018).

2.3. Solution approach

Once the model is developed, it is essential to design a solution approach to solve the problem to optimality effectively and efficiently, particularly for large-scale scenarios. Two types of algorithms are commonly adopted to address DAR problems: exact algorithms and heuristics. Most exact algorithms for DAR problems are developed based on the branch-and-bound framework, such as the branch-and-price algorithm (Parragh et al., 2014), branch-and-cut algorithm (Cordeau, 2006; Bongiovanni et al., 2019), and branch-and-price-and-cut algorithm (Luo et al., 2019). Although the exact algorithms can guarantee the solution optimality, the computational time can be very long, which is unaffordable for online fashion with a large number of requests. Thus, the exact algorithms can be only applied to the static DAR problem of limited-sized instances.

In comparison, heuristics can solve larger instances in a reasonable time, though the solution optimality cannot be guaranteed. Therefore, heuristics can be applied to both static and dynamic DAR problems. Jaw et al. (1986) were among the first to propose an efficient insertion heuristic to address the dynamic DAR problem. Since it is simple and easy to implement, the construction insertion heuristics have gained wide popularity and many advancements have been made (e.g., Luo and Schonfeld, 2007; Häme, 2011). However, given its greedy properties (e.g., assigning a request to the closest vehicle), this method is sub-optimal since it fails to consider the future dynamics and measures of the whole system. Another line of research concentrates on the metaheuristics, such as tabu search (Cordeau and Laporte, 2003), simulated annealing (Braekers et al., 2014), and adaptive large neighborhood search (Gschwind and Drexl, 2019).

The predict-then-optimize method is a data-driven optimization approach to dynamic scheduling problems by leveraging historical data to predict future demand and make proactive decisions. However, most relevant studies are governed by the repetitive one-period static model. For example, Chang et al (2021) and Chang et al (2023) presented predict-then-optimize frameworks for dynamic free-floating bike repositioning problems by the rolling horizon method. Deep learning models were developed to forecast trip distribution in the next rolling period.

The decision processes in ridesharing markets are dynamic and exhibit strong spatiotemporal dependencies. The MDP model is a promising option to capture the dynamics of ridesharing markets and derive policies with long-term returns under stochastic and unbalanced supply-demand environments. Qin et al. (2022) provided a comprehensive survey of MDP-based reinforcement learning studies in the ridesharing field. Yu et al. (2021) proposed an MDP model considering vehicle dispatching, relocation, and charging decisions to maximize overall profit. Zhu et al. (2021) formulated dynamic ridesharing systems as a mean-field MDP model to maximize revenue and service rate, and developed a representative-agent reinforcement learning algorithm with significant computational and performance advantages. Gao et al. (2024) proposed the dynamic policy of the MDP model under two rolling horizon heuristic methods for the stochastic ridesharing problem. Guo et al. (2021) developed a robust optimization version of the matching-integrated vehicle rebalancing model, generating superior policies by incorporating demand uncertainty. Huang et al. (2023) modeled online ridesharing as MDP and proposed a multi-agent hierarchical reinforcement learning approach to achieve efficient large-scale fleet management.

Approximate dynamic programming (ADP) is an approach based on a Markov decision process that efficiently aggregates information model realizations, future changes in decisions, and their interactions, which has been widely applied in solving stochastic problems for logistics and transportation (Powell et al., 2012; Soeffker et al., 2022). Novoa and Storer (2009) studied ADP algorithms for vehicle route problems (VRP) with stochastic demands. Çimen and Soysal (2017) addressed the time-dependent capacitated VRP with random vehicle speed and environment, and they showed that the solution based on ADP is more environmentally friendly. Deng and Santos (2022) proposed an ADP method for aircraft maintenance scheduling, which considers uncertain aircraft daily use and maintenance inspection time. Nguyen and Chow (2023) developed an optimization framework for dynamic rail transit network operations using ADP. Koch and Klein (2020) presented dynamic pricing for the attended home delivery services model. Their ADP approach outperformed benchmark heuristics in terms of both profit and the number of customers served.

The scenario approach is a general data-driven decision-making methodology that can deal with uncertainties of the system by sampling a set of representative scenarios. Bent and Van Hentenryck (2004) introduced a multiple-scenario approach to the dynamic vehicle routing problem, considering time windows and unpredictable customers. This approach systematically integrates future scenarios into the planning process. Hvattum et al. (2006) formulated a multistage stochastic model to address the dynamic stochastic vehicle routing problem. The authors proposed a dynamic stochastic hedging heuristic based on sample scenarios for problem-solving. Ghilas et al. (2016) developed a scenario-based sample average approximation approach to the dynamic and stochastic pickup and delivery problem, considering fixed scheduled lines and stochastic demand. Li et al. (2019) developed a metaheuristic algorithm to schedule a fleet of vans to dynamically serve passengers. The algorithm uses multiple scenarios to represent diverse realizations of stochastic requests. Gong et al. (2021) presented an integer nonlinear programming model for the train timetable optimization problem with the scenario-based representation of passenger distribution.

3. Preliminaries

3.1. Benefits and challenges of DAR operation with demand prediction

Based on the travel demand (i.e., origin and destination locations, time windows) provided by customers, the platform assigns vehicles to serve the customers. Most existing works on DAR problems assign vehicles only to currently known requests, overlooking the importance of unknown future requests that can improve their utilization. To illustrate, let us consider a vehicle and two requests (request 1 and request 2). Request 1's presence is known but far from the vehicle. Request 2 is located between the vehicle and request 1 but is unknown currently. When demand prediction is not considered, the platform would reject request 1 due to high costs. If request 2's presence can be anticipated, the plan that serves request 2 first and request 1 later will be more profitable. This toy example suggests that offering demand prediction contributes to making more informed decisions.

However, offering a perfect prediction is particularly difficult because prediction errors can exist in the future demand due to its stochastic attributes. The outputs of the prediction models (e.g., deep learning models) are also variable due to unavoidable learning errors. The prediction errors will accumulate gradually over the entire planning horizon if not well addressed, which can significantly mislead the problem-solving. To warrant the benefits of demand prediction, it is imperative to develop a prediction error correction mechanism to eliminate prediction errors and rectify the policies promptly.

3.2. Processing of DAR operation and future demand

To handle the high dynamism of the DAR problem, a rolling horizon method is employed to transform the problem into a series of interdependent sub-problems. This method has a higher matching quality than immediate matching (Yang et al., 2020). The entire planning horizon of the DAR service is divided into $|P|$ optimization periods with an interval of T. As shown in Fig. 1(a), at the start time of each period, a new batch of actual demand is known, and the platform re-plans for both the actual demand and predicted future demand; subsequently, the requests submitted during this period (from start time to end time) are stored into the request pool. For instance, in period 2, the platform plans for requests submitted during the time slot $[0, T]$, which have been already stored in the request pool, together with potential requests that may occur during the time slot $[T, 2T]$.

The DAR service is optimized by periodically solving a MDP model with an adaptive lookup table. As shown in Fig.1 (b), in each period, we consider serving actual and future requests as a sequence of events. Since each request includes both boarding and alighting services, an event can be regarded as a decision epoch that dispatches a vehicle to a specific node to pick up or drop off passengers (Note that different requests may be served at a single node simultaneously). We conduct an offline simulation of anticipatory decision-making within a short computational time, t_B, to update the lookup table of the MDP model. At the time $(p-1)T + t_B$, the optimized policy based on the updated lookup table maps a series of vehicle dispatching decisions for the current period. Subsequently, the corresponding vehicle executes the decisions until new actual demand is received, i.e., the start time of the next period. In each period, the optimization of both types of demands is based on the system state at the start time of that period, which is determined by the execution status of the decisions of the previous period.

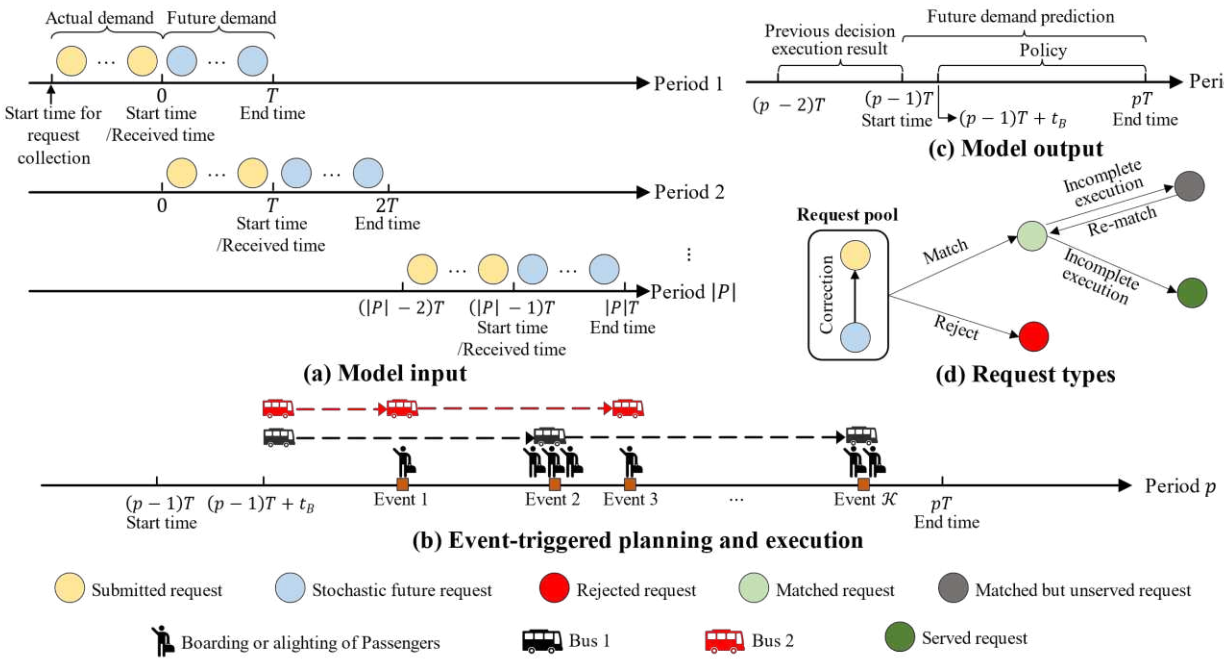


Fig. 1 Rolling horizon planning framework

The model outputs the decision execution result of the previous period, the future demand, and the policy for the current period (see Fig. 1(c)). The platform processes multiple types of requests during the DAR operation. As shown in Fig. 1(d), the uncertainty information of stochastic future requests will be updated to the actual information after receiving a new batch of actual demand. The assignment result of each request is either matching or rejection, which will be informed to the passenger after the computational time. Matched requests can be categorized as served requests and matched but unserved requests, depending on the status of the corresponding vehicle executing the decisions. The matched but unserved requests are transferred to the request pool and retained until being assigned to the same vehicle again in the next period, which are not allowed to be rejected.

3.3. Prediction-failure-risk-aware optimization framework

The optimization of the DAR service is formulated as a MDP model with the objective to reduce total system cost, including fixed transportation cost, variable transportation cost, and penalty cost. To capture the spatiotemporal characteristics of the demand, we utilize deep quantile regression and Copula methods to predict stochastic future demand. This also enables us to generate multiple future demand scenarios and corresponding probabilities to explore various possibilities of future demand (see Section 4.3). In order to promptly rectify the prediction errors, in each period, we re-predict the future demand after updating the historical request data and correcting the policy. Additionally, to expedite the solution efficiency, we derive several pruning strategies to effectively reduce the decision space and computation time (see Section 4.4).

The overall optimization framework is presented in Fig. 2. At the start time of each optimization period, the platform validates the correctness of future demand predicted in the previous period based on the relationship between the received actual demand and previous decision execution result, and if necessary, uses the correction mechanism to handle their prediction errors. Then, the platform predicts future demand utilizing historical requests. The platform performs two types of operations, with and without demand prediction, depending on whether demand prediction is considered. The operation with demand prediction embeds the future demand into the MDP model and the optimal policy is identified using the scenario approach. The optimal policy for operation with demand prediction is derived by solving the model using the ADP algorithm and the scenario approach with pruning strategies. Upon receiving new actual demand, the platform updates the state variable through a periodic state transition function, which takes into account the policy, the executed decision set, and the over-period time, thereby rolling the time to the next optimization period.

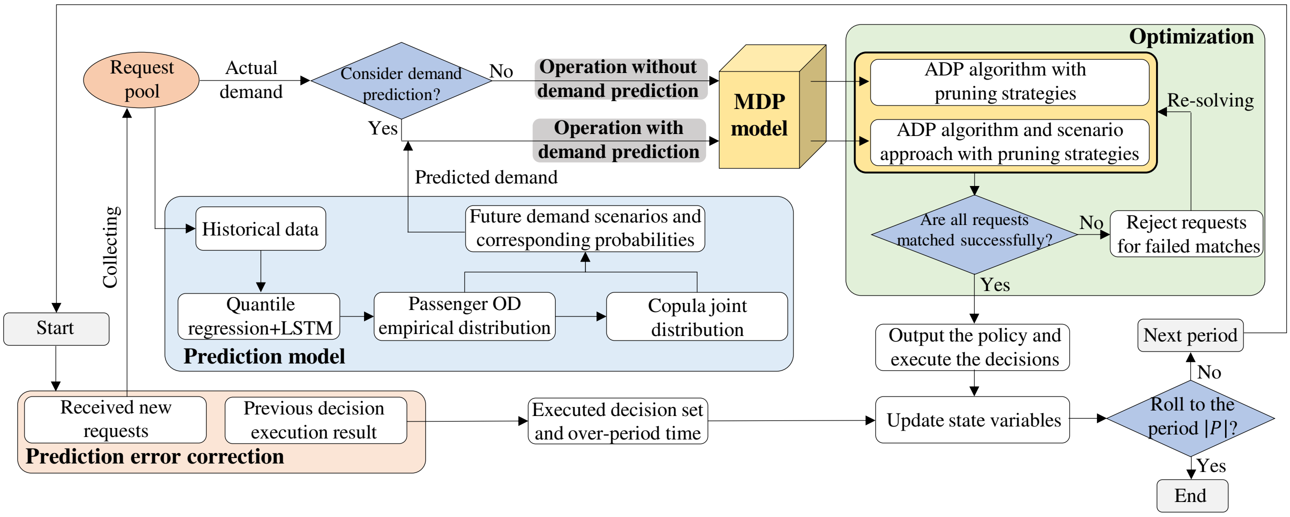

Request pool
Actual demand
Consider demand prediction?
No
Operation without demand prediction
Yes
Operation with demand prediction
MDP model
Optimization
ADP algorithm with pruning strategies
ADP algorithm and scenario approach with pruning strategies
Re-solving
Are all requests matched successfully?
No
Reject requests for failed matches
Yes
Predicted demand
Collecting
Historical data
Future demand scenarios and corresponding probabilities
Quantile regression+LSTM
Passenger OD empirical distribution
Copula joint distribution
Prediction model
Start
Output the policy and execute the decisions
Next period
No
Received new requests
Previous decision execution result
Prediction error correction
Executed decision set and over-period time
Update state variables
Roll to the period $|P|$?
Yes
End

Fig. 2 Overall optimization framework

4. Model formulation

4.1. Problem description, assumptions, and notation

The DAR service area is modelled as a directed network $G(S,\mathcal{A})$. $S = Z \cup B$ denotes the set of nodes, where $Z = \{z|z = 1,2,3, \cdots |Z|\}$ and $B = \{b|b = |Z| + 1, |Z| + 2, \cdots, |Z| + |B|\}$ represent the set of depots and physical bus stops, respectively. Each node can be used as a pick-up stop and drop-off stop. $V = \{v|v = 1,2, \dots, |V|\}$ denotes the set of vehicles. $\mathcal{A} = \{(i,j)|i,j \in S\}$ denotes the set of arcs between nodes in S. The service provider pre-collects static requests before operation (in period 1), and will receive dynamic requests during the service process (from period 2 to period $|P|$). Let $R = \{r|r = 1,2, \dots, |R|\}$ denote the set of riding requests. Each request r includes the information of desired pick-up stop $o_r \in S$ and drop-off stop $d_r \in S$, soft pick-up time window $[ET_r, LT_r]$, and the number of passengers c_r. The requests can be rejected to avoid over-allocation of fleet resources and ensure the economical viability of the DAR operation, and can be cancelled by passengers at any time after submission.

To simplify the model formulation, the following assumptions are made:

(1) The passengers of each request can wait arbitrarily long but cannot exceed a threshold $\tilde{T}$. A request must be served by only one vehicle, and the maximum detour time for each request is taken into account.

(2) The vehicle can be repeatedly dispatched but only once for each period, and it has a maximum working time for each trip. A trip represents a vehicle departing from the depot, providing service at multiple stops for boarding and alighting, and finally returning to the depot.

(3) When there are no passengers in the vehicle, it is allowed to hold at any node, including the physical bus stops and depot.

We summarize the primary notation in this study in Table 1.

Table 1 Primary notation

Sets	Description
P	The set of optimization periods, i.e., $P = \{p|p = 1,2, \dots, |P|\}$
K^p	The set of decision epochs of the MDP model in period p, $K^p = \{k^p|k^p = 1,2, \dots, |K^p|\}, p \in P$. To ease the representation, the notation k^p and $|K^p|$ is represented by k and $\mathcal{K}$, respectively.
R	The set of riding requests, $R = \{r|r = 1,2, \dots, |R|\}$
Z	The set of depots, $Z = \{z|z = 1,2,3, \cdots, |Z|\}$
B	The set of bus stops, $B = \{b|b = |Z| + 1, |Z| + 2, \cdots, |Z| + |B|\}$

S	The set of nodes, $S = Z \cup B = \{s \mid s = 1, \dots, \lvert Z \rvert + \lvert B \rvert\}$
$\mathcal{A}$	The set of arcs, $\mathcal{A} = \{(i,j) \mid \forall i, j \in S, i \neq j\}$, where node i or j represents either a depot or a bus stop
V	The set of vehicles, $V = \{v \mid v = 1,2, \dots, \lvert V \rvert\}$
L	The set of quantile levels, $L = \{l \mid l = 1,2, \dots, \lvert L \rvert\}$
$\mathcal{L}$	The set of iterations, $\mathcal{L} = \{\ell \mid \ell = 1, \dots, \lvert \mathcal{L} \rvert\}$
N	The set of future demand scenarios, $N = \{n \mid n = 1, \dots, \lvert N \rvert\}$
Auxiliary variables	
τ_l	The quantile value, which takes $\tau_l = \{5\%, 25\%, 50\%, 75\%, 95\%\}$ in this study
m_v	The depot to which the vehicle v belongs to
q_r^p	$q_r^p = 1$, if request r is received by the scheduling platform in period p; $q_r^p = 0$, otherwise
q_{ij}^p	The number of passengers from node $i \in S$ to node $j \in S$ of actual demand for period p, $\forall i \neq j$
$\tilde{q}_{ij}^p$	The predicted number of passengers from node $i \in S$ to node $j \in S$ of future demand for period p, which has been predicted at period $p - 1$, $\forall i \neq j$
$\tilde{q}_{ij}^{p(\tau_l)}$	The predicted number of passengers from node $i \in S$ to node $j \in S$ of future demand for period p when the quantile takes $\tau_l, \forall i \neq j$
Q_n^p	The future demand scenario n for period p, i.e., $Q_n^p = \left[\tilde{q}_{11}^p, \dots, \tilde{q}_{\lvert S \rvert \lvert S \rvert}^p\right]$
$F\left(q_{ij}^p\right)$	The empirical distribution function of the demand from node $i \in S$ to node $j \in S$
O_{kv}^p	The set of pick-up stops served by vehicle v in the current trip at decision epoch k of period p
D_{kv}^p	The set of drop-off stops corresponding to O_{kv}^p
$\widetilde{\Theta}_v^p$	The dispatching decision set for vehicle v of period p
State variables	
s_k^p	State variables at decision epoch k of period p
o_{kr}^p	$o_{kr}^p = 1$, if passengers of request r are on-board at decision epoch k of period p; $o_{kr}^p = 0$, otherwise
d_{kr}^p	$d_{kr}^p = 1$, if passengers of request r are arriving at decision epoch k of period p; $d_{kr}^p = 0$, otherwise
h_{kv}^p	The location of vehicle v at decision epoch k of period p
c_{kv}^p	The remaining capacity of vehicle v at decision epoch k of period p
t_{kv}^p	The accumulated travel time of vehicle v at decision epoch k of period p
u_{kv}^p	The driver's accumulated working time of vehicle v at decision epoch k of period p
Decision variables	
a_k^p	The decision variables of vehicle v at decision epoch k of period p
x_{ksvr}^p	Binary variable. $x_{ksvr}^p = 1$, if vehicle v is dispatched from the location h_{kv}^p to node s to serve request r at decision epoch k of period p; $x_{ksvr}^p = 0$, otherwise
λ_{ksvr}^p	The holding time of vehicle v at node s when serving request r at decision epoch k of period p
Exogenous information and transition function	
g_k^p	The exogenous information at decision epoch k of period p
G^p	The exogenous information of period p, $G^p = \left(\hat{t}_v^p, \Theta_v^p\right)_{\forall v \in V}$, where $\hat{t}_v^p$ and Θ_v^p is the over-period time and the executed decision set for vehicle v of period p, respectively.
$S^k\left(s_k^p, a_k^p, g_k^p\right)$	State transition function that describes the change of each attribute in s_k^p at decision epoch k
$S^p\left(s_1^p, \widetilde{\Theta}_v^p, G^p\right)$	Periodic state transition function that describes the change of each attribute in s_1^p for period p
Algorithmic variables	
$\bar{V}^\ell\left(s_k^p\right)$	The value function of the state s_k^p after ℓ-th iteration
$\bar{V}_{\pi^{MS}}^\ell\left(s_k^p\right)$	The value function of the state s_k^p after ℓ-th iteration of the multi-scenario exploration method
$\hat{v}^\ell\left(s_k^p\right)$	The unbiased sample estimate of the value of the state s_k^p at ℓ-th iteration
α_ℓ	The stepsize of ADP at ℓ-th iteration

Model parameters	
T	The time interval of each optimization period
$\tilde{T}$	The maximum allowable late time
o_r	The pick-up stop of riding request r
d_r	The drop-off stop of riding request r
c_r	The number of passengers of riding request r
ET_r	The lower bound of the time window of riding request r
LT_r	The upper bound of the time window of riding request r
$d_{(i,j)}$	The shortest distance between node i and node $j, \forall i,j \in S, \forall i \neq j,$
ε	The detour time coefficient, which represents the maximum ratio between the in-vehicle travel time and the shortest travel time of each request
$\mathscr{s}$	The vehicle's travel speed
λ	The discount factor of ADP
$c_{\max}$	The vehicle capacity
$t_{\max}$	The maximum working time for one trip of the vehicle
e	Average boarding and alighting time per person
β_f	Fixed transportation cost per vehicle trip
β_v	Unit mileage cost
β_e	Unit early arrival penalty cost
β_l	Unit late arrival penalty cost

4.2. Markov decision process for DAR optimization

We present a mathematical model of the DAR problem as a Markov decision process, whose components are defined as follows.

4.2.1. State variable

As mentioned above, in each event, the platform will make a vehicle dispatching decision. Since a vehicle can be re-dispatched only when it completes its decision, the next decision epoch is triggered by one of the following cases: (a) if the vehicle is en-route heading to node s, the next decision epoch occurs after the vehicle serves assigned request r (when s is the bus stop) or arrives (when s is the depot), (b) if the vehicle is holding at bus stop or depot, the next decision epoch occurs after the vehicle serves assigned request r.

During each optimization period, the platform makes several decisions, along with attribute changes of requests and vehicles in decision execution. To ensure that the necessary information is available before a decision is made, we define s_k^p as the state variable at decision epoch k of period p. This state variable consists of the request state $\left(o_{kr}^p, d_{kr}^p\right)_{r\in R}$ and the vehicle state $\left(h_{kv}^p, c_{kv}^p, t_{kv}^p, u_{kv}^p\right)_{\forall v\in V}$.

$$s_k^p = \left(\left(o_{kr}^p, d_{kr}^p\right)_{\forall r\in R}, \left(h_{kv}^p, c_{kv}^p, t_{kv}^p, u_{kv}^p\right)_{\forall v\in V}\right) \tag{1}$$

Since decision-making is terminated once all requests have been assigned, the number of decision epochs $\mathcal{K}$ of each period is associated with the current requests and future requests. In period 1, since the requests have not yet been served, the value of $\mathcal{K}$ is the number of pick-up stops plus the number of drop-off stops of the requests of this period. Note that the $\mathcal{K}$ is an induced variable from period 2 to period $|P|$, which is related to the number of unserved requests to date. Based on the request state $\left(o_{kr}^p, d_{kr}^p\right)_{r\in R}$, the value of $\mathcal{K}$ can be calculated as follows:

$$\mathcal{K} = |K^p| = \begin{cases} \sum\limits_{r\in R} 2q_r^p, \text{if } p = 1 \\ \sum\limits_{\wp=1}^{p} \sum\limits_{k\in K^p} \sum\limits_{r\in R} q_r^{\wp}\left[\left(1 - o_{kr}^{\wp}\right) + \left(1 - d_{kr}^{\wp}\right)\right], \forall p \in P \setminus \{1\} \end{cases} \tag{2}$$

4.2.2. Decision variable

Since vehicle holding can increase the flexibility of DAR service, the departure time at the depot and holding time at bus stops are introduced into the decisions. The decision a_k^p at decision epoch k of period p is represented by Eq. (3).

$$a_k^p = \left(\left(x_{ksvr}^p, \lambda_{ksvr}^p\right)_{\forall s \in S, v \in V, r \in R}\right) \tag{3}$$

Although optimizing the holding time exclusively by discretization can enhance exploration accuracy, it poses another modeling challenge in that the decision space will be significantly enlarged. Specifically, at decision epoch k of period p, x_{ksvr}^p is initially a three-dimensional decision variable, but it will become a more complex four-dimensional decision variable if the holding time is determined using the discretization method.

Given that x_{ksvr}^p has already assigned vehicle v to request r at node s, we can calculate the minimum holding time directly based on ET_r and the vehicle's arrival time at this node, without additionally exploring the current holding time. To reduce the decision space and efficiently represent the decisions, we consider λ_{ksvr}^p as an exogenous decision variable of x_{ksvr}^p and the optimal value of λ_{ksvr}^p can be calculated by Eq. (4).

$$\lambda_{ksvr}^p = \begin{cases} \max\left\{x_{ksvr}^p\left(ET_r - t_{kv}^p - \dfrac{d_{(\mathscr{m}_v, s)}}{\mathscr{s}}\right), 0\right\}, \text{if } s = h_{kv}^p = \mathscr{m}_v, \forall r \in R, \forall v \in V, \forall k \in K^p \\ \qquad , \forall p \in P \\ \max\left\{x_{ksvr}^p\left(ET_r - t_{kv}^p - \dfrac{d_{(h_{kv}^p, s)}}{\mathscr{s}}\right), 0\right\}, \text{if } c_{kv}^p = c_{\max} \text{ and } h_{kv}^p \neq \mathscr{m}_v, \forall s \in S \setminus \{\mathscr{m}_v\} \\ \qquad , \forall r \in R, \forall v \in V, \forall k \in K^p, \forall p \in P \\ 0, \text{if } c_{kv}^p \neq c_{\max} \text{ and } h_{kv}^p \neq \mathscr{m}_v, \forall s \in S \setminus \{\mathscr{m}_v\}, \forall r \in R, \forall v \in V, \forall k \in K^p, \forall p \in P \end{cases} \tag{4}$$

When the vehicle is not dispatched ($x_{ksvr}^p = 0$), its holding time is 0. The dispatched vehicles can be classified into in-depot vehicles ($h_{kv}^p = \mathscr{m}_v$) and out-depot vehicles ($h_{kv}^p \neq \mathscr{m}_v$), depending on their current locations. For in-depot vehicles, if the vehicle arrives earlier than the specified earliest pick-up time ET_r, it is more cost-effective to hold the vehicle in the depot to reduce the driver's working time. Thus, the node s at which the vehicle arrives at this decision epoch is still depot $\mathscr{m}_v$, and the optimal holding time is $\max\left\{ET_r - t_{kv}^p - \frac{d_{(\mathscr{m}_v, s)}}{\mathscr{s}}, 0\right\}$. For out-depot vehicles, if there are still onboard passengers before heading to the pick-up stop, it is undesirable to hold the vehicle and onboard passengers at the stop. For this reason, we impose the following restrictions, i.e., if there are no on-board passengers ($c_{kv}^p = c_{\max}$), then the vehicle can go to any pick-up stop and the optimal holding time can be calculated similarly. Otherwise, if there exist onboard passengers ($c_{kv}^p \neq c_{\max}$), then the vehicle is only allowed to go to the pick-up stop where the vehicle can arrive within the soft pick-up time window.

4.2.3. Constraints

The feasible decision space is determined by the following constraints:

$$\sum_{p \in P} \sum_{r \in R} q_r^p = |R| \tag{5}$$

$$\sum_{r \in R} \left(o_{\mathcal{K}r}^{|P|} + d_{\mathcal{K}r}^{|P|}\right) = 2|R| \tag{6}$$

$$q_r^p (o_{\mathcal{K}r}^p + d_{\mathcal{K}r}^p) = 2, \forall r \in R, \forall p \in P \tag{7}$$

$$0 \leq \sum_{k \in K^p} \sum_{v \in V} x_{ksvr}^p \leq 1, \text{if } s \in \{o_r, d_r\}, \forall r \in R, \forall p \in P \tag{8}$$

$$0 \leq \sum_{k \in K^p} \sum_{v \in V} x_{kd_r vr}^p t_{k+1,v}^p - \sum_{k \in K^p} \sum_{v \in V} x_{ko_r vr}^p t_{k+1,v}^p \leq \varepsilon \frac{d_{(o_r, d_r)}}{\mathscr{s}}, \forall r \in R, \forall p \in P \tag{9}$$

$$0 \leq u_{kv}^p \leq t_{\max}, \forall v \in V, \forall k \in K^p, \forall p \in P \tag{10}$$

$$0 \leq c_{kv}^{p} - \sum_{s \in S} \sum_{r \in R} x_{ksvr}^{p} c_r \leq c_{\max}, \forall v \in V, \forall k \in K^{p}, \forall p \in P \tag{11}$$

$$ET_r \leq t_{kv}^{p} + x_{ksvr}^{p} \frac{\mathscr{d}_{(h_{kv}^{p}, s)}}{\mathscr{s}} + \lambda_{ksvr}^{p} \leq LT_r + \tilde{T}, \text{if } s = o_r, \forall v \in V, \forall r \in R, k \in K^{P}, p \in P \tag{12}$$

$$0 \leq \hat{t}_{v}^{p} \leq T, \forall v \in V, \forall p \in P \tag{13}$$

$$x_{ksvr}^{p} \in \{0,1\}, \forall s \in S, \forall v \in V, \forall r \in R, \forall k \in K^{p}, \forall p \in P \tag{14}$$

Constraints (5) ensure that all riding requests should be received by the platform. Constraints (6) ensure that each request has been either served or rejected at the end of period $|P|$. Constraints (7) ensure that the received requests are either matched or rejected in each period. Constraints (8) ensure that each request is served by only one vehicle for boarding and alighting. Constraints (9) state that the travel time of each request is no more than ε times the shortest travel time from its pick-up stop to its drop-off stop. Constraints (10) state that the driver has a maximum working time for each trip. Constraints (11) ensure that the assigned requests do not violate the remaining vehicle capacity. Constraints (12) enforce that the vehicle arrives at the pick-up stop within the soft time window of each request. Constraints (13) define the duration of the over-period time (see Section 4.2.4.2). Constraints (14) specify that the decision variable is binary.

4.2.4. Transition function

4.2.4.1. State transition function

The transition from the pre-decision state s_k^p to the next pre-decision state s_{k+1}^p is governed by the function $S^k(\ \ )$, which reflects the attribute changes of the state variable based on the selected decisions. Once the decision a_k^p is executed, the new exogenous information g_k^p is generated, including the shortest distance $\mathscr{d}_{(h_{kv}^{p}, s)}$ from the vehicle location to the decision node, the number of passengers for boarding or alighting, boarding time, and alighting time. Using this new information, the next pre-decision state can be obtained, denoted as:

$$s_{k+1}^{p} = S^{k}\left(s_k^p, a_k^p, g_k^p\right) \tag{15}$$

In this study, since the choice of a decision determines the subsequent state with certainty, we model the transition probability function deterministically as in Eq. (16). To simplify notation, we will omit $Pr\left(s_{k+1}^{p} | s_k^p, a_k^p\right)$ in subsequent equations.

$$Pr\left(s_{k+1}^{p} | s_k^p, a_k^p\right) = \begin{cases} 1, \text{if } S^{k}\left(s_k^p, a_k^p, g_k^p\right) = s_{k+1}^{p} \\ 0, \text{if } S^{k}\left(s_k^p, a_k^p, g_k^p\right) \neq s_{k+1}^{p} \end{cases} \tag{16}$$

The attribute changes of the state variables under different decision variables are as follows:

$$o_{k+1,r}^{p} = o_{kr}^{p} + \sum_{v \in V} x_{ksvr}^{p}, \text{if } s = o_r, \forall r \in R, \forall k \in K^{p}, \forall p \in P \tag{17}$$

$$d_{k+1,r}^{p} = d_{kr}^{p} + \sum_{v \in V} x_{ksvr}^{p}, \text{if } s = d_r, \forall r \in R, \forall k \in K^{p}, \forall p \in P \tag{18}$$

$$h_{k+1,v}^{p} = \begin{cases} s, \text{if } \sum_{r \in R} x_{ksvr}^{p} = 1, \forall s \in S \setminus \{\mathscr{m}_v\}, \forall v \in V, \forall k \in K^{p}, \forall p \in P \\ \mathscr{m}_v, \text{if } \sum_{r \in R} x_{k\mathscr{m}_v vr}^{p} = 1, \forall v \in V, \forall k \in K^{p}, \forall p \in P \\ h_{kv}^{p}, \text{otherwise} \end{cases} \tag{19}$$

$$c_{k+1,v}^{p}=\begin{cases}c_{kv}^{p}-\sum_{r\in R}x_{ko_{r}vr}^{p}c_{r}+\sum_{r\in R}x_{kd_{r}vr}^{p}c_{r}\text{, if }\sum_{r\in R}x_{km_{v}vr}^{p}\neq 1,\forall v\in V,\forall k\in K^{p},\forall p\in P\\ c_{\max}\text{, if }\sum_{r\in R}x_{km_{v}vr}^{p}=1,\forall v\in V,\forall k\in K^{p},\forall p\in P\end{cases}\tag{20}$$

$$t_{k+1,v}^{p}=t_{kv}^{p}+\sum_{s\in S}\sum_{r\in R}x_{ksvr}^{p}\frac{d_{(h_{kv}^{p},s)}}{\mathscr{s}}+\sum_{s\in S}\sum_{r\in R}\lambda_{ksvr}^{p}+e\sum_{r\in R}\sum_{s\in S}x_{ksvr}^{p}c_{r}\text{, }\forall v\in V,\forall k\in K^{p},\forall p\in P\tag{21}$$

$$u_{k+1,v}^{p}=\begin{cases}u_{kv}^{p}+\sum_{s\in S}\sum_{r\in R}x_{ksvr}^{p}\frac{d_{(h_{kv}^{p},s)}}{\mathscr{s}}+\sum_{s\in S}\sum_{r\in R}\lambda_{ksvr}^{p}+e\sum_{r\in R}\sum_{s\in S}x_{ksvr}^{p}c_{r}\\ \qquad\text{, if }\sum_{r\in R}x_{km_{v}vr}^{p}\neq 1,\forall v\in V,\forall k\in K^{p},\forall p\in P\\ 0\text{, if }\sum_{r\in R}x_{km_{v}vr}^{p}=1,\forall v\in V,\forall k\in K^{p},\forall p\in P\end{cases}\tag{22}$$

In Eqs. (17) and (18), o_{kr}^{p} and d_{kr}^{p} are updated once the passengers of request r board at the pick-up stop and alight at the drop-off stop, respectively. Eq. (19) specifies the location of vehicle v. In Eq. (20), when the decision node s is a bus stop, the remaining capacity of vehicle v decreases by the number of passengers boarding, or increases by the number of passengers alighting of request r; when decision node s is a depot, the capacity becomes full. In Eqs. (21) and (22), when the decision node s is a bus stop, the accumulated travel time and driver's working time of vehicle v increase by the travel time between the location and node s, holding time, and service time. The current trip ends after the vehicle is pulled into its depot, and the driver's working time is reset to 0.

4.2.4.2. Periodic state transition function

The new batch of riding requests is received periodically under the rolling horizon framework, which poses another challenge in that, the final decision epoch $\mathcal{K}$ can exceed the end time of the current period when the number of requests surges. In other words, the decisions cannot be fully executed by vehicles before the platform receives new actual demand, resulting in the mismatch between the expected state of optimization and the actual state after execution. Therefore, $S^{p}(\quad)$ is needed to govern the transition of states between adjacent periods.

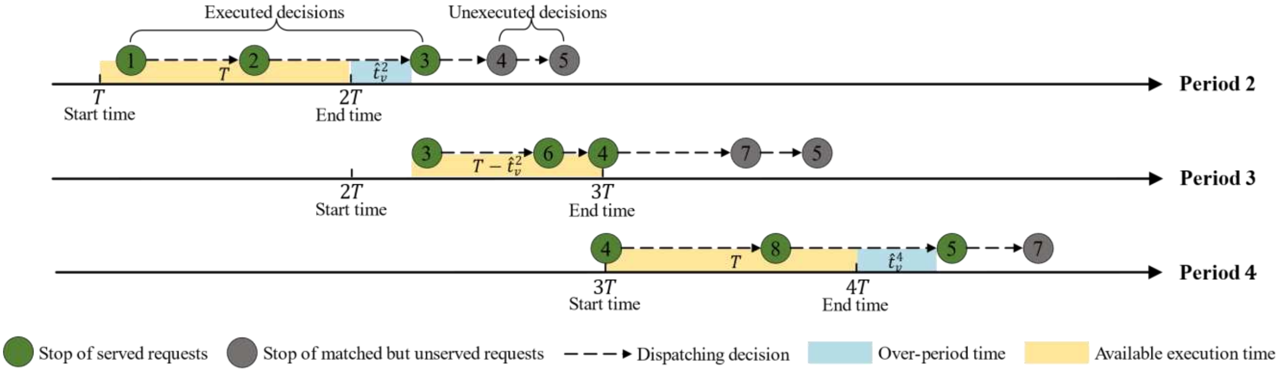


Fig. 3 Illustration of the over-period time

For vehicles executing a decision at the start time of the next period, if they have not reached the decision node s that triggers the next decision epoch, the state variable is unable to transition. To address this issue, we introduce the over-period time $\hat{t}_{v}^{p}$. Fig. 3 illustrates the decision execution process for the vehicle v from period 2 to period 4. For instance, at the start time of period 2, the vehicle executes the decision heading to stop 3. If the over-period time is not considered, the decision cannot be executed effectively. A plausible way is to continue executing the decision, such that the original executed decision set of period 2 is changed from $\Theta_{v}^{2}=1\rightarrow 2$ to $\Theta_{v}^{2}=1\rightarrow 2\rightarrow$

3. The extra time spent is recorded as the over-period time $\hat{t}_v^2$, and the available execution time is changed from the original T to $T - \hat{t}_v^2$ in period 3. Similarly, the executed decision set in period 4 is $\Theta_v^4 = 4 \rightarrow 8 \rightarrow 5$, with the over-period time being $\hat{t}_v^4$.

Based on the received time of new actual demand and the previous decision execution results, we can derive the exogenous information $G^p = \left(\hat{t}_v^p, \Theta_v^p\right)_{\forall v \in V}$ between periods. Specifically, the decision immediately before the available execution time $T - \hat{t}_v^p$ is consumed up is first identified, which is added to Θ_v^p together with decisions that have been executed in the current period, thereby constituting the executed decision set Θ_v^{p+1}. The state variables of the next period s_1^{p+1}can be calculated by the change of s_1^p resulting from Θ_v^{p+1}. The pseudo-code of the periodic state transition function is presented in Table 2.

Table 2 The pseudo-code of the periodic state transition function

Algorithm 1: Periodic state transition function $\boldsymbol{S^p\left(s_1^p, \widetilde{\Theta}_v^p, G^p\right)}$

Input: Initial state variable s_1^p, dispatching decision set $\widetilde{\Theta}_v^p$ of period p, and exogenous information G^p

Output: Initial state variable $s_1^{p+1}$and executed decision set Θ_v^{p+1} of the next period

```
Initialize the variables: $\Delta t_v^{p+1} \leftarrow 0$; $\Theta_v^{p+1} \leftarrow \Theta_v^p$
While decision variable $a_k^p$ s.t. $a_k^p \in \widetilde{\Theta}_v^p, \Delta t_v^{p+1} \leq T - \hat{t}_v^p, \forall v \in V$ do
    Calculate the time change : $\Delta t_v^{p+1} \leftarrow \Delta t_v^{p+1} + \sum_{s \in S} \sum_{r \in R} x_{ksvr}^p \frac{d_{\left(h_{kv}^p, s\right)}}{\mathcal{s}} + \sum_{s \in S} \sum_{r \in R} \lambda_{ksvr}^p + e \sum_{r \in R} \sum_{s \in S} x_{ksvr}^p c_r$, and add this decision $a_k^p$ to the executed decision set of vehicle $v$: $\Theta_v^{p+1} \leftarrow \Theta_v^{p+1} \cup s$
    Update the attributes in $s_1^p$: $\left(\left(o_{kr}^{p'}, d_{kr}^{p'}\right)_{\forall r \in R}, \left(h_{kv}^{p'}, c_{kv}^{p'}, t_{kv}^{p'}, u_{kv}^{p'}\right)_{\forall v \in V}\right) \leftarrow S^k\left(s_1^p, a_k^p, g_k^p\right)$
end while
if $\Delta t_v^{p+1} > T - \hat{t}_v^p$ then
    Recode the over-period time of the next period: $\hat{t}_v^{p+1} \leftarrow \Delta t_v^{p+1} - T_v^{p+1}$
else
    $\hat{t}_v^{p+1} \leftarrow 0$
end if
If $t_{kv}^{p'} < (p+1)T$ then
    $t_{kv}^{p'} \leftarrow (p+1)T$
end if
$s_1^{p+1} \leftarrow \left(\left(o_{kr}^{p'}, d_{kr}^{p'}\right)_{\forall r \in R}, \left(h_{kv}^{p'}, c_{kv}^{p'}, t_{kv}^{p'}, u_{kv}^{p'}\right)_{\forall v \in V}\right)$
```

4.2.5. Cost models

The cost components include fixed transportation cost, variable transportation cost, and penalty cost. The fixed transportation cost is related to the number of vehicle trips for each period. Since a vehicle can have at most one trip for each period (Assumption 2), the formulation is taken as follows:

$$CF^p = \beta_f \sum_{v \in V} \min\left\{\sum_{k \in K^p} \sum_{s \in S} \sum_{r \in R} x_{ksvr}^p, 1\right\}, \forall p \in P \tag{23}$$

The variable transportation cost depends on the running mileage of each vehicle dispatching decision, i.e., the distance from the dispatched vehicle location h_{kv}^p to decision node s, which can be computed as follows:

$$CV_k^p = \beta_v \sum_{s \in S} \sum_{v \in V} \sum_{r \in R} x_{ksvr}^p d_{\left(h_{kv}^p, s\right)}, \forall k \in K^p, \forall p \in P \tag{24}$$

The penalty cost includes the penalties for early arrival and late arrival. In general, late arrival is more harmful than early arrival ($\beta_l > \beta_e$). If the vehicle arrives at the pick-up stop within the specified time windows, the penalty cost is 0. Consequently, the penalty cost is taken as follows:

$$CP_k^p = \sum_{s\in S}\sum_{v\in V}\sum_{r\in R} x_{ksvr}^p \left\{\beta_e \cdot \max\left\{ET_r - \left(t_{kv}^p + \frac{d_{(h_{kv}^p,s)}}{\mathscr{s}}\right), 0\right\} + \beta_l \cdot \max\left\{(t_{kv}^p + \frac{d_{(h_{kv}^p,s)}}{\mathscr{s}}) - LT_r, 0\right\}\right\}, \forall k \in K^p, \forall p \in P \tag{25}$$

In period p, given the state s_k^p and the decision a_k^p at decision epoch k, we can calculate the newly incurred costs of post-decision by Eq. (26).

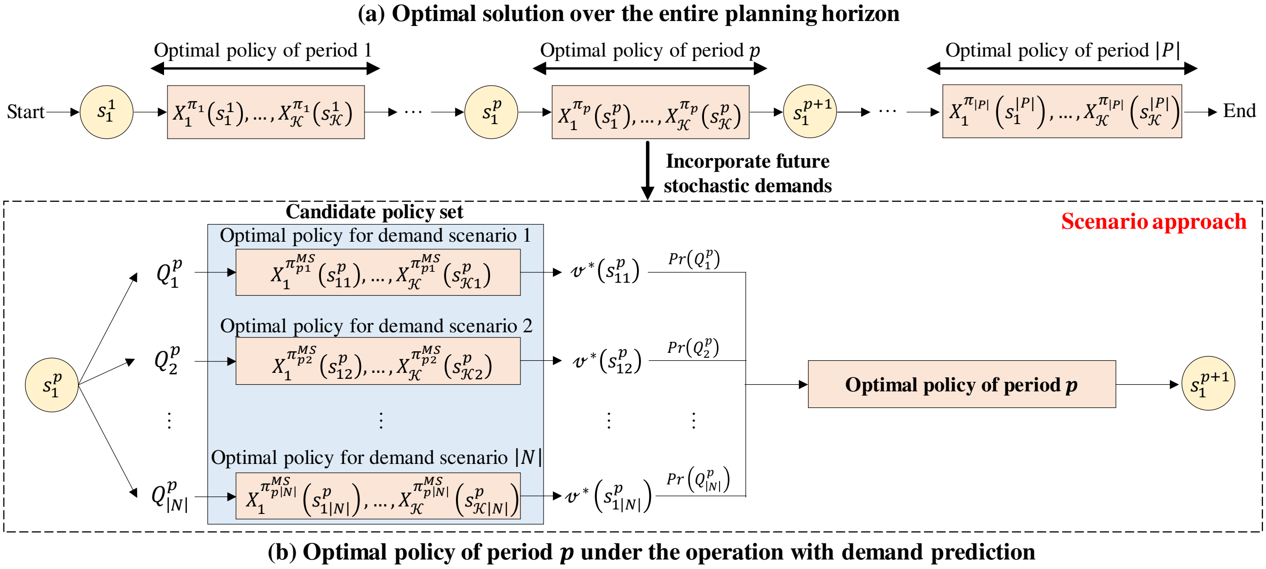
$$\mathcal{R}(s_k^p, a_k^p) = \begin{cases} CV_k^p + CP_k^p, \forall k \in K^p \setminus \{\mathcal{K}\}, \forall p \in P \\ CV_k^p + CP_k^p + CF^p, \text{if } k = \mathcal{K}, \forall p \in P \end{cases} \tag{26}$$

4.2.6. *Objective function*

The platform consistently receives a new batch of requests over time, resulting in changes in the environment. Consequently, the optimal policy found in the previous period may no longer hold optimality, so we must periodically ascertain the optimal policy for the updated environment by solving the MDP model. As illustrated in Fig. 4(a), the optimal solution over the entire planning horizon can be represented by the sequence of optimal policies $\pi = \{\pi_1, \dots, \pi_{|P|}\}$, where π_p denotes the sequence of decision rules $\{X_1^{\pi_p}(\ ), \dots, X_{\mathcal{K}}^{\pi_p}(\ )\}$ that minimizes the expected cost over the planning horizon of period p. Here, $X_k^{\pi_p}(\ )$ is a function mapping state s_k^p to decision a_k^p at decision epoch k of period p. The optimal policy sequence can be found by solving the Bellman equation for each period:

$$\mathscr{v}^*(s_1^p) = \min_{\pi_p\in\Pi_p} \mathbb{E}\left\{\sum_{k\in K^p} \mathcal{R}\left(s_k^p, X_k^{\pi_p}(s_k^p)\right) | s_1^p\right\}, \forall p \in P \tag{27}$$

where $\mathscr{v}^*(\ )$ is the optimal value function under the optimal policy for a specific state.


(a) Optimal solution over the entire planning horizon

Optimal policy of period 1
Optimal policy of period p
Optimal policy of period $|P|$
Start
s_1^1
$X_1^{\pi_1}(s_1^1), \dots, X_{\mathcal{K}}^{\pi_1}(s_{\mathcal{K}}^1)$
$\cdots$
s_1^p
$X_1^{\pi_p}(s_1^p), \dots, X_{\mathcal{K}}^{\pi_p}(s_{\mathcal{K}}^p)$
s_1^{p+1}
$\cdots$
$X_1^{\pi_{|P|}}(s_1^{|P|}), \dots, X_{\mathcal{K}}^{\pi_{|P|}}(s_{\mathcal{K}}^{|P|})$
End
Incorporate future stochastic demands
Candidate policy set
Scenario approach
Optimal policy for demand scenario 1
$X_1^{\pi_{p1}^{MS}}(s_{11}^p), \dots, X_{\mathcal{K}}^{\pi_{p1}^{MS}}(s_{\mathcal{K}1}^p)$
Optimal policy for demand scenario 2
$X_1^{\pi_{p2}^{MS}}(s_{12}^p), \dots, X_{\mathcal{K}}^{\pi_{p2}^{MS}}(s_{\mathcal{K}2}^p)$
Optimal policy for demand scenario $|N|$
$X_1^{\pi_{p|N|}^{MS}}(s_{1|N|}^p), \dots, X_{\mathcal{K}}^{\pi_{p|N|}^{MS}}(s_{\mathcal{K}|N|}^p)$
s_1^p
Q_1^p
Q_2^p
$Q_{|N|}^p$
$\mathscr{v}^*(s_{11}^p)$
$\mathscr{v}^*(s_{12}^p)$
$\mathscr{v}^*(s_{1|N|}^p)$
$Pr(Q_1^p)$
$Pr(Q_2^p)$
$Pr(Q_{|N|}^p)$
Optimal policy of period p
s_1^{p+1}

(b) Optimal policy of period *p* under the operation with demand prediction

Fig. 4 Illustration of the optimal solution

Under the operation with demand prediction, the optimal solution over the entire planning horizon, denoted as $\pi^{MS} = \{\pi_1^{MS}, \dots, \pi_{|P|}^{MS}\}$, should explicitly consider the impact of future stochastic demands. As illustrated in Fig. 4(b), we generate a scenario-based uncertain future demand set from historical information. Each scenario represents a realization of future stochastic demands sampled from the joint probability distribution of all OD pairs, where scenario n and its occurrence probability $Pr\left(Q_n^p\right)$ are known once a period starts (see Section 4.3). The scenario approach is applied to identify the optimal policy of each period, which considers the future demand within each scenario as deterministic (Li et al., 2019). This is also supported by Lian et al. (2024) who employed the scenario approach to represent possible realizations of stochastic future requests. Hence, in period p, we parallelly solve a series of deterministic optimization problems to find the optimal policy for each scenario:

$$\mathscr{v}^*\left(s_{1n}^p\right) = \min_{\pi_{pn}^{MS} \in \Pi_{pn}^{MS}} \mathbb{E}\left\{\sum_{k \in K^p} \mathcal{R}\left(s_k^p, X_k^{\pi_{pn}^{MS}}\left(s_{kn}^p\right)\right) | s_1^p, Q_n^p\right\}, \forall n \in N \tag{28}$$

All optimal policies constitute the candidate policy set, for instance, the candidate policy set of period p are $\left\{\pi_{p1}^{MS}, \ldots, \pi_{p|N|}^{MS}\right\}$, where the π_{pn}^{MS} represents the optimal policy in scenario n. We select the one policy from the set that minimizes the expected value for all scenarios as the optimal policy for the current period:

$$\pi_p^{MS} = \underset{\pi_{pn}^{MS} \in \left\{\pi_{p1}^{MS}, \ldots, \pi_{p|N|}^{MS}\right\}}{\operatorname{argmin}} \left(\sum_{n' \in N} Pr\left(Q_n^p\right) \mathscr{v}\left(\pi_{pn}^{MS} | s_1^p, Q_{n'}^p\right)\right) \tag{29}$$

where the $\mathscr{v}\left(\pi_{pn}^{MS} | s_1^p, Q_{n'}^p\right)$ represents the value function of executing policy π_{pn}^{MS} at the state s_1^p in the scenario $Q_{n'}^p$.

4.2.7. Request rejection mechanism

The passenger-to-vehicle assignment is essential to decision-making. However, the received requests cannot be always matched subject to supply limitations, such as the fleet size, vehicle capacity, and driver's working time. In this section, we propose a practical request rejection mechanism, whose specifications are described as follows.

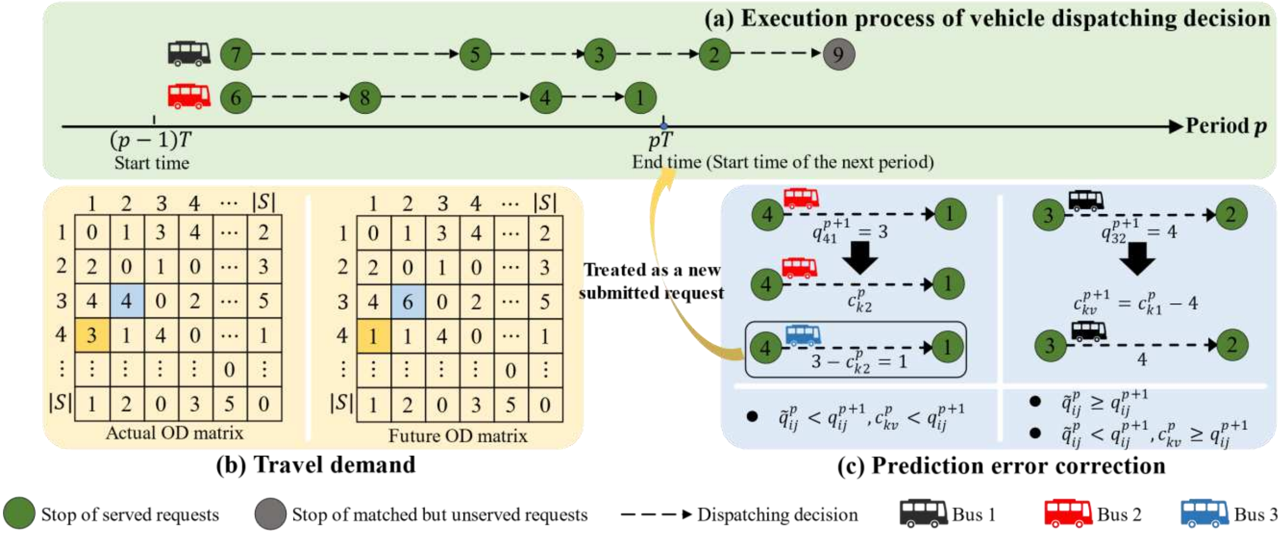
When received request r cannot be matched in period p, we have $q_r^p\left(o_{\mathcal{K}r}^p + d_{\mathcal{K}r}^p\right) \neq 2$, and the requests can be potentially rejected. We remove the request with the largest number of matching failures and then resolved the MDP model one by one until the remaining requests can be matched. Based on this principle, the pseudo-code for the request rejection mechanism is provided in Table 3.

Table 3 The pseudo-code of the request rejection mechanism

```
Algorithm 2: Request rejection mechanism
Input: State variables $s_1^p$
Output: Updated state variables $s_1^p$ after rejection, the policy of period $p$
While request $r$ s.t. $q_r^p\left(o_{\mathcal{K}r}^p + d_{\mathcal{K}r}^p\right) \neq 2$ do
  Make a vehicle dispatching decision and transition the state through $S^k$
  if all unserved requests can be matched then
    Return the policy and end while
  else
    Solve the MDP model a few times with $s_1^p$ parallelly, and record the number of matching failures request $r'$
    Update $o_{1r'}^p$ and $d_{1r'}^p$ by removing the request $r'$ of the highest matching failure: $o_{1r'}^p \leftarrow 1$; $d_{1r'}^p \leftarrow 1$
    Update and return the state variables: $s_1^p \leftarrow \left(\left(o_{1r}^p, d_{1r}^p\right)_{\forall r \in R}, \left(h_{1v}^p, c_{1v}^p, t_{1v}^p, u_{1v}^p\right)_{\forall v \in V}\right)$
  end if
end while
```

4.2.8. Prediction error correction mechanism

The prediction errors may introduce significant risks to request assignment since the decision-making module can use unreliable or erroneous predictions, resulting in vehicle detours and incomplete service. To address this issue, we propose a prediction error correction mechanism.


(a) Execution process of vehicle dispatching decision
7
5
3
2
9
6
8
4
1
Period p
$(p-1)T$
Start time
pT
End time (Start time of the next period)
Treated as a new submitted request
4
1
$q_{41}^{p+1} = 3$
c_{k2}^{p}
$3 - c_{k2}^{p} = 1$
• $\tilde{q}_{ij}^{p} < q_{ij}^{p+1}, c_{kv}^{p} < q_{ij}^{p+1}$
3
2
$q_{32}^{p+1} = 4$
$c_{kv}^{p+1} = c_{k1}^{p} - 4$
4
• $\tilde{q}_{ij}^{p} \geq q_{ij}^{p+1}$
• $\tilde{q}_{ij}^{p} < q_{ij}^{p+1}, c_{kv}^{p} \geq q_{ij}^{p+1}$
(c) Prediction error correction
● Stop of served requests
● Stop of matched but unserved requests
– – – ▸ Dispatching decision
Bus 1
Bus 2
Bus 3

	1	2	3	4	⋯	$\lvert S\rvert$
1	0	1	3	4	⋯	2
2	2	0	1	0	⋯	3
3	4	4	0	2	⋯	5
4	3	1	4	0	⋯	1
⋮	⋮	⋮	⋮	⋮	0	⋮
$\lvert S\rvert$	1	2	0	3	5	0

Actual OD matrix

	1	2	3	4	⋯	$\lvert S\rvert$
1	0	1	3	4	⋯	2
2	2	0	1	0	⋯	3
3	4	6	0	2	⋯	5
4	1	1	4	0	⋯	1
⋮	⋮	⋮	⋮	⋮	0	⋮
$\lvert S\rvert$	1	2	0	3	5	0

Future OD matrix

(b) Travel demand

Fig. 5 Illustration of prediction error correction mechanism

Fig. 5(a) illustrates the execution process of anticipatory dispatching decisions for vehicles in period p. Specifically, bus 1 and bus 2 have executed decisions 7→5→3→2 and 6→8→4→1, respectively. The partial decisions (3→2, 4→1, and 2→9) for period p are made to serve future demand $\tilde{q}_{ij}^{p}$, and the true information of $\tilde{q}_{ij}^{p}$ is known only when the actual demand q_{ij}^{p+1} is received. Although the unexecuted decision 2→9 can be re-optimized in the next period based on the true information, the decisions 3→2 and 4→1 with prediction errors, which have already been executed, cannot be altered. To promptly eliminate the negative impact on the expected execution result, during the computation time t_B of the next period, we will check these decisions and re-dispatch the vehicles to correct the executed decisions.

The prediction error is reflected by the OD offsets, the number deviation, and whether the request is cancelled, all of which contribute to the inconsistency between the predicted and the actual number of passengers on the arc (i, j). As shown in Fig. 5(b), we track the executed decisions that have deviations in demand, such as 3→2 and 4→1. As shown in Fig. 5(c), the prediction errors can be classified into two types, and their corresponding correction mechanisms are as follows:

- Overestimation of future demand, i.e., $\tilde{q}_{ij}^{p} \geq q_{ij}^{p+1}$.

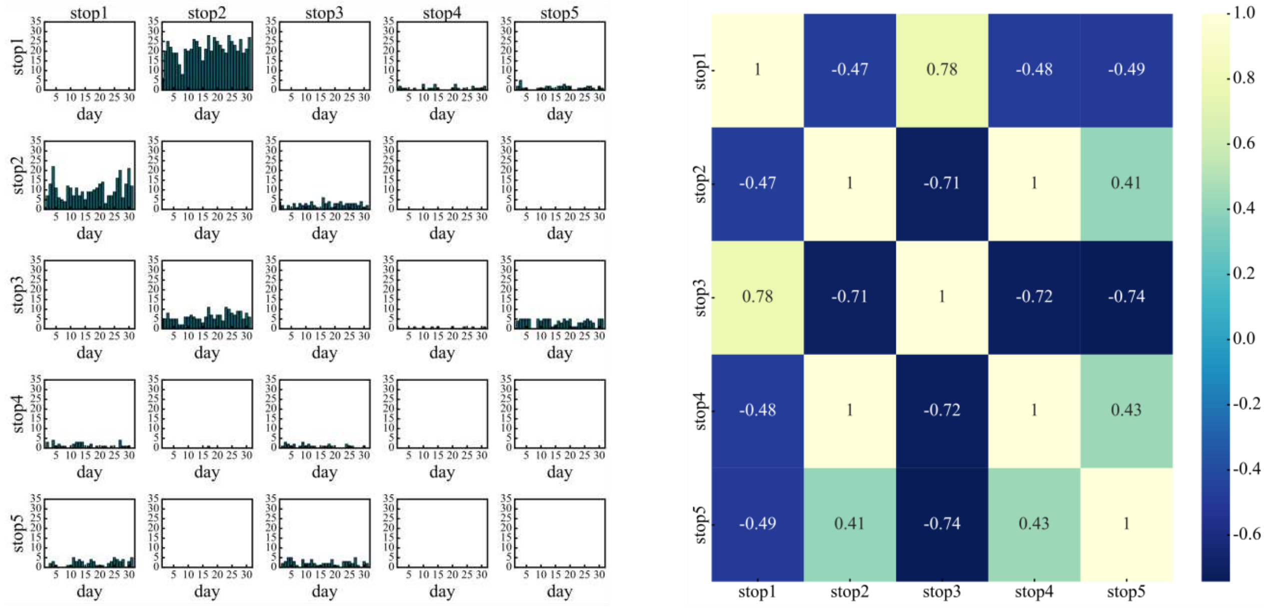
The executed decision ensures the sufficient remaining capacity to accommodate the actual demand, so the original value c_{kv}^{p} can be simply updated to the correct value $c_{kv}^{p+1} = c_{kv}^{p} - q_{ij}^{p+1}$.

- Underestimation of future demand, i.e., $\tilde{q}_{ij}^{p} < q_{ij}^{p+1}$.

If the remaining vehicle capacity c_{kv}^{p} is less than the actual demand, the executed decision is invalid such that a detour results. To avoid wasting mileage and working time, we pick up only c_{kv}^{p} passengers for those split requests, while the other $q_{ij}^{p+1} - c_{kv}^{p}$ passengers that exceed the vehicle capacity are treated as newly submitted requests and are dispatched by other vehicles in the next period. In contrast, the non-split requests have to be rejected subject to the remaining capacity. If the remaining vehicle capacity is not less than the actual demand ($\tilde{q}_{ij}^{p} < q_{ij}^{p+1}, c_{kv}^{p} \geq q_{ij}^{p+1}$), it is equivalent to the overestimation type. For cancelled requests, the distance to the arc that reaches their pick-up stop is considered a detour distance, regardless of the prediction error type.

4.3. Demand prediction with uncertainty

The prediction module is essential to predictive DAR optimization. Fig. 6 shows the DAR travel demand between five stops in a real-world bus network of a ‘Shared Bus’ project in Guangzhou for 31 days (see Section 6.3). Three findings are observed. First, the demand exhibits significant temporal relationships, and the majority of stops exhibit sparse demand. Second, the distribution of demands is highly irregular. For instance, the temporal regularity of OD demands from stop 3 to stop 5 is relatively strong, while that between stop 2 and stop 1 fluctuates sharply. Third, as shown in Fig. 6(b), the absolute values of the correlation coefficients range from 0.41 to 1, indicating a significant spatial correlation in demands. Therefore, a challenge of operation with demand prediction is to simultaneously capture the spatiotemporal characteristics of demand, including uncertainty, sparsity, and spatial correlation. Given the effectiveness of long short-term memory (LSTM) networks in extracting temporal relationships (Zhang et al., 2020; Ma et al., 2021; Liu et al., 2019), we combine LSTM with quantile regression methods to capture the temporal features of short-term sparse passenger demands. Additionally, Copula functions are employed to capture the spatial features between stops.



(a) Time-dependent OD matrix

(b) Heat map of Pearson correlation coefficients

Fig. 6 Temporal and spatial characteristics of demand distribution between five stops

To achieve rolling forecasting and reveal the possibilities for future stochastic demands, we design a multi-scenario prediction model considering spatial correlation that widely captures spatiotemporal characteristics. The principle is similar to Bent and van Hentenryck’s (2004) approach, where future demand scenarios are generated through sampling from its probability distribution. Specifically, we update the historical requests that serve as the training set periodically, and estimate the marginal distribution of each OD in the service area by accumulating the quartiles of the distribution function. After that, Copula functions are introduced to capture the spatial correlation of marginal distributions of different OD pairs to construct joint probability distributions. By multiple sampling, future demand scenarios and corresponding occurrence probabilities are obtained. The main steps are provided as follows:

(1) The historical actual demand $\left\{q_{ij}^{1},\dots,q_{ij}^{p-1}\right\}$ for each OD is obtained. The future demand $\tilde{q}_{ij}^{p(\tau_l)}$ with a quantile level of τ_l is predicted using LSTM and quantile regression.

(2) The corresponding empirical distribution $F\left(q_{ij}^{p}\right)$ is constructed based on $\tilde{q}_{ij}^{p(\tau_l)}$, and $|N|$ future demand scenarios are obtained by randomly sampling from the empirical distribution.

(3) Gaussian Copula function C_G is adopted to capture the spatial correlation between the empirical distributions of each OD pair.

(4) The $|N|$ future demand scenarios are imported into the C_G to obtain its occurrence probability $Pr\left(Q_n^p\right)$.

The advantages of our prediction model are significant, primarily attributed to four aspects: (1) the use of interval estimates of quartiles, instead of traditional point estimates, can retain the uncertainty in travel demand; (2) the sampling of a sufficient number of demand scenarios enables exploration of various possibilities of future demand; (3) the utilization of Copula function accommodates spatial correlation, of which the generated joint probabilities provides the foundation for identifying the optimal policy from the candidate policy set; (4) the periodic update training set can prevent the accumulation of prediction errors. Furthermore, we implement a prediction error correction mechanism that can eliminate the impact of errors. In cases where future demand does not match the actual demand, we handle executed decisions with subtlety rather than completely removing them. This approach allows us to preserve the optimized policy in distinguished future demand scenarios.

4.3.1. Deep quantile regression for time-dependent OD demand

To build the nonlinear relationship between the input and output variables with different quartile levels, we estimate the future demand for each OD pair by combining the LSTM with quantile regression methods. By incorporating the quantile regression into the loss function of the LSTM model, the stochasticity of future demands can be produced without prior assumptions of demand data.

The training dataset consists of the received request information from all previous periods, and $\tau_l = \{5\%, 25\%, 50\%, 75\%, 95\%\}$ is selected to encompass a range of quantile levels. Since the future demand of different quantile levels needs to be predicted, the loss function of the LSTM is modified to align with the optimization equation of quantile regression, that is,

$$loss = \sum_{\wp=1}^{p} \sum_{l=1}^{5} \max\left\{\tau_l \cdot (q_{ij}^{\wp} - \tilde{q}_{ij}^{\wp(\tau_l)}), (1-\tau_l) \cdot \left(\tilde{q}_{ij}^{\wp(\tau_l)} - q_{ij}^{\wp}\right)\right\}, \forall i,j \in S, \forall l \in L \tag{30}$$

In doing so, the quantile sample $\tilde{q}_{ij}^{p(\tau_l)}$ of future demand of each OD pair can be obtained. The actual demand distribution follows the quantile sample performance, i.e., $q_{ij}^{p} {\sim} \varepsilon\left(\tilde{q}_{ij}^{p(\tau_l)}\right)$, which can be expressed as the following empirical distribution:

$$F\left(q_{ij}^{p}\right) = \begin{cases} 0 & , \forall i,j \in S, q_{ij}^{p} < \tilde{q}_{ij}^{p(0.05)} \\ \dfrac{l}{5} & , \forall i,j \in S, q_{ij}^{p} \in (\tilde{q}_{ij}^{p(\tau_l)}, \tilde{q}_{ij}^{p(\tau_{l+1})}], \text{if } l = 1, \dots, 4 \\ 1 & , \forall i,j \in S, q_{ij}^{p} > \tilde{q}_{ij}^{p(0.95)} \end{cases} \tag{31}$$

As a result, the predicted value with uncertainty for each OD pair can be obtained by sampling from the empirical distribution $F\left(q_{ij}^{p}\right)$. By sampling all OD pairs, a stochastic future demand scenario Q_n^p for period p can be obtained, with a dimension of $|S|^2$.

$$Q_n^p = \begin{bmatrix} F\left(q_{11}^{p}\right)^{-1} & \dots & F\left(q_{1|S|}^{p}\right)^{-1} \\ \dots & & \dots \\ F\left(q_{|S|1}^{p}\right)^{-1} & \dots & F\left(q_{|S||S|}^{p}\right)^{-1} \end{bmatrix}, \forall n \in N, \forall p \in P \tag{32}$$

4.3.2. Copula function joint distribution for capturing spatial demand correlation

As discussed previously, the marginal distributions of different OD pairs exhibit spatial correlation. For this reason, the Copula function is adopted to capture this correlation by combining the marginal distributions into a joint demand distribution.

The marginal distribution $F\left(q_{ij}^{p}\right)$ of each OD pair is mapped into an interval [0,1] using the standard cumulative distribution function (CDF) Φ. According to Sklar's Copula theory (Sklar, 1973), the multivariate cumulative distribution function can be defined as a function of these CDFs and a Copula function. In this study, we employ the Gaussian Copula function C_G, represented by Eq. (33).

$$C_G\left(q_{11}^p, \dots, q_{|s||s|}^p\right) = \Phi\left[\Phi^{-1}\left(\Phi\left(q_{11}^p\right)\right), \dots, \Phi^{-1}\left(\Phi\left(q_{|s||s|}^p\right)\right); \sigma\right] \quad (33)$$

where Φ^{-1} represents the inverse function of Φ, and σ denotes the covariance matrix. By substituting Q_n^p into C_G, we can obtain the occurrence probability $Pr\left(Q_n^p\right)$.

4.4. Pruning strategies

In this section, we propose several families of pruning strategies to strengthen the dynamic programming formulations, which improve the computational efficiency and solution quality of ADP. The principle is to reduce unnecessary decisions and state space, and avoid incorrect vehicle decisions that lead to large update deviations and variance of the state value function.

4.4.1. Upper bound of the driver's working time

Since vehicle v must deliver all passengers of assigned requests to their drop-off stops, it makes sense to determine whether the driver's working time is sufficient to complete the subsequent route before going to the new pick-up stop o_r of request r. The following theorems are given along with their proofs.

Theorem 1: There is always an upper bound for the driver's working time $\sup u_{kv}^p$ when dispatch vehicle v to pick-up stop o_r of request r at decision epoch k of period p.

Proof: Suppose that the driver has sufficient working time, there are two scenarios when the vehicle arrives at the pick-up stop.

• Scenario 1. The vehicle is empty

If the vehicle has not served the pick-up stop in previous decision epochs, its subsequent route is either: (a) pick-up stop→ corresponding drop-off stop→ pick-up stop of request r, or (b) pick-up stop of request r → corresponding drop-off stop→ depot $\mathscr{m}_v$.

For route (a), when a vehicle is dispatched to the pick-up stop of request r, the vehicle will hold and serve at this pick-up stop, and a corresponding drop-off stop of this request will be generated subsequently in its route. Since this pruning strategy needs to be identified each time the vehicle arrives at the pick-up stop, it is only necessary to consider whether the remaining working time after holding and serving can complete the trip to the corresponding drop-off stop. Furthermore, the vehicle is forced to return to the depot from the drop-off stop, i.e., route (b). In summary, for Scenario 1, there exists an upper bound of the working time when the driver serves the pick-up stop, such that $\sup u_{kv}^p = t_{\max} - \frac{d_{\left(h_{kv}^p, o_r\right)} + d_{(o_r, d_r)} + d_{(d_r, \mathscr{m}_v)}}{\mathscr{s}} - \lambda_{k o_r v r}^p - 2ec_r, \forall v \in V, \forall k \in K^p, \forall p \in P$.

• Scenario 2. The vehicle is loaded with passengers

If the driver has served no less than one pick-up stop in previous decision epochs, i.e., $\left|O_{kv}^p\right| \geq 1$, the driver needs to serve all drop-off stops in the set D_{kv}^p. Similarly, its subsequent route is either: (c) pick-up stop→ D_{kv}^p → pick-up stop of request r, or (d) pick-up stop of request r → D_{kv}^p → depot $\mathscr{m}_v$.

Similar to the analysis of Scenario 1, route (c) is equivalent to route (d). For route (d), its starting stop and end stop are fixed as o_r and $\mathscr{m}_v$, respectively. Although different permutations of drop-off stops in the set D_{kv}^p constitutes various routes, there must be a minimum travel time.

To ease representation, route (d) is defined as $o_r \to \inf D_{kv}^p \to \mathscr{m}_v$, and its upper bound of the driver's working time satisfies $\sup u_{kv}^p = t_{\max} - \frac{d_{\left(h_{kv}^p, i_r\right)} + d\left(o_r \to \inf D_{kv}^p \to \mathscr{m}_v\right)}{\mathscr{s}} - \lambda_{k o_r v r}^p - e\left(2c_r + c_{\max} - c_{kv}^p\right)$, where $d\left(o_r \to \inf D_{kv}^p \to \mathscr{m}_v\right)$ denotes the shortest travel distance for a vehicle to serve all drop-off stops in the set D_{kv}^p from the starting stop o_r to the end stop $\mathscr{m}_v$, which can be calculated by the Dijkstra algorithm, and where $e\left(2c_r + c_{\max} - c_{kv}^p\right)$ denotes the serving time of request r (both boarding and alighting) and other passengers in-vehicle (only alighting). For $\sup u_{kv}^{p} = t_{\max} - \left(d_{(h_{kv}^{p},o_r)} + d_{(o_r,d_r)} + d_{(d_r,m_v)}\right)/s - \lambda_{ko_rvr}^{p} - 2ec_r$ in Scenario 1, D_{kv}^{p} contains only one drop-off stop d_r. Combining Scenario 1 with Scenario 2, the upper bound value of the driver's working time can be expressed as follows:

$$\sup u_{kv}^{p} = t_{\max} - \frac{d_{(h_{kv}^{p},i_r)} + d\left(o_r \to \inf D_{kv}^{p} \to m_v\right)}{s} - \lambda_{ko_rvr}^{p} - e\left(2c_r + c_{\max} - c_{kv}^{p}\right), \forall v \in V, \forall k \in K^{p}, \forall p \in P \tag{34}$$

This completes the proof of the existence of the upper bound of the driver's working time. Using this pruning strategy, we can eliminate the decisions where the driver's working time is greater than its upper bound. Since for these decisions the vehicle cannot serve the subsequent drop-off stops and return to the depot successfully, the eliminated routes are incorrect and cannot be optimal policy.

Theorem 2: Compared with the standard dynamic programming, the state space of decision epoch k of period p will be reduced by $\left(\sum_{r\in R} q_r^{p} - |O_{kv}^{p}|\right)\left(|O_{kv}^{p}| + 1\right)!$ if the decision is pruned.

Proof: In period p, the number of received requests is $\sum_{r\in R} q_r^{p}$. Suppose that vehicle v has served $|O_{kv}^{p}|$ pick-up stops at the decision epoch k and its remaining capacity can guarantee that the vehicle serves passengers at $\sum_{r\in R} q_r^{p} - |O_{kv}^{p}|$ pick-up stops. According to the dynamic programming algorithm, the vehicle will traverse $\left(\sum_{r\in R} q_r^{p} - |O_{kv}^{p}|\right)$ pick-up stops. Once the vehicle arrives at each pick-up stop, the set of corresponding drop-off stops D_{kv}^{p} is formed, of which length equals to $|O_{kv}^{p}| + 1$ and contains totally $\left(|O_{kv}^{p}| + 1\right)!$ permutations. Thus, the state space of decision epoch k equals $\left(\sum_{r\in R} q_r^{p} - |O_{kv}^{p}|\right)\left(|O_{kv}^{p}| + 1\right)!$, which will not be recorded if the decision is pruned.

4.4.2. Holding time

Since vehicle v can hold in depot m_v, the departure time can be optimized to reduce the time penalty cost for the first-served pick-up stop. Since the heading pick-up stop o_r after holding has been determined and its subsequent decisions cannot be observed, we only optimize the arrival time at this stop. Eq. (35) calculates the optimal holding time λ_{ksvr}^{p}.

$$\lambda_{ksvr}^{p*} = \underset{\lambda_{ksvr}^{p}}{\operatorname{argmin}}\left\{\beta_e \cdot \max\left\{ET_r - \left(t_{kv}^{p} + \frac{d_{(m_v,o_r)}}{s} + \lambda_{ksvr}^{p}\right), 0\right\} + \beta_l \cdot \max\left\{\left(t_{kv}^{p} + \frac{d_{(m_v,o_r)}}{s} + \lambda_{ksvr}^{p}\right) - LT_r, 0\right\}\right\} \tag{35}$$

Eq. (36) calculates the optimal penalty cost CP_k^{p*}.

$$CP_k^{p*} = \beta_e \cdot \max\left\{ET_r - \left(t_{kv}^{p} + \frac{d_{(m_v,o_r)}}{s} + \lambda_{ksvr}^{p*}\right), 0\right\} + \beta_l \cdot \max\left\{\left(t_{kv}^{p} + \frac{d_{(m_v,o_r)}}{s} + \lambda_{ksvr}^{p*}\right) - LT_r, 0\right\} \tag{36}$$

4.4.3. Return to the depot

When the vehicle has served each drop-off stop and no onboard passengers left, the vehicle may return to the depot at the next decision epoch to achieve lower costs, and thus its subsequent route is either: (e) pick-up stop of request r → corresponding drop-off stop, or (f) the vehicle's depot m_v → pick-up stop of request r → corresponding drop-off stop. Since the pruning strategy of the upper bound of the driver's working time in Section 4.4.1 has eliminated the incorrect vehicle routes, the premise of route (e) is that the remaining working time is sufficient. Hence, we prune the decision from the cost perspective.

The cost of route (f) must be higher than route (e) without considering the time window penalty, even if the fixed transportation cost is ignored (similar to Clarke and Wright's saving method). To reduce the penalty cost and total cost, it might be more economical to dispatch another vehicle from the depot to the pick-up stop for serving request r. Based on this principle, the pruning strategy is set as follows: When $\beta_v\left(d_{(h_{kv}^p,o_r)}+d_{(o_r,d_r)}+d_{(d_r,m_v)}\right)+CP_k^p<\beta_v\left(d_{(h_{kv}^p,m_v)}+d_{(m_v,o_r)}+d_{(o_r,d_r)}+d_{(d_r,m_v)}\right)+CP_{k+1}^{p*}$, the route (e) is adopted; when $\beta_v\left(d_{(h_{kv}^p,o_r)}+d_{(o_r,d_r)}+d_{(d_r,m_v)}\right)+CP_k^p\geq\beta_v\left(d_{(h_{kv}^p,m_v)}+d_{(m_v,o_r)}+d_{(o_r,d_r)}+d_{(d_r,m_v)}\right)+CP_{k+1}^{p*}$, the route (f) is adopted, where the formulation of CP_k^{p*} is shown in Section 4.4.2.

5. Solution algorithm

For the DAR optimization, the state space and decision space of the MDP model are so enormous that the optimal policy cannot be searched quickly using standard dynamic programming. Approximate dynamic programming is effective in tackling the curse of dimensionality (Powell, 2007). In this study, we select the prevailing VFA to speed up the solving of the MDP model.

Under the operation with demand prediction, we need to search for the optimal policy in a stochastic environment mixed with actual and future demands. For the learning of optimal policy in a stochastic environment, previous studies on ADP or reinforcement learning methodologies have focused on exploring the stochastic environment by attempting feasible decisions and adaptive approximation updates (Kool et al., 2018; Çimen and Soysal, 2017). Another stream has applied exact algorithms (e.g., branch-and-price) or stochastic programming (e.g., chance-constrained programming) to solve the vehicle dispatching problem in a stochastic environment (Florio et al., 2022; Lee et al., 2022; Wu et al., 2022; Wu et al., 2024). However, the former approach is difficult to maintain the stability and high quality of the solution with dynamic pop-up requests, while the latter approach is inefficient in periodic re-optimization for DAR due to computational complexity. Distinct from prior research, considering both actual and stochastic future demands simultaneously, we search for the optimal policy for each period via the ADP algorithm and scenario approach. Specifically, in each period, we approximate the value function of the state to find the candidate policy set for $|N|$ scenarios, and identify the optimal policy from the candidate policy set. Meanwhile, under the rolling horizon framework, since unexecuted decisions can cause state skipping, we still need to devise a method to maintain a stable iteration of the value function. In doing so, the set of scenarios sufficiently reveals the possibilities of future stochastic demands, which can produce reliable and high-quality solutions in a reasonable time by parallel computing. Moreover, the periodic future demand scenarios generation and the prediction error correction mechanism can prevent the accumulation of prediction errors, thereby ensuring the stability of the policy.

Based on this principle, we propose a multi-scenario exploration and rolling ADP framework to achieve high-quality management of DAR operation considering demand prediction. This framework integrates the value function rolling method and the multi-scenario exploration method, to tackle the challenge of deviation of the value function in iterations between adjacent periods and identify the optimal policy from multiple future demand scenarios. Besides, we provide three stepsize rules and discuss the approximation strategies. The framework and stepsize of ADP are presented in Section 5.1 and Section 5.2, respectively, and the approximation strategy is discussed in Section 5.3. In Section 6.2.4, we will compare the performance of our approach with state-of-the-art.

5.1. Multi-scenario exploration and rolling ADP framework

The ADP algorithm tackles the challenge of the curse of dimensionality by estimating the true value of the state value function. The VFA is a widely used method that converges the estimated values to their true values. Specifically, the VFA iteratively generates sample policies through repeated simulation from the initial state to the end state, and these sample policies are employed to approximate the state value function. The sample policy is continuously refined based on the updated value function approximation until it converges to the optimal value function $\mathscr{v}^*(s_k^p)$, at which point the optimal policy is found. Our study aims to find the optimal solution π^{MS} for operation with demand prediction, and identify the sequence of optimal policies $\{\pi_1^{MS}, \dots, \pi_{|P|}^{MS}\}$. Thus, in each period, we perform $|\mathcal{L}|$ iterations to simulate decision-making.

We now introduce the procedure of VFA policy and the temporal difference update method. Let $\mathcal{A}_k^p$ denote the decision space at decision epoch k of period p with constraints and pruning strategies. In general, for period p, the VFA policy starts at the initial state and iterates ℓ times along the simulated sample decision $a_k^{p\ell}$. The unbiased sample estimate of the value of the state s_k^p at ℓ-th iteration is calculated as follows:

$$\hat{\mathscr{v}}^{\ell}(s_k^p) = \mathcal{R}(s_k^p, a_k^{p\ell}) + \bar{\mathcal{V}}^{\ell-1}(s_{k+1}^p) \tag{37}$$

where the value function $\bar{\mathcal{V}}^{\ell-1}(s_{k+1}^p)$ denotes the estimated value of s_{k+1}^p at $(\ell-1)$-th iteration.

At ℓ-th iteration, the service provider makes the decision $a_k^{p\ell}$ using the value function $\bar{\mathcal{V}}^{\ell-1}(s_{k+1}^p)$ after $(\ell-1)$-th iteration.

$$a_k^{p\ell} = \underset{a_k^p \in \mathcal{A}_k^p}{\operatorname{argmin}} \left(\mathcal{R}(s_k^p, a_k^p) + \bar{\mathcal{V}}^{\ell-1}(s_{k+1}^p)\right) \tag{38}$$

There are a few value function approximation methods in the literature. In this study, the temporal difference update method $TD(\lambda)$ is adopted to update the value function approximation by the estimated value of the state. When $\lambda = 0$ in $TD(\lambda)$, there is a special case as shown in Eq. (39). We refer interested readers to Powell (2007) for more details, and the methods experimentally are compared in Section 6.2.4.

$$\bar{\mathcal{V}}^{\ell}(s_k^p) = \begin{cases} \bar{\mathcal{V}}^{\ell-1}(s_k^p) + \alpha_{\ell-1} \sum\limits_{\Bbbk=k}^{|K^p|} \lambda^{\Bbbk-k} \left(\mathcal{R}(s_k^p, a_k^{p\ell}) + \bar{\mathcal{V}}^{\ell-1}(s_{k+1}^p) - \bar{\mathcal{V}}^{\ell-1}(s_{\Bbbk}^p)\right), TD(\lambda) \\ \bar{\mathcal{V}}^{\ell-1}(s_k^p) + \alpha_{\ell-1} \left(\mathcal{R}(s_k^p, a_k^{p\ell}) + \lambda \bar{\mathcal{V}}^{\ell-1}(s_{k+1}^p) - \bar{\mathcal{V}}^{\ell-1}(s_k^p)\right), TD(0) \end{cases} \tag{39}$$

The VFA can accurately approximate the value function of the state of each decision epoch of period p, but is not suitable under the rolling horizon framework. For example, when a new batch of actual demand is received, the decisions may not be executed completely. In such cases, the unexecuted decisions should not participate in the approximation of the value function. Therefore, a value function rolling method is required.

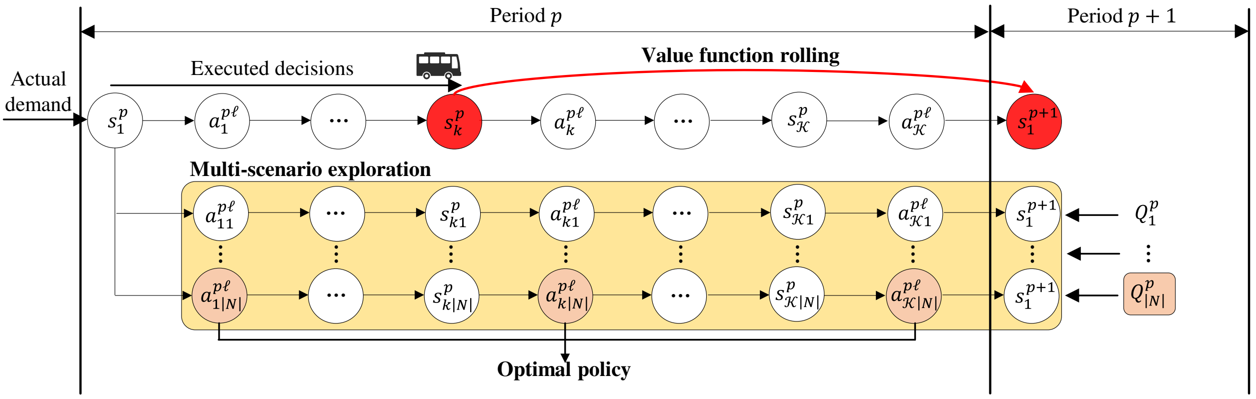


Fig. 7 Illustration of the multi-scenario exploration and rolling ADP framework

Fig. 7 describes the process of the value function rolling method. If the decisions $\{a_1^{p\ell}, \dots, a_{\mathcal{K}}^{p\ell}\}$ for period p is executed up to $a_{k-1}^{p\ell}$ when new actual demand is received, the initial state for period $p+1$ is s_k^p after transition through S^p, i.e., $s_1^{p+1} = s_k^p$. To ensure the stability of the update of the value function between periods, we roll the estimated value of the state s_k^p to state s_1^{p+1} after $|\mathcal{L}|$ iterations in Eq. (40).

$$\bar{\mathcal{V}}^1(s_1^{p+1}) = \bar{\mathcal{V}}^{|\mathcal{L}|}(s_k^p | s_k^p = s_1^{p+1}) \tag{40}$$

Up to now, employing the aforementioned methods, we can conduct $|\mathcal{L}|$ iterations in each period to explore the optimal policy, yielding the optimal solution π for operation without demand prediction. Nevertheless, upon incorporating future demands, in each period, additional iterations are required to explore the future demand scenarios to formulate the candidate policy set, and the optimal policy π_p^{MS} is subsequently identified from these candidate policies. To integrate the scenario approach and ADP algorithms, we propose the multi-scenario exploration method for VFA, which enables the approximation of value function in the scenario-based uncertainty future demand set. Specifically, as shown in Fig. 7, at the start time of period p, we generate $|N|$ simulated channels of sample policies. In each channel, we iterate ℓ times to approximate the value function to find the candidate policy for the corresponding scenario. Given the future demand scenario Q_n^p, we make the sample decision $a_{kn}^{p\ell}$ using the value function $\bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{k+1,n}^p)$ of the state s_{k+1}^p after $(\ell-1)$-th iteration in scenario n:

$$a_{kn}^{p\ell} = \underset{a_k^p \in \mathcal{A}_k^p}{\operatorname{argmin}} \left(\mathcal{R}(s_{kn}^p, a_k^p) + \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{k+1,n}^p) | Q_n^p \right) \tag{41}$$

The sample policy in scenario n is formed when the sample state s_{kn}^p runs to $s_{\mathcal{K}n}^p$. Similarly, in each channel, we use $TD(\lambda)$ and $TD(0)$ methods to approximate the value function.

$$\bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell}(s_{k+1,n}^p) = \begin{cases} \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{kn}^p) + \alpha_{\ell-1} \sum_{\Bbbk=k}^{|K^p|} \lambda^{\Bbbk-k} \left(\mathcal{R}(s_{\Bbbk n}^p, a_{\Bbbk n}^{p\ell}) + \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{\Bbbk+1,n}^p) - \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{kn}^p) \right), TD(\lambda) \\ \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{kn}^p) + \alpha_{\ell-1} \left(\mathcal{R}(s_k^p, a_{kn}^{p\ell}) + \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{\Bbbk+1,n}^p) - \bar{\mathcal{V}}_{\pi_{pn}^{MS}}^{\ell-1}(s_{kn}^p) \right), TD(0) \end{cases} \tag{42}$$

The procedures of the multi-scenario exploration and rolling ADP framework are summarized in Table 4.

Table 4 The pseudo-code of the multi-scenario exploration and rolling ADP framework

Algorithm 3: Multi-scenario exploration and rolling ADP framework

```
Initialize $\bar{\mathcal{V}}^1(s_k^p), \forall k \in K^p, \forall p \in P$
    Set the iteration counter $\ell = 1$, and set the maximum number of iterations $\mathcal{L}$
for $1 \le p \le |P|$ do
  for $1 \le n \le |N|$ do
    for $1 \le \ell \le |\mathcal{L}|$ do
      for $1 \le k \le \mathcal{K}$ do
        Make the sample decision $a_{kn}^{p\ell}$ using Eq. (41) under the future demand scenario $Q_n^p$, and transition the state through $S^k$
      end for
      Calculate the estimate and update the value function approximation using Eq. (42)
    end for
    Solve the optimal policy $\pi_{pn}^{MS}$ in scenario $n$
  end for
    Identify the optimal policy $\pi_p^{MS}$ using Eq. (29) from the candidate policy set $\{\pi_{p1}^{MS}, \dots, \pi_{p|N|}^{MS}\}$, and transition the state variable based on this policy by $S^P$
  if $s_1^{p+1} \ne s_{\mathcal{K}}^p$ (Incomplete execution) then
    Roll the value function of $s_1^{p+1}$ using Eq. (40)
  end if
end for
```

5.2. Stepsizes of ADP

The setting of stepsizes is vital to ADP. Although the algorithm is theoretically convergent, in practice, inappropriate stepsizes can even lead to the failure in the convergence of the approximation of the value function. In this study, we use the three stepsize rules proposed by Powell (2007): fixed stepsize, Harmonic stepsize, and bias-adjusted Kalman filter (BAKF) stepsize, as listed in Table 5.

Table 5 Three stepsizes and their rules

Stepsize	Fixed	Harmonic	BAKF
Rule	$\alpha_{\ell-1} = \begin{cases} 1, \ell = 1 \\ \alpha, \text{otherwise} \end{cases}$	$\alpha_{\ell-1} = \dfrac{\alpha}{\alpha + \ell - 1}$	$\alpha_{\ell-1} = 1 - \dfrac{(\sigma^2)^\ell}{(1+\lambda^{\ell-1})(\sigma^2)^\ell + (\beta^\ell)^2}$

BAKF is a stochastic stepsize rule that can self-adjust based on the difference between the approximate value and the actual value. β^ℓ is a deviation estimate on the $\hat{v}^\ell(s_k^p)$:

$$\beta^\ell = \mathbb{E}[\hat{v}^\ell(s_k^p)] - \hat{v}^{\ell+1}(s_k^p) \tag{43}$$

$(\sigma^2)^\ell$ denotes the variance estimate for $\hat{v}^\ell(s_k^p)$:

$$(\sigma^2)^\ell = \mathbb{E}\left[\left(\bar{v}^\ell(s_k^p) - \hat{v}^\ell(s_k^p)\right)^2\right] \tag{44}$$

5.3. Approximation strategy

Approximation strategies are generally classified into three categories: lookup table, parametric representation, and nonparametric representation (Powell, 2007). Among them, parametric representation is widely used for policy function approximation, so we only discuss lookup tables and nonparametric representation typified by a neural network (Kool et al., 2018). Since the new batch of requests alters the current environment, in each period, the DAR optimization needs an offline simulation to re-find the optimal policy. In this study, we use the lookup table strategy instead of the neural network. There are two primary reasons for this: (a) Short training time. We can find the optimal policy within short computational time by the lookup table, while the neural network requires long training time such that a quick response service is difficult to achieve; (b) Periodic sparse requests. In practice, dynamic requests in each period are often sparse, leading to a small training set. However, since training the neural network relies on a huge number of samples, the neural network may have poorer training results compared to the lookup table with periodic sparse requests.

6. Computational experiments and application

6.1. Performance indicators

To validate the advantages of our model, the following indicators are provided in this paper in addition to the total cost defined in the previous section.

6.1.1. Service level indicators

(1) Response rate (RR): The ratio of served requests over the entire planning horizon.

$$RR = \left(1 - \frac{The\ number\ of\ rejected\ requests}{The\ number\ of\ requests}\right) \times 100\% \tag{45}$$

(2) Lateness rate (LR): The ratio of late-served requests over the entire planning horizon.

$$LR = \left(\frac{The\ number\ of\ late\ served\ requests}{The\ number\ of\ requests - The\ number\ of\ rejected\ requests}\right) \times 100\% \tag{46}$$

(3) Average late time (ALT): The total late time can be calculated for all late-served requests. Dividing the total late time by the number of served requests yields the average late time.

$$ALT = \frac{Total\ late\ time}{The\ number\ of\ requests - The\ number\ of\ rejected\ requests} \tag{47}$$

6.1.2. Operational efficiency indicators

(1) Effective average user cost (EAUC): Average cost of serving one request.

$$EAUC = \frac{Total\ cost}{The\ number\ of\ requests - The\ number\ of\ rejected\ requests} \tag{48}$$

(2) Effective average passenger mileage (EAPM): Average mileage of serving one request.

$$EAPM = \frac{Total\ mileage}{The\ number\ of\ requests - The\ number\ of\ rejected\ requests} \tag{49}$$

(3) Mileage utilization rate (MUR): The running mileage without passengers is called unloaded mileage. The mileage utilization rate reflects the operational efficiency of the policy, which can be calculated as follows:

$$MUR = \frac{Total\ mileage - Unloaded\ mileage}{Total\ mileage} \times 100\% \tag{50}$$

6.2. Numerical test

6.2.1. Benchmark instances

The well-known Sioux Falls network is used for validating the effectiveness of the proposed model. As shown in Fig. 8, the network has 24 stops and 38 links. There are two depots (green node) located at stop 1 and stop 2. The numbers on the links represent the lengths of the road section. Each stop can be used as both pick-up and drop-off stops. The vehicle's travel speed is constant. There are a total of 8 homogeneous vehicles of the same capacity, with 4 vehicles in each depot. To mimic the real scene, we randomly sample 118 time points from the real-world request information ('Shared Bus' project in Guangzhou, see Section 6.3) as the request's submission time during the entire planning horizon (7:30-21:00), and choose two stops from the Sioux Falls network as their pick-up and drop-off stops to simulate the many-to-many characteristics of DAR service. The results are shown in Appendix A. The pick-up time window of each request is set by adding a random time interval to its submission time for not less than 10 minutes. Each OD demand is generated following a normal distribution with a mean value of $\mu = 2.542$ and a variance of $\epsilon^2 = 1.155^2$ according to the historical data (see Section 6.3). As a result, the number of passengers for each request is generated based on the distribution. The other default parameters are listed in Table 6, except where they are in the sensitivity test.

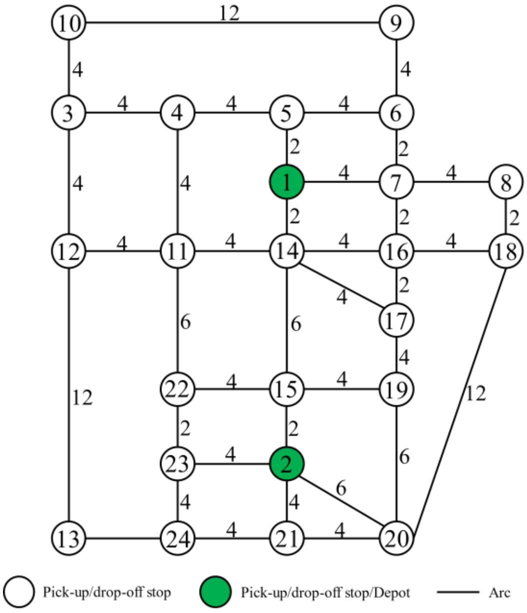


Fig. 8 Sioux Falls network

Table 6 Parameter values in the numerical test

Parameters	Values	Units
T	20	min

$\tilde{T}$	10	min
$\mathscr{s}$	30	$km \cdot h^{-1}$
$\mathcal{L}$	1000	-
λ	0.9	-
ε	2.5	-
$c_{\max}$	15	$pax \cdot vehicle^{-1}$
$t_{\max}$	240	min
e	0.1	$min \cdot pax^{-1}$
β_f	50	$¥ \cdot trip^{-1}$
β_v	1	$¥ \cdot trip^{-1}$
β_e	1	$¥ \cdot trip^{-1}$
β_l	2	$¥ \cdot trip^{-1}$

6.2.2. The effect of pruning strategies

To examine the power of the pruning strategies proposed in Section 4.4 and how they expedite the dynamic programming (DP), we plot how the state space grows with the number of nodes before and after adding the pruning strategies. Although the numbers of nodes for the DAR problem in each period are different, the problem-solving is the same, so we compare the state space under different numbers of nodes.

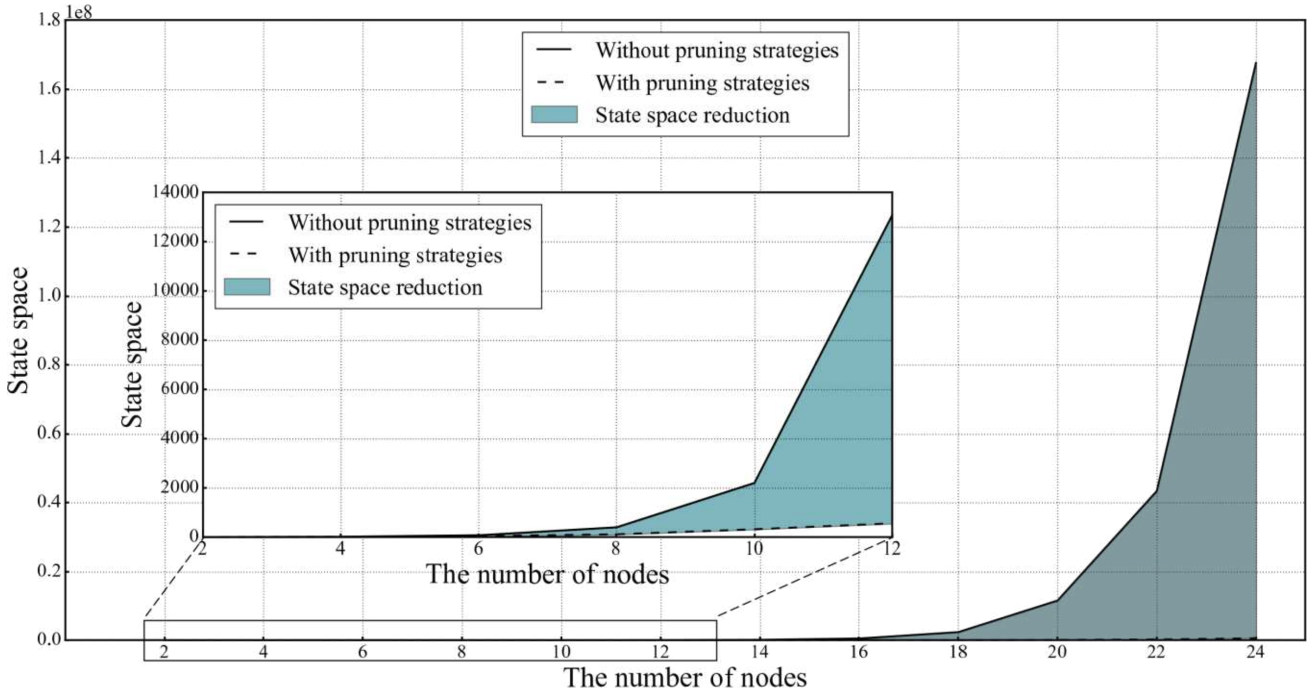


Fig. 9 State space with and without pruning strategies

Fig. 9 shows the state space of the DP corresponding to the optimal solution with and without pruning strategies. The DP requires a large memory. Our preliminary experiment shows that the memory occupancy exceeds 32 GB when the number of nodes is 26. Therefore, we set the maximum number of nodes as 24. It can be seen that the state space increases exponentially from 5 (the number of nodes is 2) to 1.68×10^8 (the number of nodes is 24) without pruning strategies. However, after adding pruning strategies, the state space increases only from 5 to 694437, which drops by 99.59% when the number of nodes is 24. This result verifies that the pruning strategies can dramatically reduce the state space of DP. Since the state space of the problem is not complex when the number of nodes is small, the states satisfying the pruning strategies are sparse. As a result, the effect of pruning strategies is not quite outstanding when the number of nodes is small. For example, when the number of nodes is 4, the state space only decreases by 27.78% from 18 to 13. As the number of nodes increases, the available space for pruning strategies grows quickly. The number of nodes can even reach thousands in actual operation, which implies promising application results of pruning strategies.

6.2.3. Algorithm tuning

Unlike the rule of BAKF stepsize that can be adjusted adaptively according to the value function iteration, the stepsize in the $TD(\lambda)$ is a fixed hyperparameter, which needs to be optimized before the commencement of the learning process. In this section, we divide the stepsize with the interval of 0.1 and set 50 random seeds for each stepsize.

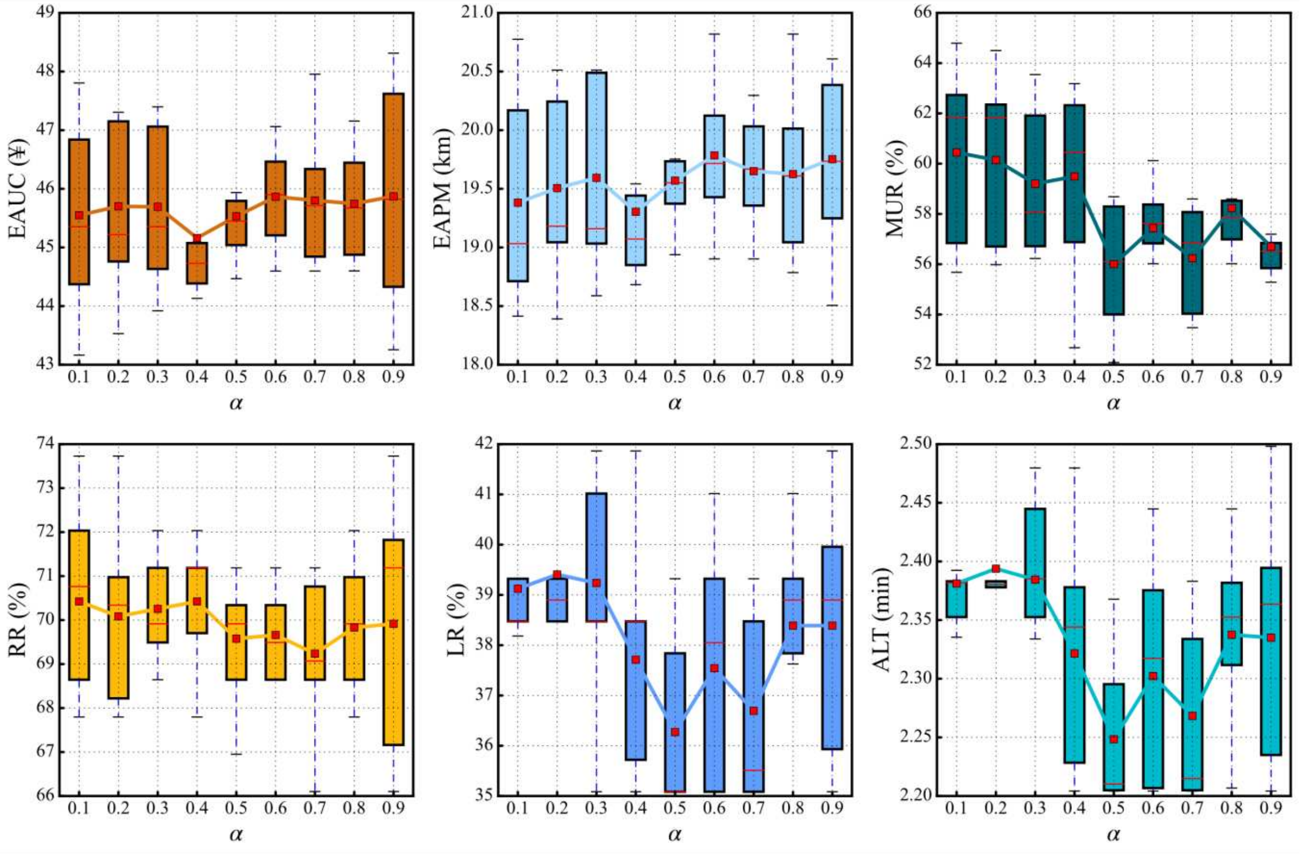


Fig.10 Tuning of the stepsize of $TD(\lambda)$ algorithm

Fig. 10 shows the tuning results, where the horizontal axis and vertical axis represent the stepsize values and six indicators, respectively. We notice that the performance of the six indicators first improves and then deteriorates as α increases, which indicates the existence of optimal values. Specifically, the EAUC reaches a minimum value of 45.16 when $\alpha = 0.4$; the EAPM reaches a minimum value of 19.30 when $\alpha = 0.4$; the MUR reaches a maximum value of 60.45% when $\alpha = 0.1$; the RR reaches a maximum value of 70.42% when $\alpha = 0.4$; the LR reaches a minimum value of 36.27% when $\alpha = 0.5$; and the ALT reaches a minimum value of 2.25 when $\alpha = 0.3$. It can be seen that $\alpha = 0.4$ outperforms in terms of EAUC, EAPM, and RR, but performs worse in terms of MUR, LR, and ALT. Since serving more requests under fixed fleet size tends to induce a higher time window penalty, a large RR can increase the LR and ALT. Since the LR and ALT can decrease under the operation with demand prediction (see Section 6.2.5), we emphasize less on the MUR, LR, and ALT in the selection of the optimal stepsize. As a result, we choose $\alpha = 0.4$ as the optimal stepsize.

6.2.4. Algorithm comparisons

In this section, we compare 6 algorithms, including $TD(\lambda)$ with three stepsize rules, $TD(0)$, ADP proposed by Çimen and Soysal (2017), and Q-network. We set 50 random seeds for each algorithm with and without pruning strategies. For all experiments, the problems are solved on an Inter(R) Core(TM) i9-10900 CPU @ 2.80GHz core processor, 32GB RAM, and the system is Windows 10 system, using Python 3.7. The results are shown in Table 7.

Table 7 shows the comparative results of different algorithms, along with the number of vehicles employed and the average computational time. As we can see, compared to those without pruning strategies, the indicator values of the algorithms with pruning strategies improve considerably. As validated in Section 6.2.2, the pruning strategies reduce significantly the state space of the DAR problem and the resulting solution time of DP. Since the ADP is iterated through the sample decisions and the exploration of ADP is stochastic, the improvements in the average computational time of different algorithms vary from 4.79% to 19.25%. Meanwhile, the pruning strategies can eliminate the interference of unreasonable decisions and guide the ADP to the decision space with high quality. The ALT of each algorithm is smaller than the maximum allowable late time $\tilde{T} = 10$ min. The average computational time of the first five algorithms t_B is significantly smaller than $T = 20$ min, which suggests that the received requests can be quickly responded to and meet the immediacy requirement of DAR service. However, the average computational time of the classical Q-Network algorithm reaches 195.11s. If the vehicle executes the decision after computation, the state can be quite different such that the expected results of policy cannot be achieved. This indicates that it is reasonable to use the lookup table strategy instead of the neural network strategy (see Section 5.3). Among the algorithms with a lookup table strategy, we select $TD(\lambda)$ with fixed stepsize having the best performance for problem-solving.

Table 7 Comparative results of different algorithms

Algorithm **(with pruning strategies)**	**Employed vehicles**	**EAUC** **(¥)**	**EAPM** **(km)**	**MUR** **(%)**	**RR** **(%)**	**LR** **(%)**	**ALT** **(min)**	**Average** t_B **(s)**
$TD(\lambda)$-Harmonic	8	48.57	20.32	54.92	64.41	38.47	2.32	12.26
$TD(\lambda)$-Fixed	8	**45.16**	**19.30**	**59.49**	**70.42**	37.71	2.32	12.51
$TD(\lambda)$-BAKF	8	45.41	19.86	56.55	70.34	**35.08**	**2.21**	13.22
Çimen and Soysal (2017)	8	46.20	19.76	59.38	69.49	40.17	2.41	**9.56**
$TD(0)$	8	47.98	20.96	53.36	68.64	41.02	2.49	12.20
Q-Network	8	44.86	19.64	57.58	71.19	38.47	2.19	195.11
Algorithm **(without pruning strategies)**	**Employed vehicles**	**EAUC** **(¥)**	**EAPM** **(km)**	**MUR** **(%)**	**RR** **(%)**	**LR** **(%)**	**ALT** **(min)**	**Average** t_B **(s)**
$TD(\lambda)$-Harmonic	8	49.69	21.33	54.92	64.41	39.32	2.35	13.62
$TD(\lambda)$-Fixed	8	46.89	19.75	56.70	67.80	40.25	2.42	13.14
$TD(\lambda)$-BAKF	8	47.58	21.01	53.78	67.80	35.59	2.23	14.15
Çimen and Soysal (2017)	8	48.49	21.26	59.38	67.80	41.19	2.45	10.32
$TD(0)$	8	50.22	23.06	53.36	68.64	42.12	2.54	13.30
Q-Network	8	45.30	20.04	57.58	71.19	39.32	2.21	241.62

6.2.5. Comparison between operation with and without demand prediction

In this section, we compare the overall performance of policies under two different types: operation without demand prediction π and operation with demand prediction π^{MS}, both of which adopt $TD(\lambda)$ with fixed stepsize under the optimal stepsize $\alpha = 0.4$. Table 8 shows the optimal schedule for each vehicle, along with the six indicators and the number of vehicles employed. The schedule includes the visited stops and the holding time at the depot and stops. For example, the schedule of the 2nd trip of vehicle 5 under π is ②2(10:08:30)→(20.00 min)→6→19→7→8→15→1→2, where 2(10:08:30) and (20.00 min) indicate that the departure time from depot 2 is 10:08:30 and the holding time is 20 minutes, respectively. To ease the representation, only the departure times from the depot are provided, and the arrival time at each subsequent stop can be calculated straightforwardly.

Table 8 Vehicle schedules of π and π^{MS}

Vehicle ID	π	π^{MS}
1	①**1(07:20:00)**→18→16→5→19→20→1→5→11→10→1 ②**1(15:20:00)**→1→15→2→8→19→11→19→14→7→	①**1(07:24:00)**→24→15→16→6→8→5→9→10→3→12→6→19→(19.90 min)→8→15→13→9→20→1

	(8.00 min)→12→3→15→1→21→19→1 ③**1(20:40:00)**→5→3→1	②**1(14:12:18)**→10→15→1 ③**1(16:40:00)**→6→12→(20.00 min)→(1.60 min)→17→1→5→10→23→(20.00 min)→1 ④**1(20:47:30)**→5→3→1
2	①**1(07:40:00)**→21→1→15→6→(1.40 min)→6→8→14→5→10→12→3→19→(20.00 min)→17→15→1 ②**1(17:04:00)**→24→11→1 ③**1(18:08:54)**→24→13→24→1→16→10→1	①**1(07:26:00)**→13→24→12→2→9→6→18→1→22→7→1→(20.00 min)→14→8→22→13→3→14→8→12→1 ②**1(16:49:00)**→(20.00 min)→11→22→13→4→14→17→6→7→18→7→6→5→10→12→10→9→2→1
3	①**1(07:40:00)**→24→2→16→9→8→18→21→24→22→1 ②**1(17:20:00)**→1→22→1 ③**1(18:20:00)**→22→20→1 ④**1(19:20:00)**→23→1 ⑤**1(20:20:00)**→23→22→12→3→1	①**1(07:20:00)**→8→11→23→24→13→(4.80 min)→1→20→24→11→19→17→15→7→8→1 ②**1(17:04:00)**→(4.00 min)→14→11→16→7→6→4→(7.20 min)→3→15→1→19→14→13→1→
4	①**1(07:40:00)**→19→6→(5.20 min)→17→9→5→4→6→18→1→23→11→22→8→1 ②**1(19:20:00)**→24→23→1 ③**1(19:52:36)**→5→12→10→9→1	①**1(07:20:00)**→18→16→5→16→19→15→20→5→11→10→1 ②**1(18:00:00)**→24→13→12→21→24→23→1 ③**1(20:00:00)**→7→19→1
5	①**2(07:40:00)**→8→24→18→2 ②**2(10:08:30)**→(20.00 min)→6→19→7→8→15→1→2 ③**2(15:20:00)**→7→14→2 ④**2(17:08:00)**→11→16→2 ⑤**2(18:00:00)**→4→14→17→1→2 ⑥**2(19:17:24)**→5→23→2 ⑦**2(20:22:00)**→7→6→19→23→2	①**2(07:22:00)**→(2.00 min)→10→12→(5.20 min)→3→9→(1.60 min)→16→8→7→24→2 ②**2(16:44:00)**→(4.00 min)→24→11→(20.00 min)→4→22→14→20→6→10→6→23→2
6	①**2(07:40:00)**→22→23→12→3→9→16→7→2 ②**2(12:00:00)**→7→8→2 ③**2(12:48:12)**→22→13→2 ④**2(15:40:00)**→6→3→(4.00 min)→11→22→(7.60 min)→22→13→2 ⑤**2(19:00:00)**→7→18→(5.20 min)→7→5→14→13→2	①**2(07:22:00)**→(2.00 min)→10→24→19→14→18→23→11→22→8→(6.00 min)→15→2→8→19→11→19→(20.00 min)→1→22→2 ②**2(18:08:36)**→6→3→(7.20 min)→10→9→5→19→(9.20 min)→22→12→2
7	①**2(07:40:00)**→10→24→2 ②**2(18:00:00)**→6→4→3→(11.90 min)→2 ③**2(20:20:00)**→10→2	①**2(07:20:00)**→21→1→22→23→16→23→21→24→6→(8.00 min)→7→14→6→3→2 ②**2(17:20:00)**→20→15→(1.40 min)→24→16→6→18→2 ③**2(21:00:00)**→8→19→2
8	①**2(07:40:00)**→10→12→14→17→2 ②**2(09:01:42)**→2→14→20→11→(2.00 min)→14→8→9→20→(2.00 min)→14→12→2 ③**2(17:00:00)**→6→12→3→10→2→22→2 ④**2(19:40:00)**→6→10→15→2	①**2(07:40:00)**→19→6→(9.20 min)→6→5→14→17→2→14→23→18→1→15→2 ②**2(18:00:00)**→12→19→17→7→17→2 ③**2(20:12:06)**→10→15→2

	Employed vehicles and trips	**EAUC (¥)**	**EAPM (km)**	**MUR (%)**	**RR (%)**	**LR (%)**	**ALT (min)**
π	8 vehicles and 33 trips	43.86	19.49	56.37	73.73	31.69	2.20
π^{MS}	8 vehicles and 21 trips	34.56	19.00	61.44	90.99	1.30	0.20

Note. The prediction error ratio is set as $A = 5\%$ (see Section 6.2.6.3) for the π^{MS}

The results show that, given the same number of vehicles employed, the total trip length under π^{MS} is approximately 2-3 times longer than that under π, such that the number of trips is considerably reduced under π^{MS}. For example, the numbers of trips under π and π^{MS} are 33 and 21, respectively. In comparison, the number of trips under π^{MS} decreases by 36.36%. Fewer trips under the same fleet size indicate a more cost-effective policy, resulting in higher MUR (5.07% improvement) and RR (17.26% improvement). We also notice that the EAPM under π^{MS} decreases by 2.51%, but the EAUC decreases by 21.20%. The difference between the two indicators is caused by the power of π^{MS} in reducing the time penalty cost, which is evidenced by the LR (30.39% reduction) and ALT (2.00 minutes reduction). In particular, the LR and ALT are even reduced to 0.

6.2.6. Sensitivity analysis

In this section, we conduct sensitivity analysis of key operational parameters of our model to compare the overall performance of policies for π and π^{MS}: the period time interval T, the degree of dynamism δ, the prediction error ratio, and the cancellation ratio.

6.2.6.1. Impact of the period time interval T

Fig. 11 shows the impact of the period time interval T on the system performance. As we can see, both π and π^{MS} are sensitive to the change of T. There is an inflection point for each indicator under π, which corresponds to its optimal value. However, as T increases, the EAUC, EAPM, LR and ALT under π^{MS} decrease by 15.71%, 13.87%, 9.38% and 93.22%, respectively, whereas the MUR and RR increase by 5.46% and 11.00%, respectively. There are two primary reasons for the difference in inflection points between π and π^{MS}. First, a larger T indicates that more requests can be planned simultaneously at the expense of less flexibility in policy adjustment, such that there is an inflection point. More specifically, when T is smaller than the inflection point, more requests can be planned together at each decision epoch, which improves the optimality gradually; when T is greater than the inflection point, despite the larger number of observed requests, the policy adjustment frequency in response to newly received requests is also reduced, which in turn degrades the optimality. Second, under π^{MS}, since future demand is considered, the negative impact of the reduced frequency of policy adjustments can be effectively eliminated. Therefore, there is no inflection point in π^{MS}. Typically, when T equals half of the entire planning horizon, the prediction horizon equals the other half of the entire planning horizon, such that the system can observe all requests, i.e., the multi-period local optimization can be upgraded to the global optimization that can solve the problem to optimality.

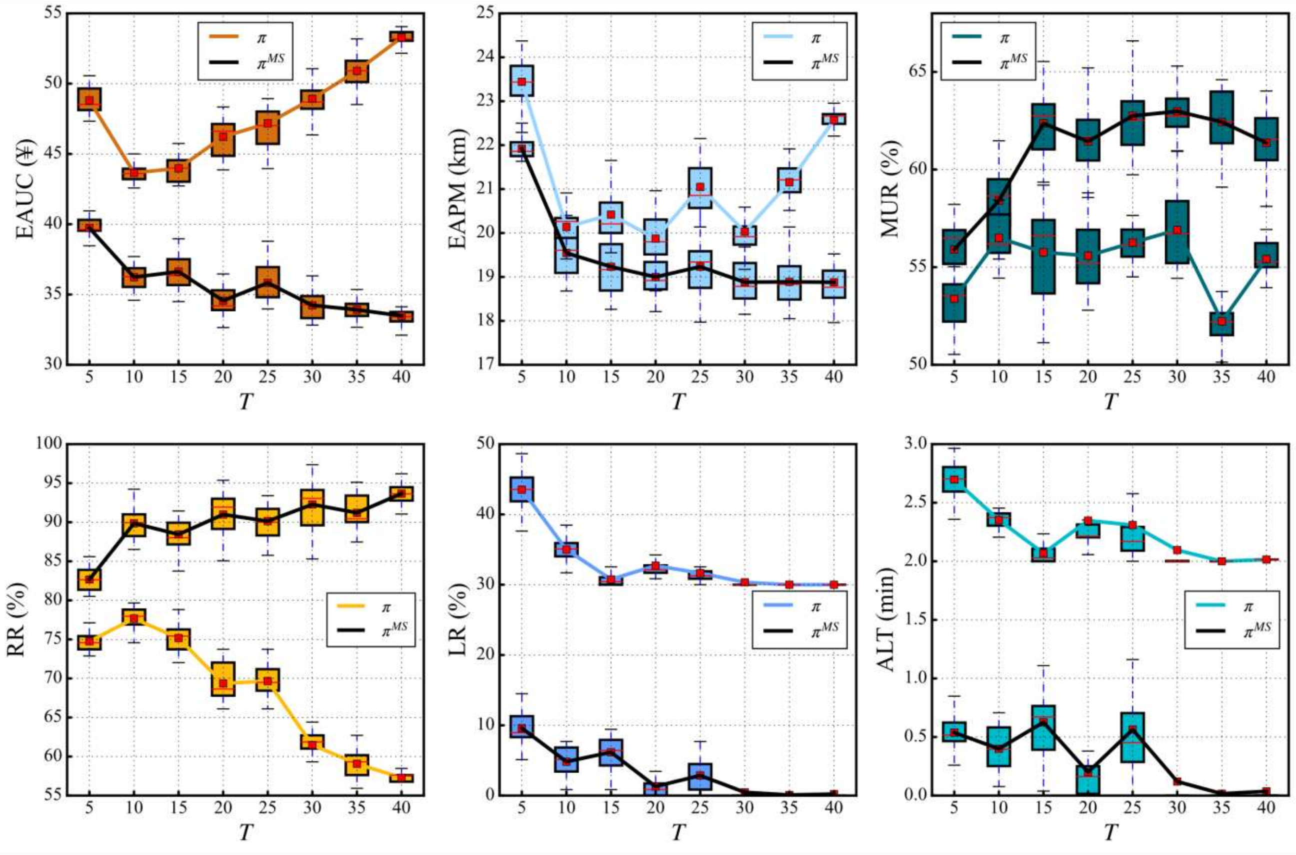


Fig. 11 Sensitivity to T

6.2.6.2. Impact of the degree of dynamism

Since our model can handle dynamic requests, it is interesting to investigate how the degree of dynamism affects the system performance. In line with Lund et al. (1996), the degree of dynamism is defined as the ratio between the number of dynamic requests and the total number of requests, of which dynamic requests are collected after the start time of period 1 during operation (see Fig. 1). We randomly select a specific number of requests from Appendix A as dynamic requests, while the remaining requests are considered static requests pre-collected before operation. Note that dynamic requests participate in the periodic update of historical data in the prediction model.

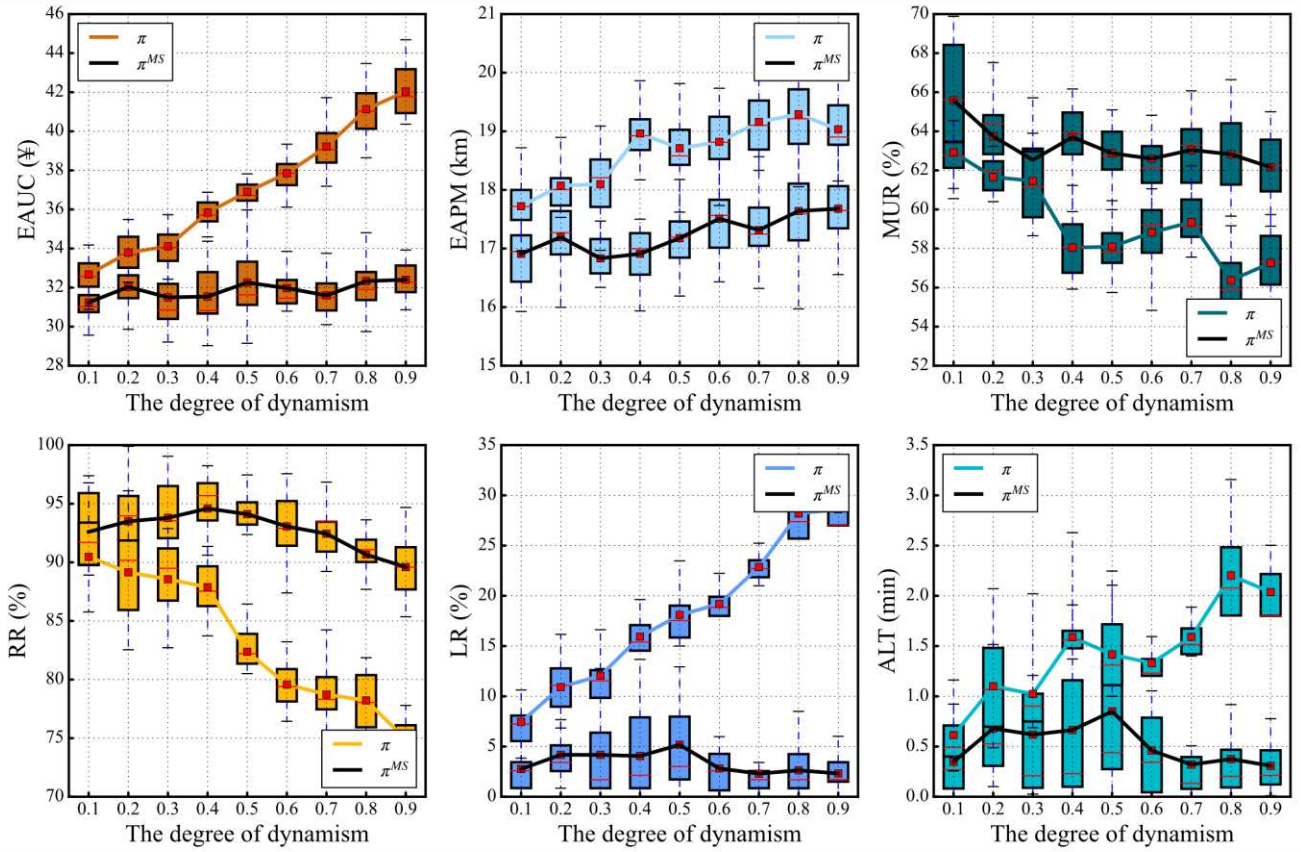


Fig. 12 Sensitivity to the degree of dynamism

Fig. 12 illustrates the impact of δ on the system performance. A greater degree of dynamism means a larger number of dynamic requests that should be processed in an online fashion. As the degree of dynamism grows, the performance under π^{MS} deteriorates slightly. Going from EAUC to RR (except for LR and ALT), the indicator values under π^{MS} deteriorate by 3.58%, 4.35%, 3.42%, and 5.00%, respectively. However, the performance under π deteriorates even more significantly. Going from EAUC to ALT, the indicator values deteriorate by 22.27%, 6.89%, 6.55%, 15.55%, 21.16%, and 69.86%, respectively. This is explainable by the fact under π^{MS}, the impact can be mitigated by revealing future information. Moreover, there is a performance gap between the policy under π and π^{MS}, and the gap increases with a higher degree of dynamism. This is because a larger number of dynamic requests that involve rolling updates can improve the prediction accuracy, leveraging the efficacy of π^{MS}.

6.2.6.3. Resilience testing on the impact of prediction errors

While it is shown that the prediction of future demand improves the system performance to a large extent, it is not uncommon for the service provider to encounter the issue of prediction reliability. The question remains open whether the predictive optimization model can outperform the model without prediction in the presence of prediction errors. We assume that predicted demand biases the actual demand uniformly, and the prediction error is a random variable following a uniform distribution with an interval length depending on ϵ. Eq. (51) represents the number of passengers for selected OD pairs with prediction errors, which equals the total number of requests multiplied by a ratio A. A larger value of A indicates larger prediction errors and larger number of selected OD pairs. In the real world, the number of passengers for each request of DAR services is usually less than 4, that is, the error between the predicted and actual values will not exceed 4. As A increases by 10%, the prediction error increases by 1.155. Therefore, the error range from 0 to 4 can be converted to the grey area in Fig. 13. In other words, the resilience testing covers all possible cases in real-world operations.

$$\tilde{q}_{ij}^p = \begin{cases} q_{ij}^p - \Delta & \text{if } q_{ij}^p - \Delta \geq 0 \\ \Delta - q_{ij}^p & otherwise \end{cases}, \forall p \in P$$
$$\Delta \sim U[-10A\epsilon, 10A\epsilon] \tag{51}$$
$$q_{ij}^p \sim N[\mu, \epsilon^2], \forall p \in P$$

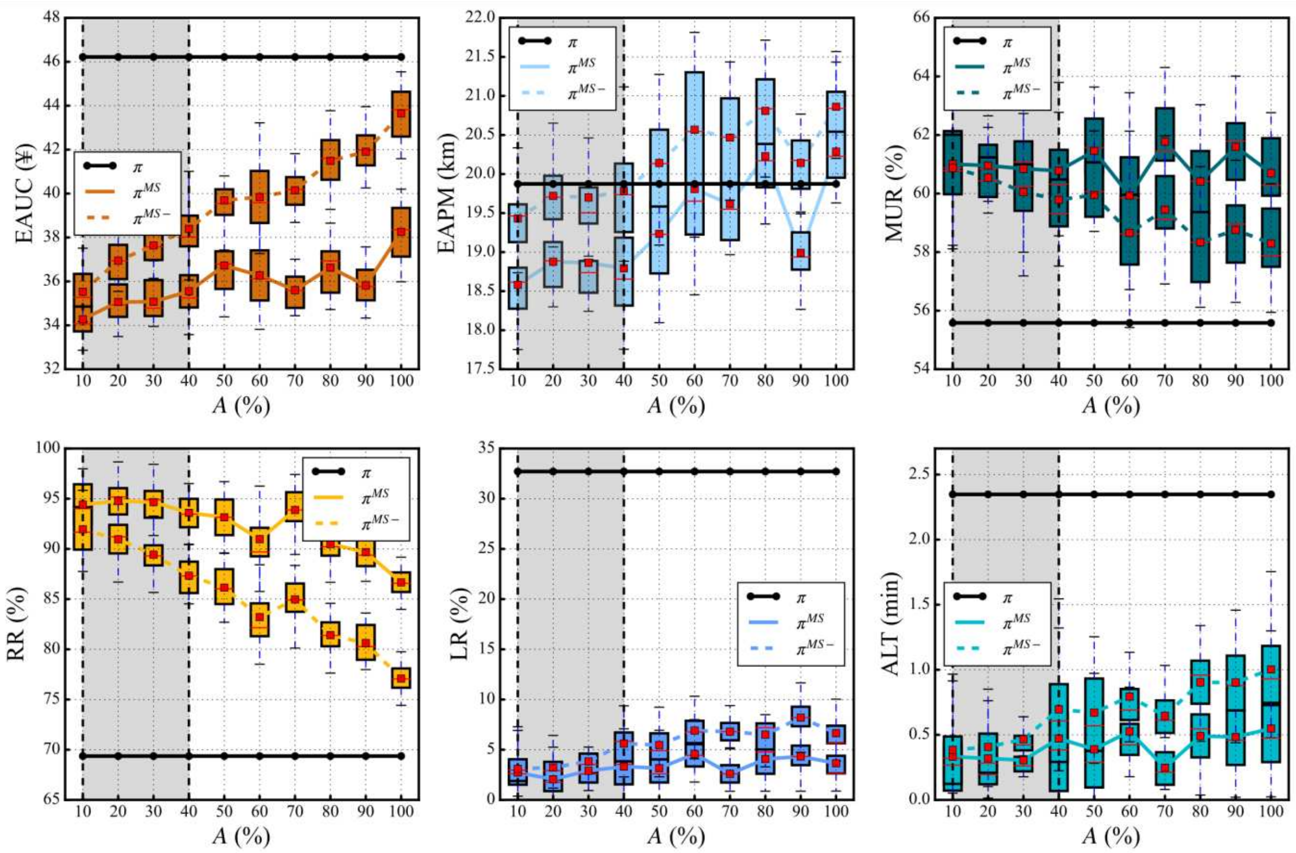


Fig. 13 Sensitivity to prediction errors

Fig. 13 presents the impact of A on the system performance, where the horizontal lines correspond to the indicator values under π with default parameter settings. To examine the benefit of prediction error correction, the results of the prediction-failure-risk-aware policy π^{MS} are benchmarked with its simplified version without awareness of prediction failure risk π^{MS-}, i.e., operation with demand prediction but without prediction error correction. Results show that degradation of the prediction accuracy undermines the overall performance of π^{MS}. Specifically, as A increases from 10% to 100%, the values of EAUC, EAPM, LR, and ALT increase by 11.60%, 9.19%, 0.92%, and 65.11%, respectively, whereas the value of RR decreases by 7.74%. Despite the imperfect prediction accuracy, π^{MS} outperforms by far π. Going from EAUC to ALT, the optimal indicator values ($A = 10\%$) under π^{MS} are improved by 25.83%, 6.51%, 5.42%, 25.04%, 29.99%, and 85.85%, respectively, compared to π. Notably, the prediction-failure-risk-aware policy π^{MS} outperforms the policy π^{MS-} without awareness of prediction failure risk, and the gaps between them are larger with a higher value of A. This suggests that the correction mechanism is effective in eliminating the negative impact of prediction errors, especially when prediction errors are large. In addition, the gap of the EAPM between π and π^{MS} gradually narrows as A increases, while the values of EAPM under both operations are nearly identical when $A = 100\%$. In this vein, we anticipate that when A increases to a certain threshold, the prediction errors may counteract the look-ahead advantage of demand prediction, or even weaken the performance of π^{MS}. Hence, it is beneficial to improve demand prediction accuracy.

6.2.6.4. Impact of the cancellation ratio

As the provision of cancelled requests is a new feature in our study, the impact of cancellation on the policy deserves some discussion. To this end, we define the cancellation ratio as the percentage of all requests that are cancelled temporarily by passengers. Since the request is cancelled by the passenger, it is removed from the calculation process of the indicators.

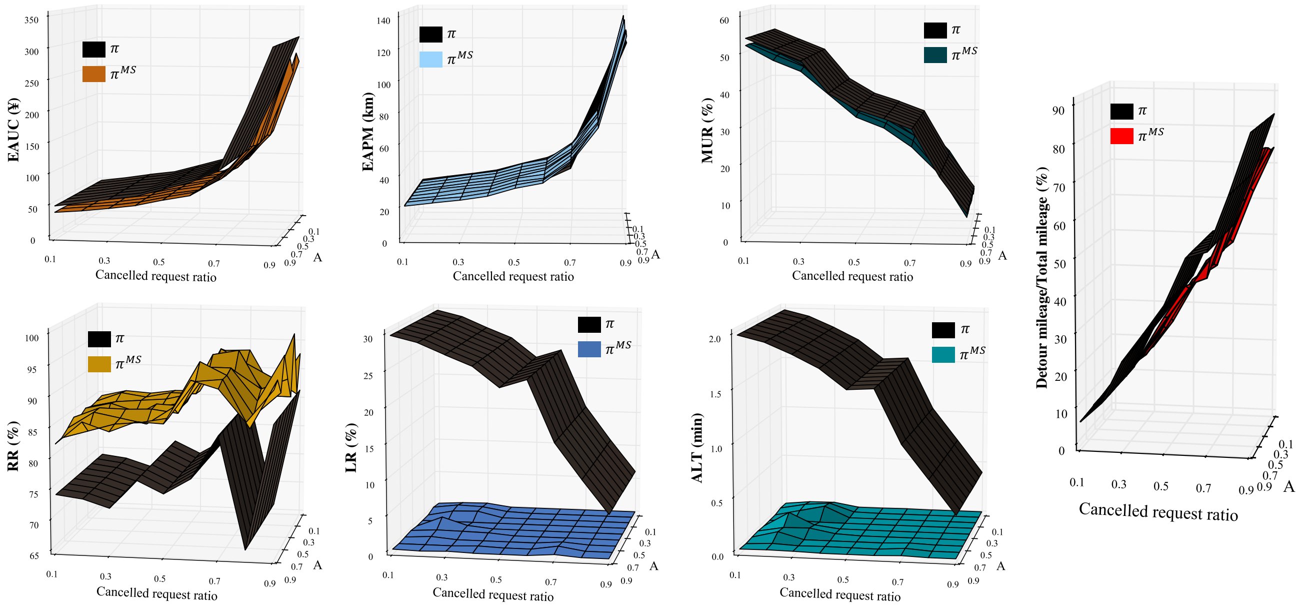


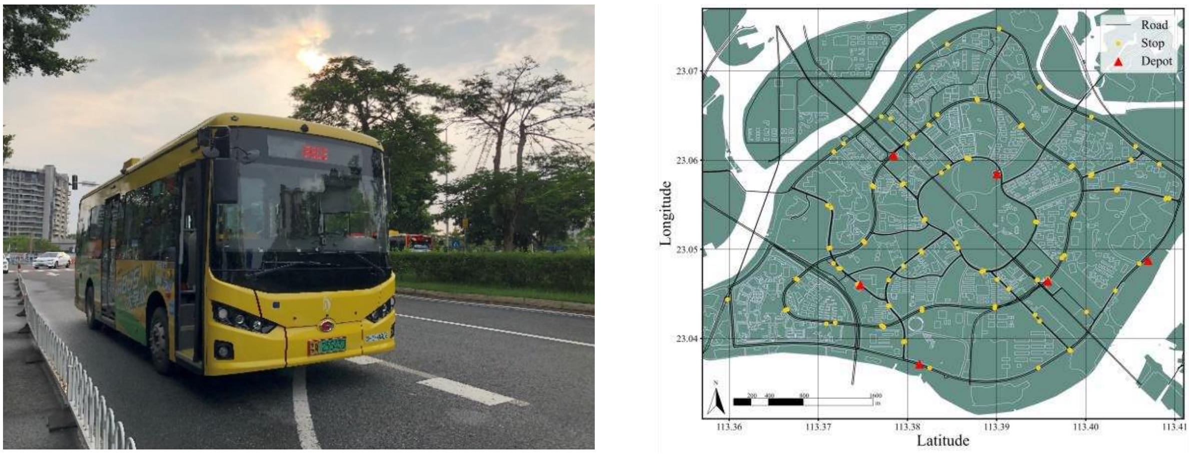
Fig. 14 Sensitivity to the cancelled request ratio under different values of A

Fig. 14 shows the impact of the cancellation ratio on the system performance under different prediction error ratios A. As the cancelled request ratio increases, the number of requests received by the platform decreases, leading to a considerable increase in EAUC and EAPM, since the total cost and total mileage are shared by less served requests. For example, when the cancelled request ratio is 0.9, the EAUC under π^{MS} increases from 36.14 to 263.29 (average value for different values of A), while that under π increases from 46.90 to 303.98. Compared to π, the peak value of EAUC under π^{MS} decreases by 13.39%. This suggests that demand prediction can hedge against the negative effects of request cancellation, which is also supported by the results of EAPM. As the cancelled request ratio increases, the MUR under both operations decreases substantially, while the percentage of detour mileage (the ratio between detour mileage and total mileage) increases significantly. This is because cancelling requests results in a large number of detour mileage and resulting unloaded mileage. As the cancelled request ratio increases from 0.1 to 0.9, the RRs under π and π^{MS} increase by 12.45% and 8.01% (average value for different values of A), which is because cancelling requests releases fleet resources, allowing for serving more requests.

6.3. Case study

6.3.1. Case description

In this section, we report on the applicability of our model and algorithm through a real-world bus network, i.e., Guangzhou Higher Education Mega Center, and compare the performance of both operations with and without demand prediction with that of state-of-the-practice. At present, 68 pick-up stops and 100 drop-off stops have been set up in this area for the 'Shared Bus' project. Fig. 15 shows the DAR vehicle and the distribution of stops and depots in this area. The demand and operation data are provided by a local bus company.



(a) DAR vehicle

(b) Distribution of bus stops and depots

Fig. 15 ‘Shared Bus’ project in Guangzhou Higher Education Mega Center

6.3.2. Comparisons of prediction model performance

Based on the historical data, we predict the number of passengers for each OD pair from June 21, 2021, to July 1, 2021, by using the ‘LSTM+Quantile+Copula’ model proposed in Section 4.3. Fig. 16 shows five prediction curves at 5%, 25%, 50%, 75%, and 95% quantile levels, where prediction curves at the 5% and 95% quantile levels cover almost the historical data. This indicates the practical significance of the empirical distribution $F\left(q_{ij}^{p}\right)$ in Eq. (31). We perform random sampling from the quantile coverage area, which can reduce the huge deviation in a single prediction curve due to the sparse request data.

To evaluate the forecasting performance of the proposed prediction models, a few benchmark models are adopted, including the ‘LSTM+Quantile+Copula’ model, the ‘LSTM+Quantile’ model, the ‘LSTM’ model, and the tensor decomposition model. The principle of the tensor decomposition model is to decompose the sparse tensor by estimating the missing part in the matrix, which can well address the issue of sparse data.

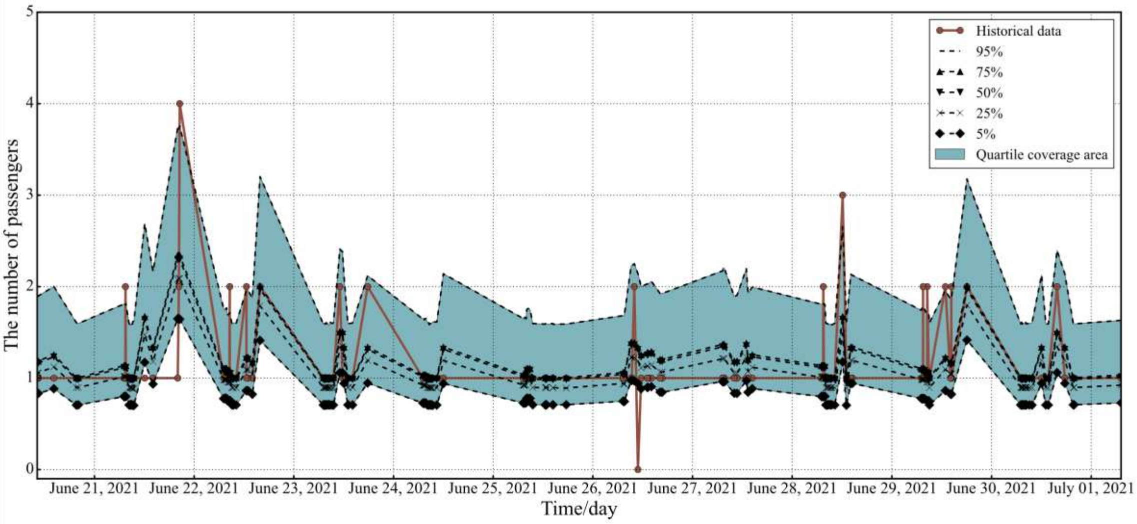


Fig. 16 Results of predicted demand for a typical OD pair by using ‘LSTM+Quantile+Copula’ model

We predict requests in the ‘Shared Bus’ project from August 25, 2021, to August 31, 2021, using the above prediction models. The prediction results are provided in Table 9, where the prediction performance ranks as follows: ‘LSTM+Quantile+Copula’ model > tensor decomposition model > ‘LSTM+Quantile’ model > ‘LSTM’ model. This result demonstrates that the proposed prediction model outperforms the traditional tensor decomposition model. Meanwhile, the Copula function is vital to the prediction model. Although the tensor decomposition model outperforms the 'LSTM+Quantile' model for each indicator, the 'LSTM+Quantile' model with the Copula function, in turn, outperforms the tensor decomposition model. This reinforces the message that the Copula function can fully capture the spatial characteristics of the sparse demand data and improve prediction accuracy. Moreover, the quantile regression mitigates the sparsity effect of the data. As a result, the 'LSTM+Quantile' model outperforms the 'LSTM' model.

Table 9 Prediction performance of four prediction models

Model	MSE	RMSE	MAE	MAPE (%)
LSTM+Quantile+Copula	**0.3871**	**0.6222**	**0.1663**	**5.37**
LSTM+Quantile	1.0036	1.0018	0.4308	14.33
LSTM	1.2342	1.1110	0.5352	17.91
Tensor decomposition	0.4742	0.6886	0.1675	6.55

6.3.3. Results and discussion

Fig. 17 shows the performance of the state-of-the-practice, the operation without demand prediction π, and the operation with demand prediction π^{MS} using four prediction models. Compared to π, the π^{MS} can improve the EAUC, EAPM, and MUR. Note that the LR and ALT under both π and π^{MS} even decrease to 0 and late arrivals are eliminated. Interestingly, the RR under π^{MS} is approximately 1%-3% lower than that under π, which is counter-intuitive at first sight. However, this is reasonable since the request data in actual operation are so sparse that the prediction error ratio can be even higher than 100%, even though Fig. 13 shows that the improvement by prediction is significant when the prediction error ratio does not exceed 100%. The system performance under four π^{MS} follows generally the prediction performance ranks in Table 9, with only a few exceptions. For example, the tensor decomposition model for Aug. 29 performs slightly worse than other models in terms of EAUC. This anomaly is possible because the performance fluctuates slightly as the prediction errors increase.

Next, we compare the π and π^{MS} to state-of-the-practice with respect to six indicators and fleet size. We observe that π and π^{MS} outperforms by far state-of-the-practice for all indicators. In comparison, the improvement by π^{MS} ('LSTM+Quantile+Copula') over 7 days on average are as follows: the EAUC decreases by 38.85%, the EAPM decreases by 19.32%, the MUR increases by 6.38%, the RR increases by 10.69%, the LR decreases by 14.46%, and the ALT decreases by 1.81 minutes. Commendably, the π^{MS} can save fleet size (23% improvement) and resulting investment, which reveals better overall delivery efficiency and service quality.

We also compare DP-based methods with a representative metaheuristic, i.e., genetic algorithm (GA). The results show that both π and π^{MS} outperform GA for all indicators. In comparison, the improvements by π over 7 days on average are as follows: the EAUC decreases by 16.67%, the EAPM decreases by 9.54%, the MUR increases by 2.67%, the RR increases by 7.63%, the LR decreases by 10.17%, and the ALT decreases by 1.24 minutes. These improvements are even more outstanding when demand prediction is considered.

Compared to the 'LSTM+Quantile' model, the improvements by the 'LSTM+Quantile+Copula' model over 7 days are as follows: the EAUC increases by 3.86%, the EAPM decreases by 1.65%, the MUR increases by 0.87%, which indicates that considering the effect of spatial demand correlation can improve the service efficiency of π^{MS}. Compared to the 'LSTM' model, the overall improvements by the 'Quantile+LSTM' model over 7 days are as follows: the EAUC decreases by 0.39%, the EAPM decreases by 0.53%, and the MUR increases by 0.37%. Since the interval estimation of the quantile regression can overcome the prediction uncertainty of the sparse request data, the prediction deviation caused by the default values and the abnormal peaks can be mitigated. Therefore, we can replace the point estimation of the 'LSTM' model with interval estimation to resolve the challenge of sparse request data.

In conclusion, we suggest that the policy π^{MS} with the 'LSTM+Quantile+Copula' prediction model has the best performance, which achieves a win-win situation for both users and operators.

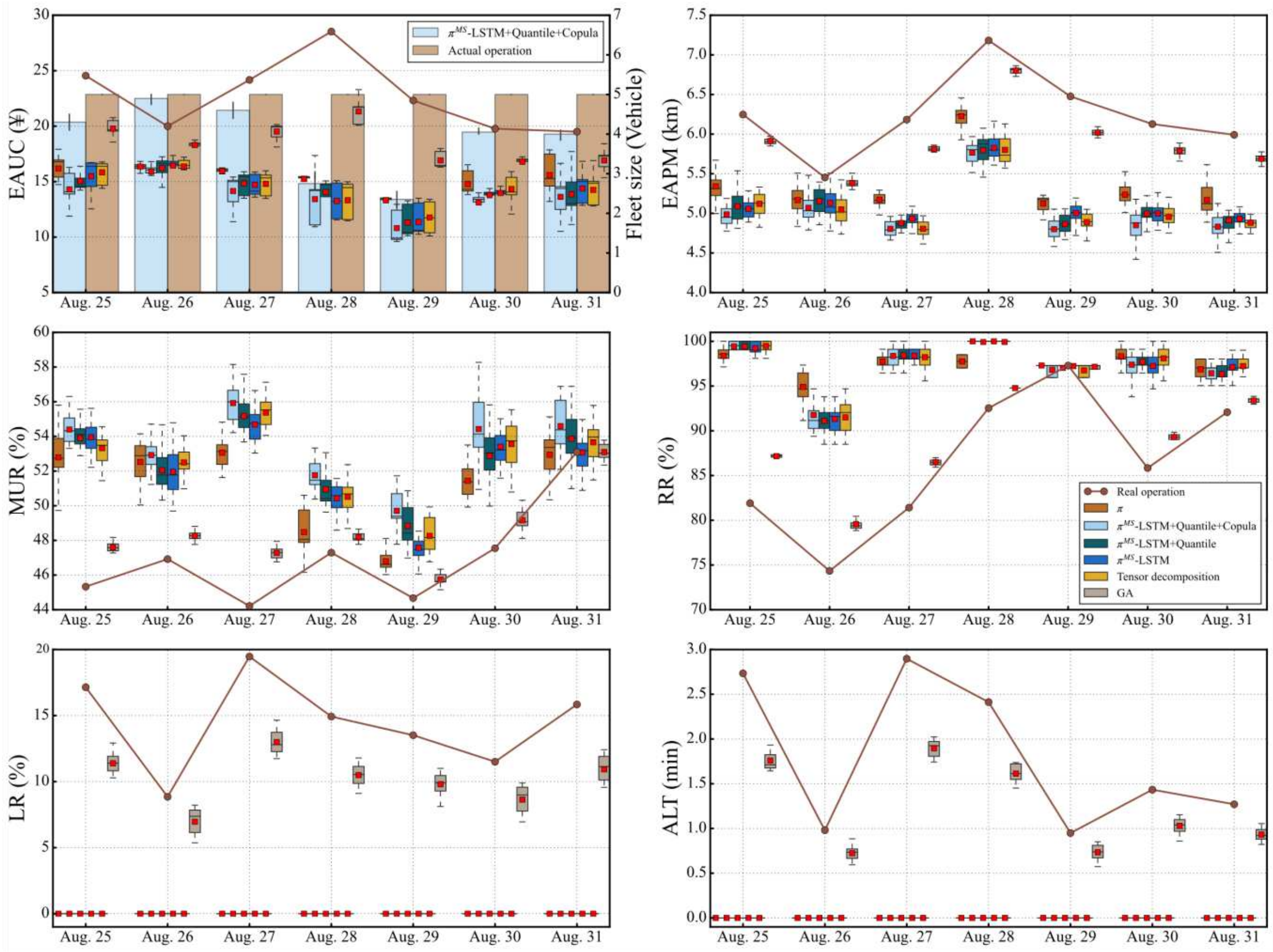


Fig. 17 Results of the case study

7. Conclusion

The DAR service is an essential component of the multimodal transit system. Distinct from prior research, we investigate a prediction-failure-risk-aware online DAR scheduling problem with stochastic and correlated customers. Request selection and cancellation are also explicitly considered. The problem is formulated as a MDP model and solved by the ADP algorithm. We propose a demand prediction model that can capture the unique characteristics of DAR travel demand (i.e., uncertainty, sparsity, and spatial demand correlation) by comprehensively using deep quantile regression, Copula function joint distribution, and scenario sampling approach. Commendably, we propose several families of pruning strategies based on model properties to reduce unnecessary and incorrect decisions. Furthermore, we introduce a prediction error correction mechanism to eliminate prediction errors and rectify policies promptly. In addition, the value function rolling method and multi-scenario exploration method are proposed within the ADP framework, to tackle the challenges of deviation in iteration between adjacent periods and identify the optimal policy from multiple future demand scenarios.

Our model is validated by the Sioux Falls network and a real-world case study in Guangzhou, China. We provide 6 indicators to investigate the performance of the operation with and without demand prediction for various operational settings. A list of new insights and their practical implications are drawn, as follows: (1) the pruning strategies can improve both solution quality and computation efficiency for ADP; (2) the operation with demand prediction can always find high-quality solutions efficiently by serving more requests with fewer fleet sizes and delays; (3) the operation with demand prediction performs more robust against the changes of operational settings than operation without demand prediction. Since more global information can be observed under operation with demand prediction, superior performance can be maintained in complex and dynamic environments. As such, the operation with demand prediction can hedge against the impact of request cancellation; (4) there is a trade-off between planning for more requests simultaneously and less flexibility in policy adjustment, whereas an operation with demand prediction can mitigate the negative impact of reduced frequency of policy adjustments; (5) incorporating demand forecasting and spatial correlation into the DAR operation can increase the system profit and improve its operational performance. The improvement due to prediction is significant even when the prediction is imperfect.

For future work, more exogenous factors and solution approaches can be embedded in our modeling framework. For instance, while this paper primarily focuses on the dynamic DAR problem with a single-vehicle type, it might be interesting to explore the heterogeneous dynamic DAR problem with different vehicle types and capacities. In addition, as our proposed ADP is a generation methodological contribution, it is worth investigating its application to problems in other fields.

Acknowledgements

This work is jointly supported by the National Science Foundation of China (Project No. 72071079, 52272310), Guangdong Basic and Applied Research Foundation (Project No. 2023A1515011696), the Fundamental Research Funds for the Central Universities (Project No. 2023ZYGXZR026), and the open project of the Key Laboratory of Advanced Public Transportation Science, Ministry of Transport, PRC (Project No. 2022-APTS-06).

References

Anderson J., Kalra N., Stanley K., Sorensen P., Samaras C., Oluwatola O., 2014. Autonomous Vehicle Technology: A Guide for Policymakers. RAND Corporation, Santa Monica, California.

Azadeh Sh., Atasoy B., Ben-Akiva M.E., Bierlaire M., Maknoon M.Y., 2022. Choice-driven dial-a-ride problem for demand responsive mobility service. Transportation Research Part B, 161, 128-149.

Bent R.W., van Hentenryck P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research, 52(6), 977-987.

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

Braekers K., Kovacs A.A., 2016. A multi-period dial-a-ride problem with driver consistency. Transportation Research Part B, 94, 355-377.

Berg P. L., Essen J. T., 2019. Scheduling non-urgent patient transportation while maximizing emergency coverage. Transportation Science, 53(2), 492-509.

Bongiovanni C., Kaspi M., Geroliminis N., 2019. The electric autonomous dial-a-ride problem. Transportation Research Part B, 122, 436-456.

Braverman A., Dai J.G., Liu X., Ying L., 2019. Empty-car routing in ridesharing systems. Operations Research, 67(5), 1437-1452.

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

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

Çimen M., Soysal M., 2017. Time-dependent green vehicle routing problem with stochastic vehicle speeds: An approximate dynamic programming algorithm. Transportation Research Part D, 54, 82-98.

Chandakas E., 2020. On demand forecasting of demand-responsive paratransit services with prior reservations. Transportation Research Part C, 120, 102817.

Chen R., Levin M.W., 2019. Dynamic user equilibrium of mobility-on-demand system with linear programming rebalancing strategy. Transportation Research Record, 2673(1), 447-459.

Chen C., Yao F., Mo D., Zhu J., Chen X.M., 2021. Spatial-temporal pricing for ride-sourcing platform with reinforcement learning. Transportation Research Part C, 130, 103272.

Chang X., Wu J., Sun H., Homen de G., Correia A., Chen J., 2021. Relocating operational and damaged bikes in free-floating systems: A data-driven modeling framework for level of service enhancement. Transportation Research Part A, 153, 235-260.

Chang X., Wu J., Sun H., Yan X., 2023. A Smart Predict-then-Optimize method for dynamic green bike relocation in the free-floating system. Transportation Research Part C, 153, 104220.

Desrosiers J., Dumas Y., Soumis F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. American Journal of Mathematical and Management Sciences, 6(3-4), 301-325.

Detti P., Papalini F., de Lara G.Z.M., 2017. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. Omega, 70, 1-14.

Deng Q., Santos B.F., 2022. Lookahead approximate dynamic programming for stochastic aircraft maintenance check scheduling optimization. European Journal of Operational Research, 299(3), 814-833.

Engelen M., Cats O., Post H., Aardal K., 2018. Enhancing flexible transport services with demand-anticipatory insertion heuristics. Transportation Research Part E, 110, 110-121.

Feng S., Ke J., Xiao F., Yang H., 2022. Approximating a ride-sourcing system with block matching. Transportation Research Part C, 145, 103920.

Florio A.M., Gendreau M., Hartl R.F., Minner S., Vidal T., 2022. Recent advances in vehicle routing with stochastic demands: Bayesian learning for correlated demands and elementary branch-price-and-cut. European Journal of Operational Research, 306(3), 1081-1093.

Ghilas V., Demir E., Van Woensel T., 2016. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. Transportation Research Part B, 91, 34-51.

Galarza Montenegro B.D., Sörensen K., Vansteenwegen P., 2021. A large neighborhood search algorithm to optimize a demand-responsive feeder service. Transportation Research Part C, 127, 103102.

Gong C., Shi J., Wang Y., Zhou H., Yang L., Chen D., Pan H., 2021. Train timetabling with dynamic and random passenger demand: A stochastic optimization method. Transportation Research Part C, 123, 102963.

Guo Q., Karimi H., 2017. A novel methodology for prediction of spatial-temporal activities using latent features. Computers, Environment and Urban systems, 62, 74-85

Gschwind T., Drexl M., 2019. Adaptive Large Neighborhood Search with a Constant-Time Feasibility Test for the Dial-a-Ride Problem. Transportation Science, 53(2), 480-491.

Guo Z., Hao M., Yu B., Yao B.Z., 2021. Robust minimum fleet problem for autonomous and human-driven vehicles in on-demand ride services considering mixed operation zones. Transportation Research Part C, 132, 103390.

Guo X., Caros N.S., Zhao J., 2021. Robust matching-integrated vehicle rebalancing in ride-hailing system with uncertain demand. Transportation Research Part B, 150, 161-189.

Guo Z., Yu B., Shan W., Yao B.Z., 2023. Data-driven robust optimization for contextual vehicle rebalancing in on-demand ride services under demand uncertainty. Transportation Research Part C, 154, 104244.

Gao Y., Zhang S., Zhang Z., Zhao Q., 2024. The stochastic share-a-ride problem with electric vehicles and customer priorities. Computers & Operations Research, 164, 106550.

Hvattum L.M., Løkketangen A., Laporte G., 2006. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science, 40(4), 421-438.

Häme L., 2011. An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows. European Journal of Operational Research, 209(1), 11-22.

He F., Wang X., Lin X., Tang X., 2018. Pricing and penalty/compensation strategies of a taxi-hailing platform. Transportation Research Part C, 86, 263-279.

Ho S.C., Szeto W.Y., Kuo Y., Leung J.M.Y., Petering M., Tou T.W.H., 2018. A survey of dial-a-ride problems:

Literature review and recent developments. Transportation Research Part B, 111, 395-421.

Huang D., Gu Y., Wang S., Liu Z., Zhang W., 2020. A two-phase optimization model for the demand-responsive customized bus network design. Transportation Research Part C, 111, 1-21.

Huang X., Ling J., Yang X., Zhang X., Yang K., 2023. Multi-Agent Mix Hierarchical Deep Reinforcement Learning for Large-Scale Fleet Management. IEEE Transactions on Intelligent Transportation Systems, 24(12), 14294-14305.

Jaw J., Odoni A.R., Psaraftis H.N., Wilson N.H., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B, 20 (3), 243-257.

Kim M., Schonfeld P., 2014. Integration of conventional and flexible bus services with timed transfers. Transportation Research Part B, 68, 76-97.

Kool W., Van Hoof H., Welling M., 2018. Attention, learn to solve routing problems!. arXiv preprint arXiv:1803.08475.

Ke J., Yang H., Li X., Wang H., Ye J., 2020. Pricing and equilibrium in on-demand ride-pooling markets. Transportation Research Part B, 139, 411-431.

Ke, J., Zhu, Z., Yang, H. and He, Q., 2021. Equilibrium analyses and operational designs of a coupled market with substitutive and complementary ride-sourcing services to public transits. Transportation Research Part E, 148, 102236.

Koch S., Klein R., 2020. Route-based approximate dynamic programming for dynamic pricing in attended home delivery. European Journal of Operational Research, 287(2), 633-652.

Lund K., Madsen O.B., Rygaard J.M., 1996. Vehicle routing problems with varying degrees of dynamism. IMM, Institute of Mathematical Modelling, Technical University of Denmark.

Luo Y., Schonfeld P., 2007. A rejected-reinsertion heuristic for the static dial-a-ride problem. Transportation Research Part B, 41(7), 736-755.

Lim A., Zhang Z., Qin, H., 2017. Pickup and delivery service with manpower planning in Hong Kong public hospitals. Transportation Science, 51(2), 688-705.

Li D., Antoniou C., Jiang H., Xie Q., Shen W., Han W., 2019. The Value of Prepositioning in Smartphone-Based Vanpool Services under Stochastic Requests and Time-Dependent Travel Times. Transportation Research Record, 2673(2), 26-37.

Luo Z., Liu M., Lim A., 2019. A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. Transportation Science, 53(1), 113-130.

Lee E., Cen X., Lo H.K., 2022. Scheduling zonal-based flexible bus service under dynamic stochastic demand and Time-dependent travel time. Transportation Research Part E, 168, 102931.

Li X., Wang T., Xu W., Li H., Yuan Y., 2022. A novel model and algorithm for designing an eco-oriented demand responsive transit (DAR) system. Transportation Research Part E, 157, 102556.

Liu Y., Liu Z., Jia R., 2019. DeepPF: A deep learning based architecture for metro passenger flow prediction. Transportation Research Part C, 101, 18-34.

Lian Y., Lucas F., Sörensen K., 2024. Prepositioning can improve the performance of a dynamic stochastic on-demand public bus system. European Journal of Operational Research, 312(1), 338-356.

Masson R., Lehuédé F., Péton O., 2013. An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transportation Science, 47(3), 344-355.

Masson R., Lehuédé F., Péton O., 2014. The dial-a-ride problem with transfers. Computers & Operations Research, 41, 12-23.

Masmoudi M.A., Hosny M., Braekers K., Dammak A., 2016. Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. Transportation Research Part E, 96, 60-80.

Ma D., Song X., Li P., 2021. Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns. IEEE Transactions on Intelligent Transportation Systems, 22(5), 2627-2636.

Molenbruch Y., Braekers K., Caris A., Berghe G.V., 2017. Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. Computer & Operations Research, 77, 58-71.

Novoa C., Storer R., 2009. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. European Journal of Operational Research, 196(2), 509-515.

Nguyen H.T., Chow A.H., 2023. Adaptive rail transit network operations with a rollout surrogate-approximate dynamic programming approach. Transportation Research Part C, 148, 104021.

Psaraftis H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transportation Science, 14(2), 130-154.

Powell W.B., 2007. Approximate Dynamic Programming: Solving the curses of dimensionality. John Wiley & Sons.

Powell W.B., Simao H.P., Bouzaiene-Ayari B., 2012. Approximate dynamic programming in transportation and logistics: a unified framework. EURO Journal on Transportation and Logistics, 1(3), 237-284.

Parragh S.N., Sousa J., Almada-Lobo B., 2014. The dial-a-ride problem with split requests and profits. Transportation Science, 49(2), 311-334.

Posada M., Andersson H., Häll C.H, 2017. The integrated dial-a-ride problem with timetabled fixed route service. Public Transport, 9, 217-241.

Pimenta V., Quilliot A., Toussaint H., Vigo D., 2017. Models and algorithms for reliability-oriented Dial-a-Ride with autonomous electric vehicles. European Journal of Operational Research, 257(2), 601-613.

Perumal S.S.G., Larsen J., Lusby R.M., Riis M., Sørensen K.S., 2019. A matheuristic for the driver scheduling problem with staff cars. European Journal of Operational Research, 275, 280-294.

Qin X., Yang H., Wu Y., Zhu H., 2021. Multi-party ride-matching problem in the ride-hailing market with bundled option services. Transportation Research Part C, 131, 103287.

Qin Z., Zhu H., Ye J., 2022. Reinforcement learning for ridesharing: An extended survey. Transportation Research Part C, 144, 103852.

Qu Y., Bard J.F., 2013. The heterogeneous pickup and delivery problem with configurable vehicle capacity. Transportation Research Part C, 32, 1-20.

Özkan E., Ward A.R., Dynamic matching for real-time ride sharing. Stochastic Systems, 10(1), 29-70.

Ren J., Jin W., Wu W., 2022. Multi-objective optimization for multi-depot heterogeneous first-mile transportation system considering requests' preference ranks for pick-up stops. Transportmetrica A, 19(3), 2103205.

Sklar A., 1973. Random variables, joint distribution functions, and copulas. Kybernetika, 9(6), 449-460.

Sayarshad H.R., Chow J.Y.J., 2015. A scalable non-myopic dynamic dial-a-ride and pricing problem. Transportation Research Part B, 81, 539-554.

Shehadeh K.S., Wang H., Zhang P., 2021. Fleet sizing and allocation for on-demand last-mile transportation systems. Transportation Research Part C, 132, 103387.

Tellez O., Vercraene S., Lehuédé F., Péton O., Monteiro T., 2018. The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. Transportation Research Part C, 91, 99-123.

Vansteenwegen P., Melis L., Aktaş D., Montenegro B., Vieira F., Sörensen K., 2022. A survey on demand-responsive public bus systems. Transportation Research Part C, 137, 103573.

Wong K.I., Han A.F., Yuen C.W., 2014. On dynamic demand responsive transport services with degree of dynamism. Transportmetrica A, 10 (1), 55-73.

Wang X., Liu W., Yang H., Wang D., Ye J., 2019. Customer behavioural modelling of order cancellation in coupled ride-sourcing and taxi markets. Transportation Research Procedia, 38, 853-873.

Wang H., Yang H., 2019. Ridesourcing systems: A framework and review. Transportation Research Part B, 129, 122-155.

Weidinger F., Albiński S., Boysen N., 2023. Matching supply and demand for free-floating car sharing: On the value of optimization. European Journal of Operational Research, 308(3), 1380-1395.

Wu W., Lin Y., Liu R., Jin W., 2022. The multi-depot electric vehicle scheduling problem with power grid characteristics. Transportation Research Part B, 155, 322-347.

Wu W., Li Y., 2024. Pareto truck fleet sizing for bike relocation with stochastic demand: Risk-averse multi-stage approximate stochastic programming. Transportation Research Part E, 183, 103418.

Xu K., Saberi M., Liu W., 2022. Dynamic pricing and penalty strategies in a coupled market with ridesourcing service and taxi considering time-dependent order cancellation behaviour. Transportation Research Part C, 138, 103621.

Yang H., Qin X., Ke J., Ye J., 2020. Optimizing matching time interval and matching radius in on-demand ride-sourcing markets. Transportation Research Part B, 131, 84-105.

Yu G., Liu A., Zhang J., Sun H., 2021. Optimal operations planning of electric autonomous vehicles via asynchronous learning in ride-hailing systems. Omega, 103, 102448.

Zhang C., Zhu F., Wang X., Sun L., Tang H., Lv Y., 2020. Taxi demand prediction using parallel multi-task learning model. IEEE Transactions on Intelligent Transportation Systems, 23(2), 794-803.

Zhou Y., Yang H., Ke J., Wang H., Li X., 2022. Competition and third-party platform-integration in ride-sourcing markets. Transportation Research Part B, 159, 76-103.

Zhu Z., Ke J., Wang H., 2021. A mean-field Markov decision process model for spatial-temporal subsidies in ride-sourcing markets. Transportation Research Part B, 150, 540-565.

Appendix A: Request data in the numerical test

No.	Submission time	Pick-up time window	Pick-up stop	Drop-off stop	The number of passengers	No.	Submission time	Pick-up time window	Pick-up stop	Drop-off stop	The number of passengers
1	7:15:39	7:50-7:59	19	6	4	60	11:44:12	12:00-12:09	7	8	2
2	7:17:13	7:40-7:49	18	16	4	61	11:46:08	12:00-12:09	13	9	2
3	7:22:05	7:50-7:59	10	12	4	62	11:51:03	12:10-12:19	14	8	4
4	7:22:16	7:40-7:49	24	9	4	63	12:33:17	12:50-12:59	22	13	4
5	7:23:27	7:40-7:49	8	24	1	64	12:51:55	13:10-13:19	9	20	1
6	7:26:17	7:40-7:49	21	1	1	65	13:00:55	15:10-15:19	14	12	1
7	7:30:54	7:50-8:06	15	6	2	66	14:19:10	14:30-14:39	10	15	4
8	7:31:01	8:30-8:39	12	9	1	67	14:22:50	14:40-14:49	3	8	2
9	7:31:44	7:50-7:59	2	10	2	68	15:01:59	15:20-15:29	1	15	4
10	7:31:46	7:50-7:59	24	2	3	69	15:06:18	15:20-15:29	7	14	3
11	7:32:27	8:00-8:09	22	23	1	70	15:15:08	15:30-15:39	15	2	2
12	7:35:22	7:50-7:59	10	24	1	71	15:26:20	15:40-15:49	6	3	3
13	7:36:20	7:50-7:59	13	12	4	72	15:36:33	15:50-15:59	8	19	4
14	7:41:02	8:00-8:09	20	6	2	73	15:57:13	17:00-17:09	11	22	4
15	7:45:09	8:00-8:09	11	13	4	74	15:59:39	17:20-17:29	14	7	2
16	7:46:24	8:10-8:19	5	20	3	75	16:00:22	16:30-16:39	11	19	2
17	7:47:23	8:30-8:39	17	4	4	76	16:41:12	17:20-17:29	22	13	1
18	7:49:14	8:20-8:29	3	9	3	77	16:51:06	17:20-17:29	11	16	1
19	7:49:52	8:00-8:09	2	11	1	78	16:52:58	17:10-17:19	6	12	2
20	7:50:52	8:40-8:49	16	21	3	79	16:55:39	17:20-17:29	24	11	4
21	7:50:59	8:10-8:19	13	20	3	80	17:05:45	17:20-17:29	1	22	3
22	7:52:50	8:40-8:49	19	5	3	81	17:06:40	17:20-17:29	22	13	3
23	7:54:21	8:30-8:39	24	18	3	82	17:31:31	17:50-17:59	20	15	3
24	7:58:19	8:10-8:19	13	20	1	83	17:38:40	18:30-18:39	12	21	2
25	7:58:32	8:20-8:29	23	13	1	84	17:47:23	18:10-18:19	10	2	4
26	8:00:29	8:20-8:29	16	5	4	85	17:47:41	18:00-18:09	3	22	3
27	8:01:57	8:30-8:39	6	14	4	86	17:49:14	18:00-18:09	6	4	1
28	8:02:34	8:30-8:39	16	20	1	87	17:52:21	18:10-18:19	24	13	1
29	8:04:59	8:30-8:39	2	14	1	88	17:54:10	18:10-18:19	4	14	4
30	8:05:31	8:30-8:39	5	14	4	89	17:55:03	18:30-18:39	22	20	4
31	8:07:36	8:30-8:39	8	10	3	90	17:56:35	18:10-18:19	6	3	1
32	8:10:01	8:30-8:39	17	9	4	91	18:01:21	18:20-18:29	12	19	4
33	8:10:04	8:30-8:39	16	15	1	92	18:10:19	18:40-18:49	3	15	4
34	8:12:15	8:30-8:39	3	9	1	93	18:14:04	18:30-18:39	17	1	2
35	8:16:59	8:40-8:49	14	17	3	94	18:15:08	18:30-18:39	24	16	2
36	8:23:08	8:40-8:52	5	12	3	95	18:23:53	19:00-19:09	7	17	2
37	8:23:39	8:40-8:49	5	12	4	96	18:31:01	18:50-18:59	1	10	3
38	8:32:55	8:50-8:59	5	18	1	97	18:34:47	18:50-18:59	17	7	3
39	8:34:35	8:50-8:59	2	14	3	98	18:42:47	19:00-19:09	7	18	4
40	8:45:24	9:10-9:19	16	7	3	99	18:43:16	19:00-19:09	10	9	3
41	8:46:27	9:00-9:09	19	14	4	100	18:55:35	19:10-19:19	6	18	3
42	8:50:39	9:10-9:19	1	11	4	101	18:56:22	19:10-19:19	5	23	1
43	8:51:33	9:10-9:19	23	21	2	102	19:03:41	19:20-19:29	23	1	2
44	8:52:41	9:10-9:19	8	24	3	103	19:06:34	19:20-19:29	1	19	3
45	8:54:19	9:20-9:29	6	1	2	104	19:14:56	19:30-19:39	24	23	2
46	8:56:43	9:20-9:29	20	11	3	105	19:15:02	19:30-19:39	7	5	2
47	8:58:39	9:30-9:39	3	19	1	106	19:21:35	19:40-19:49	14	13	4
48	9:05:27	9:20-9:29	23	24	2	107	19:21:36	19:40-19:49	5	19	1
49	9:11:18	9:30-9:39	24	19	2	108	19:28:50	19:40-19:49	6	10	3
50	9:12:38	9:30-9:39	18	22	1	109	19:34:15	19:50-19:59	5	12	1
51	9:15:11	9:30-9:39	23	18	1	110	20:06:42	20:40-20:49	10	2	2
52	9:25:23	9:40-9:49	24	6	4	111	20:07:38	20:20-20:29	23	3	3
53	9:32:26	10:10-10:19	6	19	4	112	20:09:53	20:20-20:29	10	15	4
54	9:37:00	9:50-9:59	11	10	1	113	20:17:41	20:30-20:39	22	12	4
55	9:59:09	10:10-10:19	23	11	4	114	20:19:00	20:30-20:39	7	19	1
56	10:05:41	10:50-10:59	22	8	1	115	20:31:26	20:50-20:59	10	9	1
57	10:37:04	10:50-10:59	17	15	1	116	20:32:54	20:50-20:59	5	3	2
58	10:39:39	11:00-11:09	7	1	2	117	20:32:58	20:50-20:59	6	23	3
59	10:52:23	11:10-11:19	8	15	3	118	20:37:18	20:50-20:59	8	19	2