Research Article

Modeling and Optimization of Multiaction Dynamic Dispatching Problem for Shared Autonomous Electric Vehicles

Ning Wang and **Jiahui Guo**

School of Automotive Studies, Tongji University, Shanghai, China

Correspondence should be addressed to Jiahui Guo; 1710971@tongji.edu.cn

Received 25 August 2021; Accepted 15 October 2021; Published 15 November 2021

Academic Editor: Zhihong Yao

Copyright © 2021 Ning Wang and Jiahui Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The fusion of electricity, automation, and sharing is forming a new Autonomous Mobility-on-Demand (AMoD) system in current urban transportation, in which the Shared Autonomous Electric Vehicles (SAEVs) are a fleet to execute delivery, parking, recharging, and repositioning tasks automatically. To model the decision-making process of AMoD system and optimize multiaction dynamic dispatching of SAEVs over a long horizon, the dispatching problem of SAEVs is modeled according to Markov Decision Process (MDP) at first. Then two optimization models from short-sighted view and farsighted view based on combinatorial optimization theory are built, respectively. The former focuses on the instant and single-step reward, while the latter aims at the accumulative and multistep return. After that, the Kuhn–Munkres algorithm is set as the baseline method to solve the first model to achieve optimal multiaction allocation instructions for SAEVs, and the combination of deep Q-learning algorithm and Kuhn–Munkres algorithm is designed to solve the second model to realize the global optimization. Finally, a toy example, a macrosimulation of 1 month, and a microsimulation of 6 hours based on actual historical operation data are conducted. Results show that (1) the Kuhn–Munkres algorithm ensures the computational effectiveness in the large-scale real-time application of the AMoD system; (2) the second optimization model considering long-term return can decrease average user waiting time and achieve a 2.78% increase in total revenue compared with the first model; (3) and integrating combinatorial optimization theory with reinforcement learning theory is a perfect package for solving the multiaction dynamic dispatching problem of SAEVs.

1. Introduction

Three revolutions of Electrification, Automation, and Sharing are booming in current urban transportation [1]. The fusion of L4/L5 level autonomous driving, electric vehicles, and shared mobility mode is forming a new Autonomous Mobility-on-Demand (AMoD) system [2, 3]. In the AMoD system, SAEVs can automatically pick up and deliver passengers from origin to destination, drive to the nearby charging station/pile for electricity supplement, reposition to hotspots with low vehicle supply and high trip demand, and park on the road waiting for the new assignment [4–6]. On one hand, Shared Autonomous Electric Vehicles (SAEVs) help reduce environmental pollution, carbon emissions, and traffic congestion in cities [7,8]. On the other hand, SAEVs can provide on-demand mobility service to satisfy the immediate trip demand of users [9,10].

In this paper, the multiagent multiaction dynamic dispatching problem based on SAEVs fleet is the main research theme, which corresponds to the optimal decision-making process of assigning different SAEVs to serve the passenger's trip request, drive to the charging station/pile for recharging, and head for hotspots to supply vehicles in advance [11]. Previously, few scholars focused on the dispatching problem of SAEVs since it is a new emerging topic in recent years. The most studied area is the static or dynamic relocation problem of the electric car-sharing system based on manned vehicles [12–17]. However, these studies almost focus on the pickup and delivery task for users and barely consider the recharging task of electric vehicles and repositioning task of redundant vehicles [18–20]. Meanwhile, methods involved in the above studies mainly include nonlinear programming models and solving algorithms with high time complexity, which ignores the computational efficiency in large-scale

application [13, 21, 22]. Hence, the existing methods are not suitable for the future AMoD system. A new method for large-scale multiaction dispatching of SAEVs comprehensively considering delivery task, parking task, recharging task, and repositioning task is in urgency.

Our goal is to model the decision-making process of the AMoD system and optimize multiaction dynamic dispatching of SAEVs over a long horizon (e.g., several days or a month) instead of a static dispatching. It relies on a proper mathematical modeling method to illustrate the whole multiaction dispatching process of SAEVs, an optimization model to figure out the optimal objective and constraints, and an efficient solving algorithm to realize fast processing to achieve optimal task allocation instructions for SAEVs in large-scale application.

To accomplish this, the MDP framework including agent, state, action, and reward is adopted to model the multiagent multiaction dynamic dispatching problem of SAEVs at first. Then, combinatorial optimization method is employed to establish the multiaction dispatching optimization model, and the optimal task allocation instructions are solved by the KM algorithm. After that, to realize a long-term and global optimization, The Bellman Equation from reinforcement learning theory is used to transform the instant and single-step reward into an accumulative and multistep return, which is represented as a new match value between each SAEV and each task to update the above combinatorial optimization model. The deep Q-learning and KM algorithms are combined to solve the new optimization model to achieve a better task assignment instruction considering the future impact. Finally, a toy example based on simple data and two dispatching simulators based on actual vehicle trajectory and trip request data of Didi Chuxing platform (<https://gaia.didichuxing.com>) and charging pile information (<http://admin.bjev520.com/jsp/beiqi/pcmap/do/index.jsp>) are conducted to verify the effectiveness of the above methods. Results show that the latter optimization model considering accumulative and multistep return can bring an improvement compared with the former one.

We overcome several practical issues to make the proposed methods suitable for large-scale application of future SAEVs fleet, including computational efficiency and multi-action coordination. The contribution of this research is summarized as follows:

- (1) To the best of our knowledge, this is the first batch of work to study multiaction dynamic dispatching problem of SAEVs, which comprehensively considers delivery task, recharging task, and repositioning task simultaneously. Even though Al-Kanj [23] also studied the related problem, which involved the decisions of assigning orders to cars, recharging batteries, repositioning, and parking the vehicles, our research adopts a bipartite graph modeling to achieve a faster solving process of optimal dispatching scheme.
- (2) By modeling multiaction dynamic dispatching problem as a sequential decision-making problem through MDP framework and taking the

accumulative and multistep reward into account based on the Bellman Equation, the proposed methods belong to the category of reinforcement learning. Results show that reinforcement learning can be applied into large-scale real-time AMoD system very well.

- (3) A stochastic integer linear programming model is established based on combinatorial optimization theory and solved by the combination of Q-learning algorithm and KM algorithm to achieve the optimal multiaction allocation instructions for SAEVs. Results show an improvement over the baseline KM algorithm, and it turns out that integrating combinatorial optimization theory with reinforcement learning theory is a perfect package for solving the multiaction dynamic dispatching problem of SAEVs.

The rest of the paper is organized as follows: Section 2 is a literature review of the dispatching problem of the Mobility-on-Demand system. Section 3 provides the analyzing framework and details the methods, including problem description, problem assumption, and mathematical formulation. Case studies including a toy model and two dispatching simulators are then described in Section 4. Section 5 concludes the research results and discusses the future work.

2. Literature Review

Generally, the fleet operation process of SAEVs is usually involved with 8 subproblems, including trip demand, fleet size, traffic assignment, vehicle assignment, vehicle distribution, pricing, charging, and parking [24]. Vehicle assignment, charging and parking assignment, and vehicle repositioning are the main focus when studying the multiagent multiaction dynamic dispatching problem based on SAEVs in AMoD system in this paper. Most scholars study these three subproblems separately (as shown in Table 1) while this research integrates the above three subproblems into one methodology framework by combining deep reinforcement learning and combinatorial optimization method.

2.1. Vehicle Assignment. Vehicle assignment means assigning vehicles to the customers, and then vehicles execute pickup and delivery tasks to satisfy customers' trip request. Certain rules, heuristics, or precise optimization algorithm are three main methods to solve this problem. For modeling dynamic process of vehicle assignment, a rule-based vehicle assignment method is usually implemented [26, 28, 44–48]. The most widely used rule is assigning the nearest vehicle to the user request. Further optimization models and solving algorithms are developed to optimize vehicle assignment performance. Liang et al. [39] modeled a dial-a-ride problem of ride-sharing SAEVs in urban road network into an integer nonlinear programming model and designed a customized Lagrangian relaxation algorithm to solve the optimal assignment scheme. However, this methodology is not practice-ready due to the computation

TABLE 1: Summary of literature review.

	Vehicle assignment	Charging and parking assignment	Vehicle repositioning	Methods
Fagnant and Kockelman [7]			✓	Simulation
Zhang and Pavone [25]			✓	Optimization
Chen et al. [26]		✓		Optimization
Azevedo et al. [27]		✓		Optimization
Zhang and Guhathakurta [28]		✓		Optimization
Alonso-Mora et al. [29]			✓	Optimization
Bauer et al. [30]		✓		Optimization
Babicheva et al. [31]			✓	Optimization
Farhan et al. [32]	✓			Optimization and simulation
Rossi et al. [33]	✓		✓	Optimization
Iacobucci et al. [34]		✓		Optimization
Vosooghi et al. [6]			✓	Optimization
Dandl et al. [35]			✓	Optimization
Iacobucci et al. [36]		✓		Optimization
Jones and Leibowicz [37]		✓		Optimization
Mao et al. [38]			✓	Reinforcement learning
Liang et al. [39]	✓			Optimization
Zhang et al. [40]		✓		Simulation
Al-Kanj et al. [23]	✓	✓	✓	Reinforcement learning and optimization
Shi et al. [41]	✓			Deep reinforcement learning and optimization
Melendez et al. [42]	✓	✓		Optimization
Zhang and Chen [43]		✓		Simulation

time and computation gap between the upper and lower bounds. This is also a typical problem in other studies [32, 49–51]. To make the method more practical, Shi et al. [41] developed a reinforcement learning-based algorithm to operate an electric vehicle fleet, which can also be referenced to apply to the SAEVs assignment problem. The goals of designing the reward function are to minimize customer waiting time, electricity cost, and operational costs of the vehicles. A deep feed-forward neural network (FNN) is parameterized to approximate the state-value function, and the KM algorithm with a time complexity of $O(n^3)$ is adopted to solve the optimal dispatching results.

2.2. Charging and Parking Assignment. Charging and parking assignment refers to monitoring real-time battery levels of SAEVs and conducting corresponding strategies to assign vehicles to charging piles or parking lots [34, 52]. Regarding charging assignment, Chen et al. [26] insisted that the charging vehicles are not allowed to undock and serve a new trip request but Bauer et al. [30] believed that still-charging vehicles are allowed to serve a new trip request. Iacobucci et al. [34, 53] developed a simulation methodology for evaluating a Shared Autonomous Electric Vehicle system interacting with passengers and charging at designated charging stations using a heuristic-based charging strategy and used electricity price information for optimizing vehicle charging in a mixed-integer optimization model by adding charging constraints over longer time scales. Jones and Leibowicz [37] used an energy optimization model to assess the impact of charging SAEVs at

times that are optimal for the energy system. Zhang et al. [40] adopted an agent-based simulation model, called BEAM, to describe the complex behaviors of both passengers and AMoD systems in urban cities. BEAM simulates the driving, parking, and charging behaviors of the SAEV fleet with range constraints and identifies times and locations of their charging demands. Melendez et al. [42] incorporated power network involving power purchase, real-time price spikes, arbitrage with battery banks, and solar generation into SAEV fleet operation planning and constructed a mixed-integer linear programming model to solve the optimal delivery and charging task decisions. To facilitate charging decisions, Basso et al. [54] propose a probabilistic Bayesian machine learning approach for predicting the expected energy consumption of electric vehicles. MAPE of this model decreases from 7.95% to 3.59% compared with prior forecast models. Regarding parking assignment, Azevedo et al. [27] used an optimization algorithm (facility location problem) to locate parking stations, wherein vehicle charging is also possible. Zhang and Guhathakurta [28] minimized cost by routing idle vehicles to low-cost parking areas. Al-Kanj et al. [23] introduced SAEVs into the ride-hailing system and combined the Markov decision process and one-stage combinatorial optimization method to realize the real-time optimal decision-making of charging and parking assignment.

2.3. Vehicle Repositioning. Vehicle repositioning, also referred as “vehicle rebalancing or redistribution,” is used to reposition excess vehicles from low demand areas to high

demand areas when modeling on-demand SAEVs service. Fagnant and Kockelman [7] designed an agent-based simulation model for SAEVs operations and introduced vehicle rebalancing into this simulation process. Results show vehicle rebalancing may save 10 times the number of cars needed for self-owned personal-vehicle travel. Vosooghi et al. [6] concluded that vehicle repositioning has a significant effect on service performance, such as modal share and fleet usage. In modern literature, reactive methods such as nearest neighbours are commonly used, but Babicheva et al. [31] compared 6 different ways to apply vehicle repositioning assignment and proposed a new index-based proactive redistribution (IBR) algorithm based on predicted near-future demand at stations. A linear programming model for vehicle redistribution is adopted by Zhang and Pavone [25] and Alonso-Mora et al. [29]. Rossi et al. [33] suggested that the problems of vehicle assignment and rebalancing vehicles can be decoupled and develop a computationally efficient routing and rebalancing algorithm for SAEVs. The rebalancing optimization problem is modeled as the Minimum Cost Flow problem. Dandl et al. [35] emphasized the importance of trip demand forecast and concluded that accurate forecast quality can help redistribute vehicles among different regions. Mao et al. [38] modeled the repositioning task dispatching problem of SAEVs alone by MDP and achieved the optimal dispatching results by actor-critic strategy gradient network.

2.4. Research Gaps. Generally, current studies concerning the multiagent multiaction dynamic dispatching problem of SAEVs operation are not enough yet, and three research gaps remain to be settled in the future.

First, vehicle assignment, charging and parking assignment, and vehicle repositioning are mostly studied separately in previous research. Rossi et al. [33] studied the combination of vehicle assignment and vehicle repositioning, and Melendez et al. [42] combined vehicle assignment and charging assignment together. During the dynamic operation process, it is better to integrate vehicle repositioning into the decision-making process since vehicle redistribution in advance has a significant effect on mobility service performance [6].

Second, the current research barely balances operational income, user satisfaction, and electricity cost when deciding which vehicle should be assigned to the specified passenger. Besides, designing the matching weight of each vehicle-passenger pair from a long-term view is verified to be better than a short-sighted instant view. Hence, it is better to design a multidimensional accumulated reward function to represent the matching weight value between each vehicle and delivery task.

Third, to guarantee to be practical-ready of these methods for SAEVs' dynamic dispatching process, a combination of deep reinforcement learning and combinatorial optimization is a promising methodology framework. Thus far, some studies have verified the effectiveness of the above framework, but they are only applied in the field of ride-hailing rather than in the field of SAEVs fleet operation. How

to model the dynamic dispatching process comprehensively considering delivery, charging, and repositioning tasks together based on the reinforcement learning framework and how to solve the best charging, delivery, and repositioning assignment scheme by combinatorial optimization methods remain to be explored.

3. Methods

To assign different tasks, including delivery, parking, recharging, and repositioning, to SAEVs, MDP is adopted to model the operation process and to transform this process into a multiagent multiaction dynamic dispatching problem at first. Then, to achieve the optimal task allocation instructions for SAEVs, two optimization models with the objective of maximum economic income from the local view (instant and single-step reward) and global view (accumulative and multistep reward) are established, respectively. Accordingly, two algorithms are designed to solve the above optimization models so that the best dispatching scheme can be obtained at each time period as soon as possible. Finally, to test and verify the effectiveness of the optimization models and the performance of the solving algorithms, two case studies are designed, including a toy model with hypothetical data including 1 time slot, 4 agents, and 4 tasks, and two dispatching simulators covering 1 month, 43200 time slots in Chengdu, China, based on the historical data generated from Didi Chuxing platform. The analyzing framework is illustrated in Figure 1.

3.1. Problem Description and Assumption

3.1.1. Problem Description. Shared automotive electric vehicles (SAEVs) are a free-floating L4/L5 level fleet that executes four tasks, including delivery, parking, recharging, and repositioning. (1) Delivery task means the SAEV picks up and delivers the passenger to the destination. (2) Parking task indicates the SAEV idles on the road waiting for the next round of task distribution. (3) Recharging task represents the SAEV drives to the nearby charging station or pile for recharging. (4) Repositioning task denotes the SAEV is instructed to drive to a specific area for vehicle replenishment.

The problem to be solved in this research is how different tasks should be assigned to SAEVs dynamically in the best way at each time slot. For practical application in the future robotaxi fleet operation, SAEVs should make decisions themselves fast and optimally by following a set of multi-action dispatching algorithms, which is exactly the purpose of this research.

3.1.2. Problem Assumption. The proposed multiaction dispatching optimization model and algorithm is developed based on the following assumptions:

- (1) SAEVs are assumed to be a fleet with L4/L5 level autonomous driving technology. SAEVs can follow the instruction of task allocation results to execute four tasks autonomously without drivers.

FIGURE 1: Method and analyzing framework.

- (2) When executing the delivery task, the position of pickup location and destination is assumed to be fixed following the initial trip request information without any change.
- (3) When executing the parking task, the parking strategy is assumed to be predetermined: the SAEV keeps still in its original position and waits for the next round of task allocation instruction.
- (4) When executing the recharging task, the SAEV is assumed to drive to the nearby predesigned charging station/pile for recharging without any help of manpower; that is, the whole charging process is conducted by the SAEV itself.
- (5) When executing the repositioning task, specific hotspots with high trip demand and low vehicle supply at different time periods and the specific quantity of surplus or missing vehicles are predetermined.

3.2. Mathematical Formulation

3.2.1. Model Parameters. Based on the problem description and assumptions above, firstly, a mathematical model will be conducted by following MDPs to describe the multi-action dispatching process realistically; then, two optimization models from the perspective of global and local respectively shall be established to realize the reward-maximum oriented task allocation instructions; and finally, two corresponding algorithms will be designed separately to solve the above two optimization models fast and efficiently. Model parameters are shown in Table 2.

3.2.2. Model Framework. The multi-action dispatching process of SAEVs is modeled based on MDPs [20]. Each SAEV behaves as an “agent” in the “environment.” The spatio-temporal status, including geographic position, time, and state of charge (SOC) of SAEV, is set as “state.” Four tasks, including delivery, parking, recharging, and repositioning, are

defined as four “actions,” respectively. In each round of task allocation, each SAEV will achieve a “reward” representing task income, carbon emission savings, or user satisfaction. A “policy” should be determined to decide how to allocate different tasks for SAEVs in the best way.

(1) *Agent and Environment.* Each independent SAEV is modeled as an agent, and the environment contains all the information, including the layout of charging stations/piles, parking spots, users’ trip orders, and other agents.

(2) *State.* The state of the SAEV is represented by location, time, and battery level. It is defined as a four-dimensional vector indicating universal time coordinated (UTC), geographic position, and state of charge (SOC). Formally, we define $s = (t, \text{lng}, \text{lat}, \text{soc}) \in S$, where $t \in T$ is the time index, (lng, lat) is the real location of the SAEV at time index t , and soc represents the remaining battery volume of the SAEV at time index t .

(3) *Action.* The action of SAEV includes delivery, parking, recharging, and repositioning. $a = (a_1, a_2, a_3, a_4)$ defines four kinds of the task of SAEVs, including delivery, parking, recharging, and repositioning. a_1 defines the delivery task, a_2 represents the parking task, a_3 denotes the recharging task, and a_4 implies the repositioning task.

(4) *Reward.* $r = (r_1, r_2, r_3, r_4)$ defines four types of the reward of SAEVs for executing each task (action), which is quantified by task income highly related to driving distance $\text{dis}_r \in \text{DIS}$, elapsed time $\text{ela}_r \in \text{ELA}$, battery energy consumption $\text{con}_r \in \text{CON}$, battery capacity $\text{bat}_r \in \text{BAT}$ of each SAEV, and charging volume $\text{vol}_r \in \text{VOL}$ of each charging process.

r_1 represents the expected return when the SAEV executes the delivery task. r_1 is linearly related to driving distance $\text{dis}_r \in \text{DIS}$ and elapsed time $\text{ela}_r \in \text{ELA}$. Assume SAEV drives from state $s_1 = (t_1, \text{lng}_1, \text{lat}_1, \text{soc}_1)$ to the next state $s_2 = (t_2, \text{lng}_2, \text{lat}_2, \text{soc}_2)$, picks up the passenger, then continues to drive to the next state $s_3 = (t_3, \text{lng}_3, \text{lat}_3, \text{soc}_3)$,

TABLE 2: Description of model parameters.

Notation	Definition
lng	Longitude
lat	Latitude
	The elapsed time of executing any task for a SAEV
$\widetilde{R}_{ip}$	$\widetilde{R}_{ip}$ implies the weighed value (or the reward) from vehicle i to targeted area p

and arrives at the destination. The formula of $r1$ is defined in equations (1)–(4). Following Mercator Projection, dis_r is simplistically determined by equations (2) and (4). It should be noted that dis_{r1} is only related to the distance between the pickup point $s2$ and destination point $s3$:

$$r1 = \alpha \times dis_{r1} + \beta \times ela_{r1}, \quad (1)$$

$$dis_{r1} = mer(lng2, lat2, lng3, lat3), \quad (2)$$

$$ela_{r1} = t3 - t1, \quad (3)$$

$$mer(lng, lat, lng', lat') = \left[\frac{\sin(lat' - lat)}{2} \right]^2 + \cos(lat) \times \cos(lat') \times \left[\frac{\sin(lng' - lng)}{2} \right]^2. \quad (4)$$

$r2$ denotes the expected return when the SAEV executes the parking task. Following the principle of opportunity cost, $r2$ is negative because if the SAEV was assigned to a delivery task instead, the return would be a positive value highly linearly related to driving distance dis_r and elapsed time ela_r . Assume the SAEV parks from state $s1 = (t1, lng1, lat1, soc1)$ to the next state $s2 = (t2, lng2, lat2, soc2)$ before the next round of task allocation instruction. The formula of $r2$ is defined in equations (5)–(7) and (4):

$$r2 = -1 \times (\alpha \times dis_{r2} + \beta \times ela_{r2}), \quad (5)$$

$$dis_{r2} = mer(lng1, lat1, lng2, lat2), \quad (6)$$

$$ela_{r2} = t2 - t1. \quad (7)$$

$r3$ implies the expected return when the SAEV executes the recharging task. If soc of the SAEV is lower than 20%, the SAEV would be assigned to the recharging task compulsively, or the SAEV would be subject to the global optimization task allocation results. The recharging task is divided into two parts: (1) the SAEV drives from its real-time location (state $s1 = (t1, lng1, lat1, soc1)$) to the charging station/pile (state $s2 = (t2, lng2, lat2, soc2)$) and (2) the SAEV stays in the charging station/pile for recharging until the soc reaches 90% (state $s3 = (t3, lng2, lat2, soc3)$, $soc3 = 90\%$). Following the principle of opportunity cost, part 1 will cause a negative reward which is linearly related to driving distance dis_r and elapsed time ela_r , part 2 will also cause a negative reward,

including the presupposed delivery task income, which is linearly related to elapsed time ela_r , and charging cost which is linearly related to battery capacity bat_r of this SAEV and charging volume vol_r of this charging process. The formula of $r3$ is defined in equations (8)–(12) and (4):

$$r3 = -1 \times \left[\left(\alpha \times dis_{r3}^{part1} + \beta \times ela_{r3}^{part1} \right) + \left(\beta \times ela_{r3}^{part2} + \theta \times vol_{r3}^{part2} \right) \right], \quad (8)$$

$$dis_{r3}^{part1} = mer(lng1, lat1, lng2, lat2), \quad (9)$$

$$ela_{r3}^{part1} = t2 - t1, \quad (10)$$

$$ela_{r3}^{part2} = t3 - t2, \quad (11)$$

$$vol_{r3}^{part2} = bat_{r3} \times (soc3 - soc2). \quad (12)$$

$r4$ indicates the expected return when the SAEV executes the repositioning task. Following the principle of opportunity cost, driving to the area (a regular hexagon with radius rad) with high trip demand and low vehicle supply will cause a certain cost for the SAEV in the short term. Instead of repositioning, the SAEV could be assigned a delivery task and earn a positive reward. Hence, if the SAEV was assigned a repositioning task, it would drive from its real-time location (state $s1 = (t1, lng1, lat1, soc1)$) to the corresponding area for replenishment. Set the central point (state $s2 = (t2, lng2, lat2, soc2)$) of the area as the destination of repositioning task. Moreover, please note that it is unnecessary for the SAEV to drive to the central point. Instead, crossing into the boundary of the regular hexagon (state $s3 = (t3, lng3, lat3, soc3)$) is enough. Hence, ζ is defined as a constant reward from the first point when crossing into the area to the central point of this area, and ζ should also be considered into $r4$. The formula of $r4$ is defined in equations (13)–(16) and (4):

$$r4 = -1 \times (\alpha \times dis_{r4} + \beta \times ela_{r4} - \zeta), \quad (13)$$

$$dis_{r4} = mer(lng1, lat1, lng2, lat2), \quad (14)$$

$$ela_{r4} = t2 - t1, \quad (15)$$

$$\zeta = [\alpha \times rad + \beta \times (t2 - t3)]. \quad (16)$$

(5) *Discount Factor.* The discount factor $\gamma \in (0, 1]$ controls the degree of how far the MDP looks into the future. It is beneficial to use a small discount factor as long horizons will introduce a

large variance on the value function. It is worth noting that, under this setting, the reward should also be discounted. For a task which lasts for T time slots with reward r and a discount factor γ , the final reward is r_γ given as follows:

$$r_\gamma = \sum_{t=0}^{T-1} \left(\gamma^t \times \frac{r}{T} \right). \quad (17)$$

(6) *Policy*. Policy π represents a strategy which is used to decide the specific task allocation instruction for SAEVs at each decision time slot. It will be determined by the combinatorial optimization method in this paper.

(7) *Value Function*. Since $r = (r_1, r_2, r_3, r_4)$ defines the instant and single-step reward, it is hard to reveal the accumulated and multistep reward when the agent chooses different actions. Hence, the value function based on the Bellman Equation is introduced into this research to design the long-term return of each action (task reward in a global view). The formula of value function V is defined in the following equations:

$$V(s_t) = \sum_{i=t+1}^T (\gamma^{i-t-1} \times r_i), \quad (18)$$

$$V(s_t) = r_t + \gamma \times V(s_{t+1}). \quad (19)$$

(8) *State Transition*. Before trip assignment, the transition of state will be predetermined by assuming different kinds of SAEV-action matches. When trip assignment is completed by solving the optimization model below, the transition of state then will be finally determined.

3.2.3. Optimization Model. Based on the MDP framework above, a combinatorial optimization model will be constructed in this section to achieve the best policy π in order to decide the best task allocation (action selection) instructions for SAEVs in a global view.

Given a bipartite graph $G = (V, E)$, V is the set of vertexes including two subsets V_1 and V_2 . V_1 is the set of SAEVs (agents) and V_2 is the set of tasks. E is the set of directed lines $e \in E$ from $v_1 \in V_1$ to $v_2 \in V_2$. $w(e)$ represents

s, s' represent state of each SAEV
 a_1, a_2, a_3, a_4 represent 4 different actions
 r_1, r_2, r_3, r_4 represent corresponding instant reward of each action

FIGURE 2: Bipartite graph of task allocation instructions for SAEVs.

the weighted value of directed line e . Assume M as a match between V_1 and V_2 among G , each SAEV (agent) can only match one task, and all the SAEVs should be assigned with corresponding tasks in each round of task allocation process. Define W as the final weighted value of the match M , and it is the sum of $w(e)$ of all the directed lines. For instance, Figure 2 illustrates the task allocation instructions of 8 SAEVs based on bipartite graph theory and MDP framework.

In the bipartite graph, if W reaches the maximum under the condition of match M , match M can be identified as the best policy π . Combinatorial optimization model will be constructed to solve the best match M , that is, the best policy π . The mathematical model of the optimization process is shown as follows:

The maximum of reward is set as the optimization objective mainly representing the total task economic income. $x_{ij}, x_{im}, x_{in}, x_{ip}$ represent whether the SAEV drives from state i to state j while the agent executes the delivery task, from state i to state m to execute parking task, from state i to state n to execute the recharging task, and from state i to state p to execute the repositioning task, respectively. $x_{ij}, x_{im}, x_{in}, x_{ip}$ are decision variables and are all 0-1 variable. The objective function is shown as follows:

$$\text{Max } W = \sum_{i=1}^I \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^P (\widetilde{R}_{ij} \cdot x_{ij} + \widetilde{R}_{im} \cdot x_{im} + \widetilde{R}_{in} \cdot x_{in} + \widetilde{R}_{ip} \cdot x_{ip}). \quad (20)$$

I, J, M, N, P denote the number of states of SAEVs, passengers, parking positions, charging stations, and targeted areas, respectively. I_s, J_s, M_s, N_s, P_s denote the set of states of SAEVs, passengers, parking positions, charging stations, and targeted areas, respectively. $\widetilde{R}_{ij}, \widetilde{R}_{im}, \widetilde{R}_{in}, \widetilde{R}_{ip}$ imply the weighed value of $w(e)$ (or the reward) from vehicle state i to passenger state j , from vehicle state i to parking position state m , from

vehicle state i to charging station state n , and from vehicle state i to targeted area state p , respectively. $\widetilde{R}_{ij}, \widetilde{R}_{im}, \widetilde{R}_{in}, \widetilde{R}_{ip}$ are supposed to be calculated from two different perspectives: one is the global view and the other is the local view.

From the perspective of local view, $\widetilde{R}_{ij}, \widetilde{R}_{im}, \widetilde{R}_{in}, \widetilde{R}_{ip}$ represent the instant and single-step return of $x_{ij}, x_{im}, x_{in}, x_{ip}$, and they will be determined by following

equations (1)–(19). However, this perspective may cause myopia since this round of task allocation result may only be suitable for this time period, not the following several stages.

Instead, from the perspective of global view, $\widetilde{R}_{ij}$, $\widetilde{R}_{im}$, $\widetilde{R}_{in}$, $\widetilde{R}_{ip}$ represent the accumulative and multistep return of x_{ij} , x_{im} , x_{in} , x_{ip} , and they can be calculated by combining the instant reward (the calculation process follows equations (1)–(16)) and the long-term value function

$$\text{Max } W = \sum_{i=1}^I \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^P \{ [r_{ij} + \gamma \times V(j)] \times x_{ij} + [r_{im} + \gamma \times V(m)] \times x_{im} + [r_{in} + \gamma \times V(n)] \times x_{in} + [r_{ip} + \gamma \times V(p)] \times x_{ip} \}. \quad (21)$$

Constraint 1: $x_{ij}, x_{im}, x_{in}, x_{ip} \in \{0, 1\}$,

Constraint 2: $r_{ij}, r_{im}, r_{in}, r_{ip} \in [-\infty, +\infty]$,

Constraint 3: $\forall i \in I_s, \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^P (x_{ij} + x_{im} + x_{in} + x_{ip}) = 1$,

Constraint 4: $\forall j \in J_s, \sum_{i=1}^I x_{ij} = 1$,

Constraint 5: $\forall m \in M_s, \sum_{i=1}^I x_{im} = 1$,

Constraint 6: $\forall n \in N_s, \sum_{i=1}^I x_{in} = 1$,

Constraint 7: $\forall p \in P_s, \sum_{i=1}^I x_{ip} = 1$.

3.2.4. Solving Algorithm. Based on the two optimization models above, two solving algorithms are separately designed to achieve the best task allocation instructions for SAEVs at each time period. Model 1 is a typical integer linear programming problem (ILPP). The Kuhn–Munkres (KM) algorithm will be adopted as the final solving algorithm for model 1. Model 2 is a stochastic combinatorial optimization model due to the randomness and nonlinearity of value function $V(s_t)$. A new algorithm combining the KM algorithm with Q-learning algorithm will be designed to solve model 2.

(1) Alternating Path. The path starting from the unmatched vertex, going through the unmatched line, matched line, unmatched line, and so on, is called an alternating path.

(2) Complete Matching. Among match M of graph G , if the number of vertices $|V_1| \leq |V_2|$ and the number of matched lines $|M| = |V_1|$, this match M will be called complete matching.

The KM algorithm mainly serves for determining the maximum weight matching under complete matching in the bipartite graph $G = (V, E)$. Assume there is vertex $i \in V_1$ and vertex $j \in V_2$. $\forall i \in V_1$ and $\forall j \in V_2$, $V_1^i + V_2^j \geq w(e)^{ij}$.

Theorem 1. Define a set made up of directed lines from vertex $i \in V_1$ to vertex $j \in V_2$ meeting the condition of $V_1^i + V_2^j = w(e)^{ij}$ as S . If there is a match that is the complete matching of set S , this match must also be the maximum complete matching of graph G .

Following Theorem 1, the core of the KM algorithm applying into model 1 is to search for the complete matching

value (the calculation process follows equations (18) and (19)). Therefore, two optimization models from two perspectives (myopic and global) are compared. The optimization mathematical model 1 from myopic perspective is listed in Appendix, and model 2 from global (long horizon) perspective can be listed as follows:

Mathematical model 2: set accumulative, multistep reward as a weighted value:

of set S and view it as the final optimal matching to instruct the task allocation of SAEVs. The flowchart of the KM algorithm is as follows (Table 3):

For the purpose of solving model 1, the KM algorithm above can realize an exact solving with the time complexity of $o(n^3)$, which is a fast and efficient algorithm for practical application. However, to solve model 2, the KM algorithm is not enough since the value function $[r_{ij} + \gamma \times V(j)]$, $[r_{im} + \gamma \times V(m)]$, $[r_{in} + \gamma \times V(n)]$, $[r_{ip} + \gamma \times V(p)]$ is not predetermined and constant, which cannot be quantified as a weighted value.

To solve this problem, an approaching method called the deep Q-learning algorithm is put forward as the first step to transfer the nonlinear and stochastic value function into an approximately determined accumulative and multistep reward. Next, the KM algorithm will be adopted to solve the optimal task allocation instructions based on the value function.

Before conducting the deep Q-learning algorithm, an experience trajectory including information of SAEVs executing different tasks at different time slots should be built based on the historical fleet operational data. Based on the experience trajectory, value function $V(s)$ will be updated iteratively by following the temporal difference (TD) principle in the following equation:

$$v(s_t) \leftarrow v(s_t) + \sigma \left[\frac{r_{t+1} + \gamma \times v(s_{t+1})}{\text{Target}} - v(s_t) \right]. \quad (22)$$

Previous studies adopt a Q-learning algorithm to update value function $V(s)$. However, Since the Q-learning algorithm can only record limited historical trajectories, billions of daily trip orders and corresponding trajectories generate in our country every day. This will cause a memory explosion if the Q-learning algorithm is expected to be applied in a large-scale operational process.

Hence, a Back Propagation-Deep Neural Network (BP-DNN) based estimator, also called the deep Q-learning algorithm, is constructed to fit the value function $V(s)$, and the flowchart is shown in Table 4.

Combining the Q-learning algorithm with the KM algorithm will transform model 2 from a stochastic combinatorial optimization model into another typical integer

TABLE 3: Flowchart of the KM algorithm.

Step 1: Initialize set S
Set SAEVs as a subset V_1 , set delivery, cruising, recharging, and repositioning tasks as a subset V_2
For V_1 , each vertex value is set as the maximum reward r started from the corresponding SAEV
For V_2 , each vertex value is set as 0
Step 2: Find alternating path
Until no vertex can be added into the alternating path
Step 3: Reset vertex value of V_1 and V_2
For $i \in V_1, j \in V_2, d^{ij} = V_1^i + V_2^j - w(e)^{ij}$
Search for all the vertexes both in the subset V_1 and in the alternating path
Search for all the vertexes in the subset V_2 not in the alternating path
Form a series of vertex pair (i, j)
Find the vertex pair whose d^{ij} is minimum
The value of all the vertex in V_1 minus d^{ij} , the value of all the vertex in V_2 plus d^{ij}
Step 4: Repeat Step 2 and Step 3
Until finding the complete matching
Step 5: Output the final matching
Transfer the final matching into the optimal task allocation instructions

linear programming problem. Specifically, the deep Q-learning algorithm will be adopted offline to train historical experience trajectory data to update the value function representing the long-term return of executing each task. The KM algorithm shall be conducted online to realize fast processing and achieving the final optimal matching instructions to guide SAEVs to execute different tasks. This combination saves plenty of online computational resources to ensure convenience and efficiency for practical application.

4. Case Study

To provide a more complete understanding of the proposed optimization models and solving algorithms, a toy example, a macrosimulation of 1 month, and a microsimulation of 6 hours are demonstrated separately. A toy example is designed based on some hypothetical data. Macro- and microsimulation cases share the same dataset. Before conducting task allocation, preliminary generation of potential delivery, parking, recharging, and repositioning tasks waiting for match follows the rules below: (1) delivery tasks to be completed are generated by following the trip order information; (2) it is assumed that each SAEV may receive a parking task; (3) all the charging piles in the hexagon will be listed to form recharging tasks for SAEVs; and (4) “hotspots” regions with more trip demand are predetermined and form the potential repositioning tasks.

4.1. Dataset Introduction. Macro- and microsimulation cases are involved with the real-time geographical location of vehicles, O-D information of real-time trip order request, static geographical location of charging piles, and static hexagonal partition information. In this paper, vehicle and trip order information comes from the open-source dataset

in the Didi platform, which is a one-month (from November 1 to November 31) ride-hailing operational dataset located in Chengdu, Sichuan, China. Charging pile information in Chengdu is achieved by crawling the website <http://admin.bjev520.com/jsp/beiqi/pcmap/do/index.jsp> of BAIC BJEV, which is the holding subsidiary of Beijing Automotive Group Co., Ltd. The FieldInfo is shown in Table 5.

4.2. Toy Example. A toy example is designed to illustrate the capacity of optimization model 1 and the KM algorithm in this multiaction dispatching problem.

Experimental setup: based on the MDP framework, there are 4 agents and 4 tasks (including 1 delivery task, 1 parking task, 1 recharging task, and 1 repositioning task) waiting to be assigned at time slot t . Area D has been predetermined as a hot spot with low SAEV supply and high trip demand at time slot t . Set $\alpha = 1, \beta = 1, \theta = 1$. Assuming that 10 k-Wh can be supplemented through each recharging process, the match weighted value between different agents and tasks is shown below by following the calculation principle of reward in equations (1)–(19) (Table 6).

(3) Result of Toy Example. By executing the KM algorithm in Table 1, the best task allocation instructions based on model 1 can be achieved as follows. Agent 1 executes the parking task and drives straight, agent 2 executes the recharging task, agent 3 executes pickup and delivery task, and agent 4 is repositioned to area D for replenishment. The total reward is 8, which is the maximum return among all the possible task allocation alternative options. The specific assigned tasks at time slot t are illustrated in Figure 3. In addition, to figure out the computational capacity of the solving algorithm, Table 7 displays the elapsed time of the KM algorithm for different fleet sizes of SAEVs (macOS High Sierra 10.13.6, 2.3 GHz, Intel Core i5). To realize fast solving of multiaction allocation instructions second by second, the KM algorithm is a competitive way for SAEVs whose fleet size is less than 1000.

However, the above result may be an optimal allocation for time slot t , but not the best one for the next several time slots. For instance, in Figure 4, if there will be two new delivery task x and task y at time slot $(t+1)$, maybe agent 4 should be assigned to task 2 at time slot t instead of being assigned to the hotspot (task 4) for replenishment. Meanwhile, agent 1 is ought to be relocated to the hotspot to execute repositioning task 4. Since at time slot $t+1$, there will be no delivery task in area A, but more delivery tasks will generate in area C. Agent 4 and agent 1 may satisfy the trip demand task x and task y at time slot $t+1$, and repositioning agent 1 to area D may be a better choice at time t . Hence, optimization model 1 and KM algorithm is a static multiaction dispatching method from the local view, which may be short-sighted. To solve this problem, a dispatching simulator considering a global view will be conducted in the next part.

4.3. Macrosimulation of One Month. The dispatching simulator of 30 days is designed to test the capacity of optimization model 2 and the combinational algorithm of deep

TABLE 4: Flowchart of BP-DNN estimator for fitting value function $V(s)$.

Initialize episode pool and the number of episode samples is N
Initialize weight parameters θ of a BP-DNN network
Repeat episode trajectory of each vehicle, from 1 to K :
 Initialize state $s_1 = \{x_1\}$, and calculate the input sequence $\mathcal{O}_1 = \mathcal{O}(s_1)$
 Repeat time steps of each episode trajectory, from $t = 1$ to T :
 Achieve state $s_{t+1} = \{x_{t+1}\}$ according to state pair (s_t, s_{t+1}, r_t) of the episode trajectory, and calculate the input sequence
 $\mathcal{O}_{t+1} = \mathcal{O}(s_{t+1})$;
 $y_i = \begin{cases} r_j, & \text{current state is the final state } \mathcal{O}_{j+1}, \\ r_j + \gamma \cdot V(\mathcal{O}_{j+1}, \theta), & \text{current state is not the final state } \mathcal{O}_{j+1}, \end{cases}$
 Calculate loss function $(y_i - V(\mathcal{O}_{j+1}, \theta))^2$ based on gradient decent method
 $\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}[(r + \gamma \cdot V(s', \theta_{i-1}) - V(s, \theta_i)) \nabla_{\theta_i} V(s, \theta_i)]$
 Until the final time slot T
Until the final episode trajectory K
Output value function $V(s, \theta)$

TABLE 5: The FieldInfo of dataset.

Field	Type	Example	Notes
Vehicle ID	String	glox.jrrlltBMvCh8nxqktdr2dtopmlH	—
Order ID	String	jkkt8kxniovIFuns9qrrlvst@iqnpkwz	—
Timestamp	Int	1501584540	Unix timestamp, unit: s
Longitude of vehicle	Float	104.04392	GCJ-02 coordinate system
Latitude of vehicle	Float	30.66703	GCJ-02 coordinate system
Remaining battery capacity of vehicle	Float	60	unit: kW·h
Order initiation time	Int	1501582195	Unix timestamp, unit: s
Longitude of drop-on position	Float	104.04392	GCJ-02 coordinate system
Latitude of drop-on position	Float	30.66703	GCJ-02 coordinate system
Longitude of drop-off position	Float	104.04392	GCJ-02 coordinate system
Latitude of drop-off position	Float	30.66703	GCJ-02 coordinate system
Pile ID	Int	1	—
Longitude of charging pile	Float	104.04392	GCJ-02 coordinate system
Latitude of charging pile	Float	30.66703	GCJ-02 coordinate system
Hexagon ID	Int	3	—
Longitude of hexagon center	Float	104.04392	GCJ-02 coordinate system
Latitude of hexagon center	Float	30.66703	GCJ-02 coordinate system

TABLE 6: Weighted value between 4 agents and 4 tasks (instant and single-step reward).

	Task 1 (delivery)	Task 2 (parking)	Task 3 (recharging)	Task 4 (repositioning)
Agent 1	16	-2	-20	-12
Agent 2	24	-2	-14	-6
Agent 3	28	-2	-12	-2
Agent 4	24	-2	-30	-4

Q-learning and KM to improve the final dispatching return from a global view.

(1) *Experimental Setup.* This simulator is conducted based on the trip data in Chengdu, Shanghai, from November 1, 2016, to November 30, 2016, on Didi Chuxing platform. The dataset mainly refers to the order data (for delivery task decision), including order ID, starting billable time, ending billable time, longitude of pickup location, latitude of pickup location, longitude of pickoff location, latitude of pickoff location, and predetermined hexagon areas (for repositioning task decision) including hexagon ID, longitude of six vertexes, and latitude of six vertexes. Also, to achieve recharging task decision, the GPS data of some

charging stations/piles are randomly generated by crawling the website <http://admin.bjev520.com/jsp/beiqi/pcmap/do/index.jsp> of BAIC BJEV. Setting 1 minute as a time slot, this simulator is ready to conduct the task allocation process of 1 month, including 30 days and 43200 minutes. At each time slot, 100 SAEVs and 100 tasks are randomly extracted from the dataset. First, set $\gamma = 0.9$, $\sigma = 0.1$, and train value function of different state $V(s)$ based on the Q-learning algorithm. Second, set $\alpha = 1$, $\beta = 1$, $\theta = 1$, and assume that $10k \cdot \text{Wh}$ can be supplemented through each recharging process, form the match weighted value between different agents and tasks, and implement the simulation by following Figure 5. A final income (without any opportunity cost) of the whole

FIGURE 3: Multi-action allocation instructions of the toy example at time slot t .

TABLE 7: Elapsed time of the KM algorithm for different fleet sizes of SAEVs.

SAEVs	Elapsed time (s)	SAEVs	Elapsed time (s)
10	0.001928	50	0.030737
100	0.089407	200	0.454243
400	2.892085	600	7.218150
800	17.566639	1000	34.524452
2000	351.554187

FIGURE 4: Multi-action allocation instructions at time slot t considering time slot $(t + 1)$.

FIGURE 5: Dispatching simulation process of 1 month.

FIGURE 6: Total reward comparison of two multiaction dispatching optimization models.

month will be counted to represent the economic benefit of this simulation experiment.

(2) *Result of Dispatching Simulation.* Compared with the local view (instant and single-step reward), multiaction allocation simulation from the global view (accumulative and multistep reward) shows a giant improvement of the total reward. As is illustrated in Figure 6, the final reward of the allocation instructions generated from optimization model 2 and the combinational algorithm of deep Q-learning and KM is twice the one generated from optimization model 1 and the KM algorithm. Besides, Figure 7 reveals that the total order revenue of the allocation instructions generated from optimization model 2 and the combinational algorithm of the deep Q-learning and KM improves by 1.2% compared with the one generated from optimization model 1 and KM algorithm. The reason why the improvement in total reward is more significant than the total revenue mainly lies in the long-term return. Since the reward in global view is calculated by adding the reward in local view with a long-term return, and the long-term return is represented by a BP-DNN estimator with a positive value. Another indication we can infer from Figures 6 and 7 is that though the total reward in the global view can improve more significantly than the total reward in the local

view, the actual order revenue can only improve slightly by adopting the global view.

4.4. Microsimulation of Six Hours. To further compare the dispatching performance of different methods, 6 hours (from 8:00 a.m. to 2:00 p.m., 15 minutes as a time step) is chosen as the target time period to measure the operational effect of SAEVs operation. Six contrastive indicators are put forward to conduct a horizontal comparison. These contrastive indicators include (1) accumulated revenue per round of dispatch; (2) scatter distribution of user waiting time of all the orders; (3) histogram distribution of user waiting time of all the orders; (4) order fulfillment rate per round of dispatch; (5) pile utilization rate per round of dispatch.

Some basic configuration, including subregional division results, initialization of SAEV fleet and charging piles, trip order requests distribution of target subregions, is essential. Figure 8 shows 19 hexagonal subregions of the inner ring of Chengdu, Sichuan, China, and these are the target studied areas when conducting the dispatching process. Figure 9 reveals the static configuration information, which contains the number of charging piles in each subregion and the number of SAEVs in each subregion at 8 a.m.

FIGURE 7: Total revenue comparison of two multiaction dispatching optimization models.

FIGURE 8: Regional division of inner ring of Chengdu, Sichuan, China.

FIGURE 9: Fleet and charging piles initialization configuration among 19 subregions.

Since recharging and repositioning task assignment will be conducted separately in each subregion and repositioning task assignment should be executed among several subregions, subregion 1 is selected as the target area to illustrate the dispatching performance. 1–7 subregions are selected as the target repositioning areas in this research. Figure 10 exhibits the trip order requests distribution of 7 subregions among 24 time steps (from 8:00 a.m.–2:00 p.m., 15 minutes as a time step).

4.4.1. The Role of Repositioning Task Assignment: Improve Order Fulfillment. The dispatching methodology put forward in this paper, comprehensively considering delivery, recharging, and repositioning tasks, can increase the order fulfillment at each time step to a great extent. As is illustrated in Figure 11, adding repositioning task assignment can keep 100% order fulfillment for 18 time steps during 8 hours (24 time steps), and the lowest order fulfillment is 50%, which is

FIGURE 10: Order requests initialization configuration of 1–7 subregions from 8:00 a.m.–2:00 p.m.

FIGURE 11: Order fulfillment of assignment process considering different tasks.

FIGURE 12: Total revenue comparison of global view and local view.

higher than the situation only considering delivery and recharging task assignment. The above results fully illustrate the importance of introducing vehicle redistribution task assignment in the dispatching process of SAEVs.

4.4.2. “Model 2 + BP-DNN + KM” versus “Model 1 + KM”. As is shown in Figure 12, the total order revenue of the allocation instructions generated from optimization

model 2 and the combinational algorithm of deep Q-learning (BP-DNN based value function) and KM algorithm improves by 2.78% compared with the one generated from optimization model 1 and the KM algorithm. This further reveals the better performance of the combination of BP-DNN based deep Q-learning algorithm and KM algorithm, which means the dispatch from global view can bring an economic income growth for SAEV fleet operators.

FIGURE 13: The loss value variation of 4080 episodes based on BP-DNN.

FIGURE 14: User waiting time comparison of global view and local view.

According to the loss function $(y_i - V(\phi_j, \theta))^2$ training process based on the Gradient Descent method in Table 5, 4080 completed historical order records are collected as a training set. Variation of loss value is shown in Figure 13, with a steady loss of 4651.600955. There is still a large room for optimization of the loss function in the future.

Results of model 2 in a global view considering instant return and long-term return show a decrease in the aspect of average user waiting time. As shown in Figure 14, the skewness of the frequency distribution of all the completed trip orders becomes larger in the global view case than the one in the local view case. Specifically, the number of

FIGURE 15: Pile utilization rate comparison of global view and local view.

completed trip orders with 0–200 s interval increases from 1600 to 2400, showing a shorter user pickup duration and better user trip satisfaction. However, around 3 orders show longer user waiting time with more than 4000 s, which reveals that the global view case may cause some longer-pickup-duration orders, but the average user waiting time still can be decreased.

Results show that different reward functions in both local view and global view cannot affect the utilization rate of charging piles. As illustrated in Figure 15, the pile utilization rate during 6 hours (24 time steps) of the local view case and global view case remains almost the same. Meanwhile, Figure 15 also reveals a common feature of two cases, that is, almost 10 time steps keep 100% pile rate utilization during the 6-hour dispatching process. This may be caused by the low supply of charging piles of the studied area, which also has implications for our future research on the location and layout of charging piles serving SAEVs.

5. Conclusions and Future Work

In this paper, the operational dispatching process of the L4/L5 level shared autonomous electric fleet, including delivery, parking, recharging, and repositioning, is put forward and modeled as a multiagent multiaction dynamic dispatching problem based on MDP. To achieve the optimal task allocation instructions for each SAEV, two multiaction dispatching optimization models from a local view (instant and single-step reward) and global view (accumulative and multistep reward) based on combinatorial optimization method are established, respectively. Correspondingly, two algorithms involved with the KM algorithm and Back Propagation-Deep Neural Network algorithm are designed to realize a rapid and exact solution. Based on the actual order and trajectory data from the Didi Chuxing platform, a toy example, a macrosimulation of 1 month, and a microsimulation of 6 hours are conducted to test the validity and effectiveness of the methods put forward in this paper. Results of the case study reveal the validity of the methods in this research. MDP is an effective method for modeling the future operating process of SAEVs as a multiagent task

allocation problem and combinatorial optimization method is feasible for solving the best multiaction allocation instructions.

First, results prove that the KM algorithm can realize fast solving of optimal assignment scheme solving for 1000 SAEVs in the practical application scenario. Second, establishing the optimization model from a global view considering accumulative and multistep reward shall bring an obvious improvement of the total multiaction allocation return compared with the local view. Third, the Q-learning algorithm and KM algorithm is a perfect combination of the offline and online methods, which can be packaged to be applied to the future SAEVs operation application. Fourth, the deep Q-learning algorithm based on Back Propagation-Deep Neural Network (BP-DNN) shows a better performance than the Q-learning algorithm, but for the fit goodness of the BP-DNN estimator, there remains room for improvement. Fifth, adopting model 2 from the global view can bring not only 2.78% increase in the total order revenue but also a decrease in the average user waiting time (i.e., an increase of user trip satisfaction).

For future works, we are committed to improving this research from 3 aspects. First, to explore a better operational performance, the reward function in the MDP framework should be further modified by adding the influence of user satisfaction and carbon emission. Second, a supply-demand forecasting model should be established to predetermine the hotspots with low vehicle supply and high trip demand. Hence, a more accurate trip demand forecast model should be studied in the future. Third, the deep Q-learning algorithm is verified to be effective in both increasing total order revenue and decreasing user waiting time. However, the loss value of the BP-DNN estimator remains higher. Further exploration will focus on the search for better deep Q-learning algorithms for improving the fit of goodness [55–60].

Appendix

Mathematical model 1: set instant, single-step reward as a weighted value:

$$\text{Max } W = \sum_{i=1}^I \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^P (r_{ij} \cdot x_{ij} + r_{im} \cdot x_{im} + r_{in} \cdot x_{in} + r_{ip} \cdot x_{ip}). \quad (\text{A.1})$$

Constraint 1: $x_{ij}, x_{im}, x_{in}, x_{ip} \in \{0, 1\}$,

Constraint 2: $r_{ij}, r_{im}, r_{in}, r_{ip} \in [-\infty, +\infty]$,

Constraint 3: $\forall i \in I_s, \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^P (x_{ij} + x_{im} + x_{in} + x_{ip}) = 1$,

Constraint 4: $\forall j \in J_s, \sum_{i=1}^I x_{ij} = 1$,

Constraint 5: $\forall m \in M_s, \sum_{i=1}^I x_{im} = 1$,

Constraint 6: $\forall n \in N_s, \sum_{i=1}^I x_{in} = 1$,

Constraint 7: $\forall p \in P_s, \sum_{i=1}^I x_{ip} = 1$.

Data Availability

In this paper, vehicle and trip order information comes from the open-source dataset in the Didi platform, which is a one-month (from November 1 to November 31) ride-hailing operational dataset located in Chengdu, Sichuan, China. Charging pile information in Chengdu is achieved by crawling the website <http://admin.bjev520.com/jsp/beiqi/pormap/do/index.jsp> of BAIC BJEV, which is the holding subsidiary of Beijing Automotive Group Co., Ltd.

Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents any conflict of interest in connection with the work submitted.

Authors' Contributions

Ning Wang conceived and designed the study; Jiahui Guo performed data collection; Jiahui Guo performed analysis and interpretation of results; and Jiahui Guo performed draft manuscript preparation. All the authors reviewed the results and approved the final version of the manuscript.

Acknowledgments

This research was supported by the project of the Dynamic Relocation Model of Electric Car Sharing funded by Special Funds for Basic Scientific Research Business Expenses of Central Colleges and Universities (kx0170020172681). In addition, part of the order and trajectory data used to train value function in the dispatching simulator of the case study were obtained from Didi Chuxing, which can be applied on the website <https://gaia.didichuxing.com>.

References

- [1] I. Riccardo, M. Benjamin, and T. Tetsuo, "Costs and carbon emissions of shared autonomous electric vehicles in a Virtual Power Plant and Microgrid with renewable energy," *Energy Procedia*, vol. 156, pp. 401–405, 2019.
- [2] H. Dia and F. Javanshour, "Autonomous shared mobility-on-demand: melbourne pilot simulation study," *Transportation Research Procedia*, vol. 22, pp. 285–296, 2017.
- [3] H. Miao, H. Jia, J. Li, and T. Z. Qiu, "Autonomous connected electric vehicle (ACEV)-based car-sharing system modeling and optimal planning: a unified two-stage multi-objective optimization methodology," *Energy*, vol. 169, pp. 797–818, 2019.
- [4] F. Ferrero, G. Perboli, M. Rosano, and A. Vesco, "Car-sharing services: an annotated review," *Sustainable Cities and Society*, vol. 37, pp. 501–518, 2018.
- [5] Z. Liu, T. Miwa, W. Zeng, M. G. H. Bell, and T. Morikawa, "Dynamic shared autonomous taxi system considering on-time arrival reliability," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 281–297, 2019.
- [6] R. Vosooghi, J. Puchinger, M. Jankovic, and A. Vouillon, "Shared autonomous vehicle simulation and service design," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 15–33, 2019.
- [7] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 1–13, 2014.
- [8] F. Steck, V. Kolarova, F. Bahamonde-Birke, S. Trommer, and B. Lenz, "How autonomous driving may affect the value of travel time savings for commuting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 46, pp. 11–20, 2018.
- [9] E. Biondi, C. Boldrini, and R. Bruno, "Optimal deployment of stations for a car sharing system with stochastic demands: a queueing theoretical perspective," in *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, November 2016.
- [10] L. Fulton, J. Mason, and D. Meroux, *Three Revolutions in Urban Transportation: How to Achieve the Full Potential of Vehicle Electrification, Automation, and Shared Mobility in Urban Transportation Systems Around the World by 2050 (No. STEPS-2050)*, UC Davis, Davis, CA, USA, 2017.
- [11] R. Mounce and J. D. Nelson, "On the potential for one-way electric vehicle car-sharing in future mobility systems," *Transportation Research Part A: Policy and Practice*, vol. 120, pp. 17–30, 2019.
- [12] A. Aitouahmed, D. Josselin, and F. Zhou, "Relocation optimization of electric cars in one-way car-sharing systems: modeling, exact solving and heuristics algorithms," *International Journal of Geographical Information Science*, vol. 32, no. 1, pp. 1–32, 2017.
- [13] B. Boyacı, K. G. Zografos, and N. Geroliminis, "An optimization framework for the development of efficient one-way car-sharing systems," *European Journal of Operational Research*, vol. 240, no. 3, pp. 718–733, 2015.
- [14] S. Illgen and M. Höck, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transportation Research Part B: Methodological*, vol. 120, pp. 193–204, 2019.
- [15] R. Lu, G. Correia, X. Zhao, X. Liang, and Y. Lv, "Performance of one-way carsharing systems under combined strategy of pricing and relocations," *Transportmetrica B: Transport Dynamics*, vol. 9, no. 4, pp. 1–19, 2020.
- [16] S. A. Shaheen and A. P. Cohen, "Carsharing and personal vehicle services: worldwide market developments and emerging trends," *International Journal of Sustainable Transportation*, vol. 7, no. 1, pp. 5–34, 2013.
- [17] N. Wang, J. Guo, X. Liu, and Y. Liang, "Electric vehicle car-sharing optimization relocation model combining user relocation and staff relocation," *Transportation Letters*, vol. 13, no. 4, pp. 315–326, 2020.
- [18] M. Nourinejad and M. J. Roorda, "A dynamic carsharing decision support system," *Transportation Research Part E*:

- Logistics and Transportation Review*, vol. 66, pp. 36–50, 2014.
- [19] M. Nourinejad, S. Zhu, S. Bahrami, and M. J. Roorda, "Vehicle relocation and staff rebalancing in one-way carsharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 81, pp. 98–113, 2015.
 - [20] M. Xu, Q. Meng, and Z. Liu, "Electric vehicle fleet size and trip pricing for one-way carsharing services considering vehicle relocation and personnel assignment," *Transportation Research Part B: Methodological*, vol. 111, pp. 60–82, 2018.
 - [21] G. H. D. A. Correia and A. P. Antunes, "Optimization approach to depot location and trip selection in one-way car-sharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 233–247, 2012.
 - [22] Y. Hua, D. Zhao, X. Wang, and X. Li, "Joint infrastructure planning and fleet management for one-way electric car sharing under time-varying uncertain demand," *Transportation Research Part B: Methodological*, vol. 128, pp. 185–206, 2019.
 - [23] L. Al-Kanj, J. Nascimento, and W. B. Powell, "Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles," *European Journal of Operational Research*, vol. 284, no. 3, pp. 1088–1106, 2020.
 - [24] S. Narayanan, E. Chanotakis, and C. Antoniou, "Shared autonomous vehicle services: a comprehensive review," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 255–293, 2020.
 - [25] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: a queueing-theoretical perspective," *The International Journal of Robotics Research*, vol. 35, no. 1–3, pp. 186–203, 2014.
 - [26] T. D. Chen and K. M. Kockelman, "Management of a shared autonomous electric vehicle fleet: implications of pricing schemes," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2572, no. 1, pp. 37–46, 2016.
 - [27] C. L. Azevedo, K. Marczuk, S. Raveau et al., "Microsimulation of demand and supply of autonomous mobility on demand," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2564, no. 1, pp. 21–30, 2016.
 - [28] W. Zhang and S. Guhathakurta, "Parking spaces in the age of shared autonomous vehicles: how much parking will we need and where?" *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2651, no. 1, pp. 80–91, 2017.
 - [29] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.
 - [30] G. S. Bauer, J. B. Greenblatt, and B. F. Gerke, "Cost, energy, and environmental impact of automated electric taxi fleets in manhattan," *Environmental Science & Technology*, vol. 52, no. 8, pp. 4920–4928, 2018.
 - [31] T. Babicheva, W. Burghout, I. Andreasson, and N. Faul, "The matching problem of empty vehicle redistribution in autonomous taxi systems," *Procedia Computer Science*, vol. 130, pp. 119–125, 2018.
 - [32] J. Farhan and T. D. Chen, "Impact of ridesharing on operational efficiency of shared autonomous electric vehicle fleet," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 310–321, 2018.
 - [33] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone, "Routing autonomous vehicles in congested transportation networks: structural properties and coordination algorithms," *Autonomous Robots*, vol. 42, no. 7, pp. 1427–1442, 2018.
 - [34] R. Iacobucci, B. McLellan, and T. Tezuka, "Modeling shared autonomous electric vehicles: potential for transport and power grid integration," *Energy*, vol. 158, pp. 148–163, 2018.
 - [35] F. Dandl, M. Hyland, K. Bogenberger, and H. S. Mahmassani, "Evaluating the impact of spatio-temporal demand forecast aggregation on the operational performance of shared autonomous mobility fleets," *Transportation*, vol. 46, no. 6, pp. 1–22, 2019.
 - [36] R. Iacobucci, B. McLellan, and T. Tezuka, "Optimization of shared autonomous electric vehicles operations with charge scheduling and vehicle-to-grid," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 34–52, 2019.
 - [37] E. C. Jones and B. D. Leibowicz, "Contributions of shared autonomous vehicles to climate change mitigation," *Transportation Research Part D: Transport and Environment*, vol. 72, pp. 279–298, 2019.
 - [38] C. Mao, Y. L. Liu, and Z. J. Shen, "Dispatch of autonomous vehicles for taxi services: a deep reinforcement learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 115, 2020.
 - [39] X. Liang, G. H. D. A. Correia, K. An, and B. van Arem, "Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 260–281, 2020.
 - [40] H. Zhang, C. J. R. Sheppard, T. E. Lipman, T. Zeng, and S. J. Moura, "Charging infrastructure demands of shared-use autonomous electric vehicles in urban areas," *Transportation Research Part D: Transport and Environment*, vol. 78, Article ID 102210, 2020.
 - [41] J. Shi, Y. Gao, W. Wang, N. P. Yu, and P. A. Ioannou, "Operating electric vehicle fleet for ride-hailing services with reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–13, 2019.
 - [42] K. A. Melendez, T. K. Das, and C. Kwon, "Optimal operation of a system of charging hubs and a fleet of shared autonomous electric vehicles," *Applied Energy*, vol. 279, Article ID 115861, 2020.
 - [43] T. Z. Zhang and T. D. Chen, "Smart charging management for shared autonomous electric vehicle fleets: a Puget Sound case study," *Transportation Research Part D: Transport and Environment*, vol. 78, Article ID 102184, 2020.
 - [44] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin," *Texas. Transportation*, vol. 45, no. 1, pp. 1–16, 2016.
 - [45] K. M. Gurumurthy and K. M. Kockelman, "Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from Orlando, Florida," *Computers, Environment and Urban Systems*, vol. 71, pp. 177–185, 2018.
 - [46] S. Hörl, "Agent-based simulation of autonomous taxi services with dynamic demand responses," *Procedia Computer Science*, vol. 109, pp. 899–904, 2017.
 - [47] B. Jäger, F. M. M. Agua, and M. Lienkamp, "Agent-based simulation of a shared, autonomous and electric on-demand mobility solution," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, October 2017.
 - [48] L. M. Mendes, M. R. Bennassar, and J. Y. J. Chow, "Comparison of light rail streetcar against shared autonomous vehicle fleet for brooklyn-queens connector in New York

- city,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2650, no. 1, pp. 142–151, 2017.
- [49] K. Braekers, A. Caris, and G. K. Janssens, “Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots,” *Transportation Research Part B: Methodological*, vol. 67, no. 9, pp. 166–186, 2014.
 - [50] L. M. Martinez and J. M. Viegas, “Assessing the impacts of deploying a shared self-driving urban mobility system: an agent-based model applied to the city of Lisbon, Portugal,” *International Journal of Transportation Science and Technology*, vol. 6, no. 1, pp. 13–27, 2017.
 - [51] J. Paquette, J.-F. Cordeau, G. Laporte, and M. M. B. Pascoal, “Combining multicriteria analysis and tabu search for dial-a-ride problems,” *Transportation Research Part B: Methodological*, vol. 52, pp. 1–16, 2013.
 - [52] R. Iglesias, F. Rossi, R. Zhang, and M. Pavone, “A BCMP network approach to modeling and controlling autonomous mobility-on-demand systems,” *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 357–374, 2018.
 - [53] R. Iacobucci, B. Mclellan, and T. Tezuka, “Costs and carbon emissions of shared autonomous electric vehicles in a Virtual Power Plant and Microgrid with renewable energy,” *Energy Procedia*, vol. 156, pp. 401–405, 2019.
 - [54] R. Basso, B. Kulcsár, and I. Sanchez-Diaz, “Electric vehicle routing problem with machine learning for energy prediction,” *Transportation Research Part B: Methodological*, vol. 145, no. 6, pp. 24–55, 2021.
 - [55] Z. Qin, X. Tang, Y. Jiao et al., “Ride-hailing order dispatching at DiDi via reinforcement learning,” *INFORMS Journal on Applied Analytics*, vol. 50, no. 5, pp. 272–286, 2020.
 - [56] Z. Shou and X. Di, “Reward design for driver repositioning using multi-agent reinforcement learning,” *Transportation Research Part C: Emerging Technologies*, vol. 119, Article ID 102738, 2020.
 - [57] X. C. Tang, Z. W. Qin, F. Zhang et al., “A deep value-network based approach for multi-driver order dispatching,” in *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD’19)*, Anchorage, AK, USA, August 2019.
 - [58] N. Wang, J. Guo, X. Liu, and T. Fang, “A service demand forecasting model for one-way electric car-sharing systems combining long short-term memory networks with Granger causality test,” *Journal of Cleaner Production*, vol. 244, pp. 118812–118816, 2020.
 - [59] Z. Xu, Z. Li, Q. Guan et al., “Large-scale order dispatch in on-demand ride-hailing platforms: a learning and planning approach,” in *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD’18)*, London, UK, July 2018.
 - [60] Z. Zhu, J. Ke, and H. Wang, “A mean-field Markov decision process model for spatial-temporal subsidies in ride-sourcing markets,” *Transportation Research Part B: Methodological*, vol. 150, pp. 540–565, 2021.