Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Two-Sided Deep Reinforcement Learning for Dynamic Mobility-on-Demand Management with Mixed Autonomy

Jiaohong Xie, Yang Liu, Nan Chen

To cite this article:
Jiaohong Xie, Yang Liu, Nan Chen (2023) Two-Sided Deep Reinforcement Learning for Dynamic Mobility-on-Demand Management with Mixed Autonomy. Transportation Science 57(4):1019-1046. https://doi.org/10.1287/trsc.2022.1188

https://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Vol. 57, No. 4, July–August 2023, pp. 1019–1046
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Two-Sided Deep Reinforcement Learning for Dynamic Mobility-on-Demand Management with Mixed Autonomy

Jiaohong Xie,[a] Yang Liu,[a,b,*] Nan Chen[a]

[a] Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576; [b] Department of Civil and Environmental Engineering, National University of Singapore, Singapore 117576

*Corresponding author

Contact: xiejiaohong@u.nus.edu, https://orcid.org/0000-0001-9008-3660 (JX); iseliuy@nus.edu.sg, https://orcid.org/0000-0002-0862-6046 (YL); isecn@nus.edu.sg, https://orcid.org/0000-0003-2495-5234 (NC)

Received: September 2, 2021
Revised: March 6, 2022; June 25, 2022; October 13, 2022
Accepted: October 21, 2022
Published Online in Articles in Advance: January 17, 2023

https://doi.org/10.1287/trsc.2022.1188

Abstract. Autonomous vehicles (AVs) are expected to operate on mobility-on-demand (MoD) platforms because AV technology enables flexible self-relocation and system-optimal coordination. Unlike the existing studies, which focus on MoD with pure AV fleet or conventional vehicles (CVs) fleet, we aim to optimize the real-time fleet management of an MoD system with a mixed autonomy of CVs and AVs. We consider a realistic case that heterogeneous boundedly rational drivers may determine and learn their relocation strategies to improve their own compensation. In contrast, AVs are fully compliant with the platform's operational decisions. To achieve a high level of service provided by a mixed fleet, we propose that the platform prioritizes human drivers in the matching decisions when on-demand requests arrive and dynamically determines the AV relocation tasks and the optimal commission fee to influence drivers' behavior. However, it is challenging to make efficient real-time fleet management decisions when spatiotemporal uncertainty in demand and complex interactions among human drivers and operators are anticipated and considered in the operator's decision making. To tackle the challenges, we develop a two-sided multiagent deep reinforcement learning (DRL) approach in which the operator acts as a supervisor agent on one side and makes centralized decisions on the mixed fleet, and each CV driver acts as an individual agent on the other side and learns to make decentralized decisions noncooperatively. We establish a two-sided multiagent advantage actor-critic algorithm to simultaneously train different agents on the two sides. For the first time, a scalable algorithm is developed here for mixed fleet management. Furthermore, we formulate a two-head policy network to enable the supervisor agent to efficiently make multitask decisions based on one policy network, which greatly reduces the computational time. The two-sided multiagent DRL approach is demonstrated using a case study in New York City using real taxi trip data. Results show that our algorithm can make high-quality decisions quickly and outperform benchmark policies. The efficiency of the two-head policy network is demonstrated by comparing it with the case using two separate policy networks. Our fleet management strategy makes both the platform and the drivers better off, especially in scenarios with high demand volume.

History: This paper has been accepted for the *Transportation Science* Special Issue on Emerging Topics in Transportation Science and Logistics.

Funding: This work was supported by the Singapore Ministry of Education Academic Research [Grant MOE2019-T2-2-165] and the Singapore Ministry of Education [Grant R-266-000-135-114].

Keywords: mobility-on-demand systems • multi-agent deep reinforcement learning • deep neural networks • online dynamic fleet management • mixed fleet • autonomous vehicle

1. Introduction

Mobility-on-demand (MoD) platforms have substantially transformed our lives. With the emergence of MoD services, it is expected that limited transportation resources can be better allocated so as to reduce traffic congestion, car ownership, and the need for parking space. The past decade has witnessed significant growth in the MoD market and the rapid development

of self-driving technologies. Compared with traditional human-driven fleets, automated fleets provide many advantages for advanced fleet management strategies (Litman 2017). Autonomous vehicle (AV) fleets are compliant with operation centers so that sophisticated system-level management strategies can be realized. For instance, AVs owned by the MoD platform can be relocated to certain areas when a shortage of vehicles occurs. Moreover, AVs are not limited by manpower constraints, so a more stable and reliable service level can be achieved in cities.

Given the benefit of AVs, MoD platforms, such as Uber, have announced they will introduce autonomous vehicles into their platform in the future (Shetty 2020). However, it is costly for platforms to maintain and manage AV fleets, and major technical barriers still exist before the platforms could transit into fully autonomous fleets (Wei et al. 2019). Moreover, the transition of MoD systems from conventional vehicle (CV) fleets to fully automated fleets will potentially eliminate millions of driving jobs (Pakusch et al. 2020). For these reasons, most MoD platforms will adopt a hybrid model, in which AVs provide mobility service along with human-driven CVs. In the foreseeable future, the market penetration of AVs is expected to gradually increase, and mobility services will be provided by mixed-fleet MoD systems (Noruzoliaee, Zou, and Liu 2018). Thus, this paper focuses on the challenges of managing the mixed fleet during the inevitable long transitional period. In the current MoD systems with a pure CV fleet, many challenges persist and require thorough investigation. For a large-scale MoD system with a large number of trip requests from a metropolitan area, one of the major challenges is to address the imbalance between demand (i.e., the trip requests) and supply (i.e., the vehicles), which is caused by the asymmetric demand distribution across time and space (Chen et al. 2020a). Currently, two types of solutions are principally studied to tackle the challenge. The first type of solution makes use of pricing schemes to influence the demand side (Wang et al. 2016, Ma et al. 2020, Wang et al. 2020a). For instance, surge pricing (Zha, Yin, and Du 2018) has been studied and implemented to influence both the demand and supply. However, it is observed that surge pricing may have a notable negative effect on trip demand but a weak positive effect on vehicle supply, so surge pricing may not improve the platform profit significantly (Chen, Mislove, and Wilson 2015). The second type of solution focuses on the supply side and relies on vehicle relocation to rebalance vehicles (Furuhata et al. 2013, Nourinejad and Roorda 2016, Chen and Liu 2022, Liu, Xie, and Chen 2022). The MoD platform relocates vehicles to locations with predicted higher demand in the next stage to better fulfill the demand. However, human drivers' behavior and responses should not be neglected when relocation decisions are made. Drivers, who play a noncooperative game to maximize individual compensation, may not be willing to follow the relocation order if they cannot benefit from relocation. Shou and Di (2020) propose to design a reward scheme to encourage MoD drivers to self-relocate without harming the demand in long-term equilibrium, whereas, in the literature, there is a lack of study on real-time instruments to dynamically influence supply and reduce the imbalance between demand and supply.

When a mixed fleet operates in the near future, MoD platform operators will face additional challenges. A mixed fleet consists of two types of vehicles with distinct characteristics and behavior. The operator matches trip requests with vehicles (i.e., matching decisions), dispatches AVs to pick-up locations (i.e., dispatching decisions), and relocates AVs to areas in shortage of vehicles (i.e., relocation decisions). AVs are fully compliant with the centralized platform's matching, dispatching, and relocation decisions, whereas human drivers take matching decisions from the platform but freely make relocation decisions in a decentralized manner to maximize their individual compensation. Different from decision making for a fully autonomous fleet or a pure CV fleet, when an operator makes decisions for a mixed fleet to maximize the system efficiency, the responses of AVs and CVs to the operator's decisions must be considered simultaneously. Therefore, it is appealing to capture the complex interactions among human drivers, AVs, and the operator so as to make efficient fleet management decisions. Also, to ensure a smooth transition that can retain a good number of drivers, the platform should take advantage of AV technology without harming drivers' interests. If drivers do not receive reasonable compensation, they might choose to leave the job.

Also, although massive data are collected by MoD platforms to better understand the spatiotemporal uncertainty in trip requests, the challenge still arises when stochastic demand arrives, and efficient real-time decisions must be made quickly. Online operation problems in large-scale MoD networks are often formulated as a Markov Decision Process (MDP). For the management of shared mobility systems in large cities, directly solving the MDP is challenging because the state space and action space are tremendously large, giving rise to the "curse of dimensionality" (Powell 2009). The reinforcement learning (RL) technique is widely adopted in literature to overcome this limitation (Chow, Yu, and Pavone 2015; Lin et al. 2018b; Sutton and Barto 2018; Xu et al. 2018; Xie et al. 2022). In RL, optimal policies are learned through continuous interactions between the agents and the environment, and an exact model of the environment is not required. When combined with modern techniques from deep learning, it becomes the so-called deep reinforcement learning (DRL) algorithm (Li 2017; Qin, Tang, and Ye 2019; Haydari and Yilmaz 2020). However, existing DRL frameworks cannot solve our fleet management problem in mixed-autonomous MoD systems, where two types of agents with distinct behavior and objectives coexist. It is also unclear how an

operator can simultaneously coordinate the AV fleet and influence and anticipate the behavior of drivers, who play a noncooperative game.

The research gaps are summarized here. First, the majority of existing studies on MoD services mainly assume exclusive CVs or AVs. To the best of our knowledge, dynamic mixed-fleet management has not been investigated. Wei et al. (2019) present analytical results incorporating a ride-hailing system with both CV fleet and AV fleet. Still, no study investigates the real-time operation problem for a mixed fleet. Second, for addressing the imbalance between demand and supply, most studies focus on the demand side. The potential instruments, which may dynamically influence the supply side, are not thoroughly studied. Last, although many recent studies have successfully applied RL techniques to real-time MoD operation problems, no research investigates the MoD platforms with both AVs and CVs, where the consideration of the complex interactions among human drivers, AVs, and the operator is critical and challenging.

To fill the research gaps and address the aforementioned challenges, we aim to optimize the dynamic fleet management in a large-scale MoD system with mixed autonomy. We consider a realistic case that human drivers take matched orders from the platform but may relocate freely and learn strategies to maximize individual compensation. In contrast, AVs are fully compliant with the platform's dispatching and relocation decisions. Furthermore, the spatiotemporal uncertainty in future trip requests must be explicitly accounted into the decision-making process. The contributions of the paper are summarized as follows.

First, unlike the studies that focus on MoD with a pure AV fleet or CV fleet, our study fills the research gap of the dynamic real-time operational problem of MoD systems with a mixed fleet. We explicitly consider the stochastic nature of travel demand and the distinct behaviors of the two types of fleets. AVs are fully compliant with the centralized platform's decisions to maximize system efficiency, whereas heterogeneous boundedly rational human drivers play a noncooperative game and make decentralized decisions to maximize their individual compensation. To utilize AV technology to complement CV services, we propose prioritizing human drivers in the matching decisions when random on-demand requests arrive in real time. Meanwhile, we consider that the platform dynamically determines the optimal commission fee (CF) and the operational decision to influence human driver behavior and optimize system efficiency. By doing so, we aim to ensure drivers receive sufficient orders while the platform offers a high level of service by taking advantage of AVs. This provides a sustainable solution for the MoD systems with a mixed fleet coexisting in the long transitional period and facilitates a smooth transition toward a fully automated system. Different from the surge pricing method, which mainly affects the demand, dynamic CF encourages drivers to relocate to areas with a predicted shortage of supply, which may potentially benefit both drivers and passengers.

Second, although there are existing studies investigating the MoD operational problem adopting multiagent reinforcement learning (MARL), the existing approaches are not applicable to solving the two-sided problem because of the distinct characteristics of the mixed fleet. We propose a novel two-sided multiagent DRL approach in which the platform operator is considered a supervisor agent on one side and the human drivers are regarded as driver agents on the other side, interacting in a common environment. The operator aims to determine the optimal decisions of AV relocation and dynamic CF in a centralized manner, whereas heterogeneous drivers with varying bounded rationality choose their relocation strategies in a decentralized manner. However, it is challenging to make efficient real-time fleet management decisions because the supervisor agent must anticipate the uncertain demand, drivers' behavior, and vehicle distribution in the future and make real-time decisions proactively. To tackle the challenges, we formulate a two-sided multiagent RL algorithm using a mean-field approximation to train the two types of agents simultaneously and enhance the approximation quality of RL by deep neural networks (DNNs). Furthermore, we formulate a two-head policy network to efficiently make multitask decisions, which enhances the performance and reduces the computational time compared with that using two separate policy networks. We also let each grid be an agent to give grid-specific AV relocation and CF decisions, which significantly reduces the action space. Our model-free approach incorporates the complexity in the interactions between driver agents and the supervisor agent without relying on a predefined model that needs to be accurate enough and provides a computationally tractable and high-quality dynamic fleet management algorithm. Our two-sided DRL approach is one of the first attempts to examine such a two-sided problem. It is a tractable and stable approach to solving the fleet management problem of MoD systems with a mixed fleet. The proposed framework may be further tuned and potentially adapted to analyze real-time operational problems in transport platforms with a dual-sourcing market (e.g., freelance drivers and contractual drivers) or other commercial systems with emerging urban mobility services (e.g., crowd-sourcing goods delivery).

Third, we conduct an extensive real-world case study in New York City (NYC) using real trip data and validate the feasibility and applicability of our two-sided multiagent DRL approach. The numerical experiments show that our algorithm can make real-time decisions rapidly and outperform benchmark policies in various performance metrics. The effectiveness of the two-head policy network is validated by comparing the

algorithm with two separate policy networks. We examine the benefits from the perspectives of various stakeholders. Our proposed framework generates a win-win situation for both the platform and boundedly rational drivers when AVs are operated optimally to complement CVs in the system, and the supervisor agent learns and influences drivers' behavior via a dynamic CF scheme. Specifically, we demonstrate that both the platform's profit and the drivers' compensations are increased compared with the scenarios without CFs. Meanwhile, managerial insights are provided by examining various AV penetration rates and demand levels. The system-level performance is generally improved by increasing AV penetration without harming drivers' benefits. Our fleet management strategies make both the platform and the drivers better off, especially in scenarios with high demand volume.

The remainder of this paper is organized as follows: First, we review related literature reviewed in Section 2. Section 3 introduces the problem and formally formulates the dynamic system as a multiagent MDP. Section 4 elaborates the details of the two-sided multiagent DRL algorithm. Section 5 conducts numerical experiments in New York City with a realistic taxi trip data set. Managerial insights are elicited according to the experiment results. Section 6 concludes the paper along with possible future directions.

2. Literature Review

Many studies examine the dynamic operation optimization of MoD systems to maintain system efficiency and enhance profit. This section first introduces classic methods for modeling and operating shared mobility systems. Then, it is followed by an introduction of the RL method and its applications in transportation systems, especially in MoD systems. Finally, research gaps in MoD systems with mixed fleets are identified to ensure the contributions of this paper.

2.1. Conventional Operation Methods

For conventional vehicle drivers' movements, operators tend to use dynamic pricing schemes, which potentially affect the short-term demand and supply, redistributing the drivers to tackle the market unbalance. Temporal and spatial surge pricing (Zha, Yin, and Du 2018; Zha, Yin, and Xu 2018; Ma et al. 2020; Yang et al. 2020a; Hu et al. 2021) is a commonly used strategy that affects both passengers' willingness to select the on-demand ride service and supply volume. In practice, Uber uses a surge multiplier that increases prices during peak hours, reducing demand and increasing profits for drivers, and eventually motivating more drivers to enter the market during peak hours. There are other choices for the MoD platforms, such as dynamic pricing, commission fee scheme, and incentive policy to influence the supply over a day (He et al. 2018, Chen et al. 2020a, Hu and Zhou 2020). Zha, Yin, and Du (2018) model the supply of MoD drivers and investigate the effect of surge pricing. The platform and drivers both achieve higher revenue. However, it may harm passengers' interest during periods with high surges compared with static pricing. It has been analytically proved that the MoD platform, which dynamically sets both prices and wages (dynamic CF), could enjoy a significantly higher profit than the platform leveraging the fixed commission (Cachon, Daniels, and Lobel 2017). Several studies investigate the dynamic CF scheme in MoD systems. For example, Chen et al. (2020a) jointly determines time-variant surge pricing, commission rate, and incentives in dynamic MoD systems. Such a strategy helps the MoD platform balance the supply and demand resources and achieve better optimization results. Shou and Di (2020) propose a piece-wise linear function to describe the service charge and optimize the service charge rate. The service charge rate varies according to the demand-supply ratio at different locations and times. Luo and Saigal (2017) also dynamically optimize the prices and commission fees for each location and time for maximizing the revenue of a ridesharing platform. From the perspective of regulation, Zha, Yin, and Du (2017) and Zha, Yin, and Yang (2016) investigate the commission rate regulation. It limits the amount of commission the platform can achieve during each time interval. They find that the commission cap regulation potentially can enhance market efficiency and prevent monopoly.

On the other hand, mobile-app-based platforms support real-time data collection and pricing adjustment, which needs suitable models that capture the stochasticity and the dynamics of drivers and passengers in the system. The queueing-network model is used to describe the vehicle-sharing system. Sayarshad and Chow (2017) build a dynamic location-allocation-queueing model to make decisions in a rolling horizon manner, whereas Banerjee, Johari, and Riquelme (2015) use a queueing-theoretical model to generate optimal platform pricing. Another stream of research models the MoD system using a user equilibrium approach to analyze the relationship between traffic congestion and ridesharing (Yang et al. 2010; Xu et al. 2015; Di, Liu, and Ban 2016; Di et al. 2017a; Liu and Li 2017; Li et al. 2020; Li, Liu, and Xie 2020). Afeche, Liu, and Maglaras (2018) evaluate two platform control regimes (demand-side admission control and supply side repositioning control) using a game-theoretic fluid model that describes the system equilibria. The complex relationships between the variables and decisions, including trip fare in an MoD market, can also be elucidated from the market equilibrium (Ke et al. 2020). Similarly, Chen and Di (2021) model the ridesharing equilibrium traffic patterns and also apply the model to congestion pricing and platform pricing considering the matching cost between each pair of driver

and rider by revising the ridesharing user equilibrium model in Xu et al. 2015. Di and Ban (2019) further integrate ridesharing and e-hailing service to derive a theoretical framework of general transportation network equilibria modeling shared mobility systems in congested networks. In terms of mixed fleet, Wei et al. (2019) and Mo, Chen, and Zhang (2022) analyze the mixed system from the perspective of market equilibrium. These studies focus on generating policy insights at an aggregate and stable level in the long run, whereas we aim to optimize the real-time dynamic decisions of operators at a microscopic level. Yang et al. (2021) formulate an MoD system model with a mixed fleet as a Stackelberg game where the platform acts as a leader and drivers act as followers. The user equilibrium and Stackelberg game formulation require prior information about the followers (drivers) to anticipate their responses to the decision of the leader (operator). However, the operator does not have prior knowledge of the drivers' responses to his or her decision because an explicit model is not available to describe the pairwise relationship between the operator's decision and the drivers' response. Therefore, our study proposes a model-free RL framework, which is able to adjust to vehicles' and passengers' complex spatial and temporal distribution patterns over time of day and anticipate the future dynamics and rewards of transportation systems.

Vehicle dispatching, relocation, and matching problems are equally significant research areas (Furuhata et al. 2013; Nourinejad and Roorda 2016; Ordóñez and Dessouky 2017; Ramezani and Nourinejad 2018; Zhang, Liu, and He 2019; Lei, Jiang, and Ouyang 2020; Li, Liu, and Xie 2020; Pang et al. 2020; Li and Liu 2021; Liu et al. 2022). Optimization of taxi dispatching considerably regulates how to match unserved passengers with vacant taxis efficiently. Lei et al. (2020) develop a multiperiod model that addresses path-based dynamic pricing and idling vehicle reposition problems in the MoD systems with fully compliant drivers. They solve the dynamic programming model with equilibrium constraints with the approximate dynamic programming (ADP) algorithm, improving system performance facing spatial and temporal uncertainties. Ramezani and Nourinejad (2018) present an optimal taxi fleet control strategy built on a macroscopic fundamental diagram that takes into account the interdependent impact of road traffic flows and taxi dynamics. The strategy enhances the taxi service performance and alleviates traffic congestion. Different from the centralized integer programming modeling framework, Nourinejad and Roorda (2016) frame a dynamic auction-based decentralized multiagent approach to accommodate both multipassenger and multidriver matches. Alonso-Mora et al. (2017) further consider the high capacity with up to 10 simultaneous passengers per vehicle and present a more general mathematical model that can quickly return near-optimal solutions. Miao et al. (2016) formulate a multiobjective optimization problem in a large-scale taxi system. Kim et al. (2020) extend the work by converting the multiobjective taxi dispatch problem into a network flow problem, solved via the minimum cost maximum flow algorithm. Theoretical properties are derived, and a performance guarantee is provided regardless of prediction accuracy.

Compared with traditional taxi fleets, autonomous fleets are more amenable to employing state-of-the-art management strategies (Coppola and Silvestri 2019). The platform can directly control the vehicles and relocate the vacant vehicles to the hotspots at any desired time without uncompliant drivers. Various studies have made efforts to model the optimal vehicle dispatching problem in the autonomous MoD system (Duan et al. 2020, Li and Liao 2020). Zhang and Pavone (2016) formulate the control of an MoD system with an AV fleet as a queuing network and design a robust relocation algorithm and apply it to New York. Lokhandwala and Cai (2018) devise a model that incorporates individual heterogeneous preferences and compares conventional taxi fleets to autonomous taxi fleets. Results show that deploying AV taxi fleets instead of human-driven taxi fleets have the potential to reduce the fleet size by 59% while the level of service is maintained. On the other hand, Shou et al. (2019) formulate the optimal sequential passenger-seeking strategy for MoD drivers as an MDP, employing dynamic programming and Monte Carlo simulation. The benefit of such an approach is demonstrated to the autonomous MoD platform as it dramatically improves the system reward and efficiency. Besides the rebalancing schemes, pricing incentives can also be implemented to manage the demand pattern in addition to the fleet management technique (Chen and Kockelman 2016, Karamanis et al. 2018, Vosooghi et al. 2019). Wollenstein-Betech et al. (2020) study optimal pricing and rebalancing policies for autonomous MoD systems using a dynamic fluid model.

To summarize, most model-based methods are not suitable for making real-time operational decisions for large-scale dynamic MoD systems where effective and quick decisions are necessary. Therefore, it is desirable to develop new approaches, which can better capture vehicles and passengers' complex spatial and temporal distribution patterns over time of day and make efficient operational decisions, anticipating the future impact of decisions on anticipation of transportation systems.

2.2. Reinforcement-Learning-Based Operation Methods

Online data open a new door to better capture the characteristics of stochastic demand in the future and make efficient real-time fleet management decisions. Online

operation problems in large-scale shared mobility networks are widely formulated as MDPs. For the management of shared mobility systems in large cities, this is challenging because the state space and action space are tremendously large, leading to the curse of dimensionality (Powell 2007). ADP and RL may overcome this limitation. ADP anticipates rewards-to-go via aggregate state representations with proper functional approximations (Godfrey and Powell 2002a, b, Simao et al. 2009, Ulmer et al. 2019, Vinsensius et al. 2020). Likewise, an exact model of the environment is not required in RL (Sutton and Barto 2018). Instead, optimal behavior is learned through continuous interactions between the agents and the environment. In the MoD system, the operator or a driver is an agent who makes decisions, such as the vacant vehicle movement and order dispatching policy. The agent interacts with the unknown environment and learns from the feedback, aiming at deriving an optimal policy to maximize the cumulated reward. Recently, fruitful studies have emerged focusing on the application of the RL technique in transportation systems, for instance, vehicle routing, highway traffic management, signalized control, and MoD system operations (Nazari et al. 2018, Wang et al. 2019, Chen et al. 2020b, Haydari and Yilmaz 2020).

Traditional RL uses a simple form of models like tabular approximation and piece-wise linear functions, leading to limited generalizability or optimality in practice. DNN further enhances the learning capability of RL on complex tasks (Li 2017; Qin, Tang, and Ye 2019; Haydari and Yilmaz 2020). In recent years, DRL has been widely adopted in modeling intellectually challenging decision-making problems, including our MoD fleet management problem. The literature applying DRL to MoD systems operations falls into two categories: centralized single-agent RL and decentralized multiagent RL.

Studies adopting the centralized RL model the operator or the platform as a central agent optimizing the policy to coordinate the fleet. Vehicles or drivers are fully compliant to execute the action from the agent. To maintain a balance between supply and demand and optimize metrics, such as total revenue and order fulfillment rate, Chow, Yu, and Pavone (2015), Xu et al. (2018), and O'Keeffe et al. (2021) optimize vehicle distribution by learning an optimal dispatching policy giving the number of vehicles needed to be directed from one location to another over time. Haliem et al. (2020) jointly optimize matching, pricing, and dispatching through a deep q-learning approach. Both drivers and passengers learn the best action strategies based on their dynamic utility functions based on each agent's characteristics. From another perspective, Mao, Liu, and Shen (2020) aim to generate an optimal vehicle dispatch strategy, including waiting time cost in the cost function, to ensure the service quality and equity of the taxi platform. The problem is solved by an RL algorithm using the advantage actor-critic (A2C) method.

Automated fleets can also be electrified so that charging decisions need to be integrated into the fleet operation. Turan, Pedarsani, and Alizadeh (2020) extend the work of Haliem et al. (2020) by further incorporating the charging scheme optimization problem. Tang et al. (2020) address the online operation problem of AV taxi fleets with electricity using a two-stage advisor-student method. The advisor first decides the number of vehicles that need to be repositioned or charged. After that, the student performs a combinatorial optimization precisely matching vehicles with passengers.

Nevertheless, studies using centralized single-agent RL take the assumption that the environment is stationary and each agent is fully cooperative in the operator's decisions. These assumptions neglect the selfishness of drivers and competition among individual driver agents.

In a decentralized multiagent setting (Buşoniu et al. 2010, Torreño et al. 2017, Qin et al. 2020), each driver is an agent that needs to make consecutive decisions on the action to be done at the next step. Different from the single-agent case, they have interaction with both the MoD environment and other agents. The A2C method is also widely used for its high training stability and good performance on large action spaces, where the critic approximates the state-action value function and the actor generates an optimal policy. For the dispatching or rebalancing problem, the idle drivers or vehicles decide where to go next based on their observation of the environment (Gao, Jiang, and Xu 2018; Guériau and Dusparic 2018; Lin et al. 2018b; Zhang et al. 2020; Yang et al. 2020b). Shojaeighadikolaei et al. (2020) implement a real-time pricing technique in the demand-responsive system in a prosumer-dominated microgrid. They enable the customers to make decisions that maximize their accumulative benefit in the long run.

Nevertheless, when large numbers of agents coexist, the computational cost to solve the multiagent system is exceptionally high. An effective way instead is to employ the mean-field theory to approximate the interaction effect of one agent by the average effect of its neighbor agents (Yang et al. 2018). Ke et al. (2019) and Li et al. (2019) deal with the online matching problem between available drivers and waiting customers. Wang et al. (2020b) jointly optimize the charging and reposition recommendation system for electric taxi drivers through hierarchical RL. These studies assume that drivers are cooperative and do not consider the noncooperative behavior of drivers. Shou and Di (2020) and Zhu, Ke, and Wang (2021) consider the competition among human drivers in which drivers pursue their own objectives. Shou and Di (2020) further adjust the reward offered to drivers to prevent them from

gathering at certain areas using Bayesian optimization, whereas Zhu, Ke, and Wang (2021) model the operator as a major agent and decide the incentives given to drivers. Shou and Di (2020) and Zhu, Ke, and Wang (2021) are most closely related to our research. We emphasize the major differences in problem settings between our paper and their papers. In terms of modeling, we investigate the operational problem of mixed fleet MoD systems in a realistic setting. Our problem involves both CV fleet and AV fleet as depicted in Figure 1, so the interactions between the operator and CVs must be considered. Different from our settings, the model in Shou and Di (2020) and Zhu, Ke, and Wang (2021) focuses on a CV fleet with noncooperative drivers. Also, we consider that the operator makes real-time multitask decisions dynamically across time of day, resulting in a significantly larger action space for the supervisor agent. Shou and Di (2020) investigate the peak-hour optimal service charge parameter, which is a single decision and does not vary with time, whereas Zhu, Ke, and Wang (2021) assume that the platform makes a single decision set (incentive to drivers). We also consider a more general setting with heterogeneous drivers in terms of their bounded rationality, which is not modeled in their papers.

We also briefly review studies adopting multihead neural networks. In deep learning, multiheaded neural networks are used to extract or predict distinct features from source data based on a "backbone network" (Arik, Jun, and Diamos 2018; Li, Ng, and Natsev 2019; Ahmed et al. 2021). In RL literature, a multiheaded network is often used to output different metrics in RL, such as value function, action-value function, and policy function (Flet-Berliac and Preux 2019), which, as far as we know, is not used for multitask learning (Van Seijen et al. 2017, Flet-Berliac and Preux 2019).

Existing RL studies focus on MoD systems with pure CV or AV fleets. How can mixed fleet MoD systems in real-world urban road networks be managed, especially in a real-time manner, is still unclear. Existing RL frameworks cannot solve our high-dimensional problem. We develop a novel two-sided multiagent DRL approach to optimize the dynamic fleet management problem in a large-scale MoD system with mixed autonomy. In our mixed-autonomous setting, the operator, acting as a supervisor agent, interacts with drivers who behave as individual agents. Our study considers and manages the complex interactions among human drivers, AVs, and the operator.

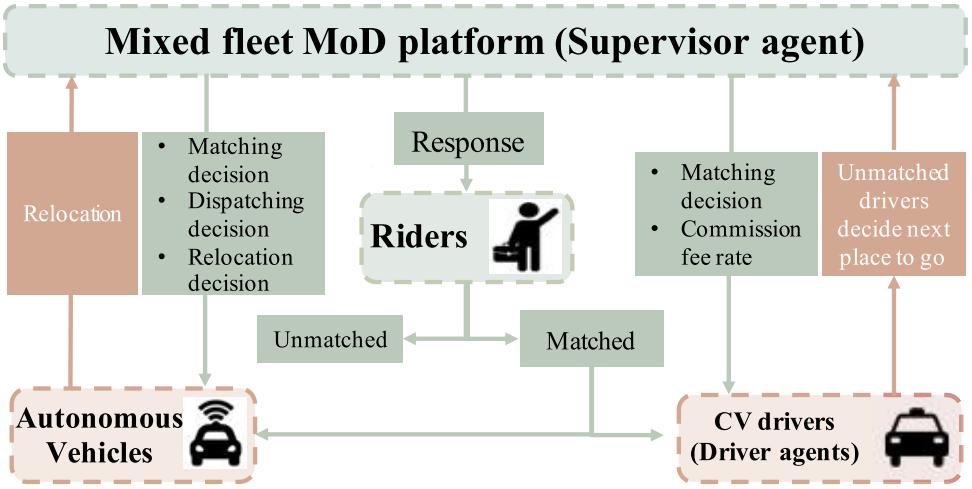
3. Model Formulation

We aim to solve the dynamic fleet management problem for an MoD system with mixed autonomy where CVs and AVs coexist to provide MoD service to passengers. The platform operator needs to dynamically make operational decisions based on the real-time system information collected from online data. First, we describe the problem in Section 3.1. Then, to model the complex dynamics of the system, we formally formulate the task as a multiagent MDP problem in Section 3.2, where both the platform and idle drivers make decisions simultaneously.

3.1. Problem Statement

This section introduces characteristics of the MoD service system providing service in an urban area. As illustrated in Figure 1, we consider a platform providing MoD services with a fleet consisting of AVs and CVs. The AV fleet is fully compliant so that the relocation movements of vacant AVs are directly controlled by the operator in a centralized manner. Conversely, human drivers play a noncooperative game and make their own relocation decisions to

Figure 1. (Color online) Illustration of the MoD System with Mixed Autonomy



maximize their compensation. Drivers make decisions simultaneously so that they compete with each other, and the system-level goal may not follow drivers' intrinsic interests. The selfishness of CV drivers may hinder the platform from achieving the primary objective to improve system-wide efficiency. The reasons are two-fold. First, we assume that each driver selects a series of optimal relocation policies to maximize his or her individual aggregated compensation. Therefore, drivers will decide where to go next based on their experience and observation rather than following the platform's relocation decisions. Second, the selfishness of human drivers will result in unbalanced supply and demand because they may gather at hotspots while seldom or never going to cold spots, resulting in unbalanced supply and demand.

The operator's goal is to maximize the total profit of the platform and achieve a high level of service. To maintain a balance between supply and demand and optimize the system-level goal, we propose that the operator determines two sets of decisions, that is, the operational decisions for AVs and the dynamic CF. On the one hand, the operator charges CF from drivers. Under such a scheme, the compensation a driver receives is the trip fare minus the CF charged. Different from the surge pricing method discouraging demand, charging CF encourages drivers to relocate themselves, which may benefit both the drivers and passengers. We adjust the CF rate both spatially and temporally to better match the demand with supply. Dynamic CF may gradually influence drivers' relocation decisions and discourage some drivers from moving to areas with anticipated excessive supply. On the other hand, the operator, who anticipates the demand and the supply using the real-time information of the entire system, makes efficient dispatching and relocation decision for the AV fleet so that AVs can complement the CV services.

Passengers send stochastic trip requests to the platform in a real-time setting. Provided an available vehicle nearby, the passenger will be assigned to a vehicle. When there is no vehicle available nearby, the passenger can wait until a maximum waiting time is reached. We make the assumption that the passengers do not have any preference for the type of vehicle. Also, if the trip requests are successfully matched to vehicles, passengers will not cancel their trips and keep waiting until they are picked up. Vehicles become idle after dropping off the passenger, and each vehicle can serve a single request at a time.

Because of the dynamics and stochasticity of supply and demand, it is challenging to quickly obtain high-quality decisions for the mixed-fleet MoD system. To capture the system uncertainty and drivers' competition behavior, we propose a two-sided multiagent MDP in which the platform is considered as a supervisor agent on one side and the human drivers are regarded as driver agents on the other side. In our approach, the supervisor agent aims to improve the system level of service quantified by the order fulfillment rate (OFR) as well as the system profit. In contrast, driver agents act as noncooperative individual agents to maximize their monetary return over the operation period. A detailed formulation of the decision-making framework is stated in the next section.

3.2. Two-Sided Multiagent Problem Setting

We model the complex and dynamic interaction and decision making of the operator and the drivers as a two-sided problem, in which the platform operator is considered as a supervisor agent on the one side and the human drivers are regarded as driver agents on the other side. In this section, we formulate the decision-making framework from the perspectives of both drivers and the platform. The decision process for both sides can be formulated as an MDP. We assume that the service region is discretized into N grids, and the operation horizon is split into T time intervals of length Δt. Let $\mathcal{N}$ denote the set of grids $\{1,2,\ldots,N\}$ and $\mathcal{T}$ denote the set of decision periods $\{1,2,\ldots,T\}$. Trip requests over one single period are aggregated together, and all the decisions are made at the end of each time period. Let D_{ijt} denote the aggregated trip requests from grid i to grid j at time t. The aggregated demand vector $\boldsymbol{D}_t$ is a vector of D_{ijt}. All vehicles in service will be available after dropping off passengers at their destinations or finishing relocation.

3.2.1. Supervisor Agent: A Single-Agent Problem. Now, we formulate the single-agent decision-making problem for the platform as an MDP. At the end of each period t, the platform makes decisions for all the idle AVs, so an AV either stays at the current location or moves to one of the neighbor grids. The MDP is formulated as a tuple $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma)$, where $\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma$ stand for the state space, action space, transition probability functions, reward functions, and the discount factor, respectively. The definitions are given as follows:

3.2.1.1. State. First, let v_{it}^a and v_{it}^c denote the number of available AVs and CVs located at grid i at time t. Then we have the AV distribution vector $\boldsymbol{V}_t^a = \{v_{it}^a\}_{i=1}^N$ and the CV distribution vector $\boldsymbol{V}_t^c = \{v_{it}^c\}_{i=1}^N$. Let $\boldsymbol{s}_t \in \mathcal{S}$ represent the state of the system at time step t. It consists of the distribution of AVs $\boldsymbol{V}_t^a$ and CVs $\boldsymbol{V}_t^c$ across the service region and trip requests D_t waiting to be answered as well as current time t (one-hot encoding):

$$\mathcal{S} := \{(\boldsymbol{V}_t^a, \boldsymbol{V}_t^c, \boldsymbol{D}_t, t) : t \in \mathcal{T}, \boldsymbol{D}_t \in \mathbb{R}_+^{N\times N}, \boldsymbol{V}_t^a \in \mathbb{R}_+^N, \boldsymbol{V}_t^c \in \mathbb{R}_+^N\}. \tag{1}$$

3.2.1.2. Action. The platform matches orders to vehicles and determines the optimal action a_t^A including idle AV relocation strategy and the CF rate.

We propose a dynamic CF scheme with both spatial and temporal dynamics, which has a direct effect on the drivers' behavior and avoids oversupply situations. We define the ratio of demand to supply (DS_{it}) as the number of requests dividing the number of vehicles (CVs and AVs) and CF as the amount of fare charged from drivers serving orders at grid i at time t. A lower DS means the grid is oversupplied, and a larger DS indicates the grid is undersupplied. The amount of CF is calculated by multiplying the order trip fare by a commission rate CR, and CR_{it} represents the commission rate in grid i at time t. Now we write the commission rate as a function of DS in a simple but efficient form. The platform's objective is to maintain DS approximately one, indicating a balance between demand and supply. If DS_{it} of a grid i is below one at time t, CR_{it} is supposed to increase to discourage drivers from arriving at the grid, whereas in a grid i with DS_{it} above one, CR_{it} is expected to be small. We also consider that drivers need to be charged a base commission fee when it is not oversupplied. The base commission fee is similar to the commonly observed service fee in the current ride-hailing market. To describe such a relationship, we calculate CR_{it} by a piece-wise linear function with a CF coefficient c_{it}:

$$CR_{it} = \begin{cases} c_{it} \times (1 - DS_{it}) + \eta & \text{if } DS_{it} \le 1 \\ \eta & \text{otherwise,} \end{cases} \tag{2}$$

where η is a predefined base commission rate required by the platform to guarantee the platform's income during peak hours. Different from Shou and Di (2020) where a CF coefficient is fixed across time and is examined for a long-term equilibrium problem, we focus on a real-time dynamic problem, where c_{it} varies across time of day as well as the locations. Note that c_{it} can be any value belonging to the set of candidate values $\mathcal{C}$, including elements ranging from 0.1–0.8 with equal distances of 0.1. Let $\boldsymbol{C}_t$ denote the vector of c_{it} at time t. The unmatched AVs are allowed to either move into one of the neighboring grids or stay in the current grid. More concisely, we denote the set of neighbor grids of grid i including itself as $\mathcal{B}_i$. Then, the action space can be defined as $\mathcal{A}^A(s_t) = \{\boldsymbol{\mathcal{X}}_t, \boldsymbol{C}_t\}$, where $\mathcal{X}_t$ denotes the AV relocation decision vector at time t, satisfying the following condition:

$$\boldsymbol{\mathcal{X}}_t : \left\{ x_{ijt} : \sum_{j \in \mathcal{B}_i} x_{ijt} = v_{it}, i \in \mathcal{N}, x_{ijt} \in \mathbb{R}_+ \right\}, \tag{3}$$

where x_{ijt} represents the number of vehicles relocated from grid i to its neighbor j and v_{it} is the total number of idle vehicles in grid i at time t.

3.2.1.3. Reward. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ collects immediate reward $r_t^A(\boldsymbol{s}_t, \boldsymbol{a}_t^A, \boldsymbol{s}')$ after transiting from $\boldsymbol{s}_t$ to a new state $\boldsymbol{s}'$ under action $\boldsymbol{a}_t^A$. There are two important performance metrics in MoD systems. The system profit measures the financial sustainability of the platform, and the number of fulfilled requests measures the level of service. We formulate the reward function for the supervisor agent as a weighted function of the platform profit (i.e., the trip profit g_t^A and the commission fee CF_t^A) and the number of fulfilled requests f_t^A:

$$r_t^A = g_t^A + CF_t^A + \omega f_t^A, \tag{4}$$

where ω denotes the weight of the number of fulfilled requests in the reward function of the supervisor agent and different values of ω reflect the preference of the platform toward the two performance measures. Because the objective of the platform is to improve the reward in the long-term, at each step, the objective function (total discounted reward) of the supervisor agent is

$$R_t^A = \sum_{k=t}^{T} \gamma^{k-t} r_k^A. \tag{5}$$

Note that in practice, the operator may want to provide incentives to drivers to encourage them to drive to undersupplied areas rather than discourage them from gathering in oversupplied areas. Our framework can be easily adapted to this case by providing drivers CF waivers as a reward. The CF reward (waiver) can be formulated as a function of the ratio of supply and demand similar to Equation (2).

3.2.2. Driver Agents: A Noncooperative Multiagent Problem. Next, we formulate the noncooperative driver decision problem to maximize individual compensation as a multiagent MDP defined by the tuple $(M, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The difference with the single-agent MDP is that the system consists of M agents, and each agent has a partial observation of the state. We consider each idle driver as an agent and assume the vehicles in the same grid at the same time are homogeneous. Because the number of idle drivers is changing over time, the number of agents M_t varies temporally.

3.2.2.1. State. The drivers draw a partial observation from the global state $\boldsymbol{s} \in \mathcal{S}$ with the operator in the training phase and only have a partial observation of the system in the evaluation and application phase.

3.2.2.2. Observation. Let $\mathcal{O}_t^m$ be the observation space of agent m at time t. Actually, the environment state $\boldsymbol{s}_t$ is not fully observable by the drivers, which means agent m draws a private observation $\boldsymbol{o}_t^m \in \mathcal{O}_t^m$ corresponding to current state $\boldsymbol{s}_t$. We assume the observation of the drivers contains two components, including the index l_m of grid m and current time t, so that we can write that $\boldsymbol{o}_t^m = \{l_m, t\}$, implying drivers at the same grid have identical observations.

3.2.2.3. Action. The allowable action $a_t^m \in \mathcal{A}_t^m(o_t^m)$ of an individual agent m is either relocating to one of the neighboring grids or remaining in the current grid. It specifies locations an agent is able to move to at $t+1$ by giving a set of discrete choices with cardinality $|\mathcal{B}_i|$, that is, the action space of the drivers depends on their current locations and neighbor grids. The joint action $\mathbf{a}_t^C = \{a_t^m\}_{m=1}^{M_t}$ of all the agents at time t instructs the relocation action of all the idle drivers; $\mathbf{a}_t^C \in \mathcal{A}_t^C = \mathcal{A}_t^1 \times \mathcal{A}_t^2 \times \ldots \mathcal{A}_t^{M_t}$, where $\mathcal{A}_t^C$, represents the joint action space for all the drivers.

3.2.2.4. Reward. The reward function for a driver collects the immediate reward after taking relocating action. Because in our case, drivers aim to maximize their own discounted cumulative reward, we take drivers' compensation as the immediate reward $r_t^m(\boldsymbol{s}_t, \mathbf{a}_t^C, \boldsymbol{s}_{t+1})$ of agent m provided that the driver is matched to a trip request after taking the relocation action. The compensation of driver m is calculated as the trip profit g_t^m (trip fare minus the distance-based travel cost) minus the CF and the relocation cost (RC_t^m):

$$r_t^m = g_t^m - CF_t^m - RC_t^m, \tag{6}$$

where $RC_t^m = u$ if the action a_t^m indicates the driver does not stay at the current location l_m in the following time step, where u is a predefined cost for relocating to a neighbor grid and is identical across all the drivers. Our paper assumes that the relocation cost is proportional to the relocation distance. Note that in practice, the travel time is stochastic because of congestion, which will lead to varying relocation costs and affect the relocation decision. Therefore, estimating the road travel time according to historical data and formulating the relocation cost as a function of travel time will be a better practice provided sufficient historical data. If $a_t^m = l_m$, $RC_t^m = 0$. The objective function for agent m is

$$R_t^m = \sum_{k=t}^{T} \gamma^{k-t} r_k^m. \tag{7}$$

During the training process, driver agents update their value functions and policies based on the experience of all the drivers. Nevertheless, during the execution (or evaluation) stage, decentralized execution is conducted, which means the information of other drivers is not accessible to driver agents anymore. This is considered the centralized learning and decentralized execution scheme in the literature (Lin et al. 2018a, Li et al. 2019).

3.3. The Two-Sided Operation Problem for MoD Platform with Mixed Autonomy

In summary, the dynamic decision procedure for an MoD platform with mixed autonomy is a two-sided problem. The goal of the operator and each driver is to learn an optimal policy π^*. We denote the operator's policy as $\pi^A : \mathcal{S} \times \mathcal{A} \to [0,1]$ and the driver m's policy as $\pi_m^C : \mathcal{O}^m \times \mathcal{A}^m \to [0,1]$. Starting from $t = 0$, at the end of each time step, after the order matching procedure, vacant AVs and CVs enter the idle status. Then, the central agent specifies an action $\mathbf{a}_t^A \in \mathcal{A}_t$, when occupying the state $\boldsymbol{s}_t \in \mathcal{S}$, according to the policy π^A. At the same time, each idle driver m choose his or her action $a_t^m \in \mathcal{A}_t^m$ based on his or her observation $\boldsymbol{o}^m \in \mathcal{O}^m$ according to the policy π_m^C. Then, the environment transits into the next state $\boldsymbol{s}' \in \mathcal{S}$ following $p(\boldsymbol{s}' \mid \boldsymbol{s}_t, \mathbf{a}_t^A)$ when action $\mathbf{a}_t^A$ is taken. Transition probabilities from one global state to another are given in $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$. The joint action $\mathbf{a}_t = \mathbf{a}_t^C \cup \mathbf{a}_t^A$ triggers a state transition from s_t to s' with transition probability $p(\boldsymbol{s}' \mid \boldsymbol{s}_t, \mathbf{a}_t)$.

Note that the state transition probability is deterministic after the action is taken before new orders come. The uncertainty comes from the stochastic demand arriving over time, which will be learned by the agents through continuous interaction with the environment. The state transition takes one period to finish. Thus, the system state becomes $\boldsymbol{s}'$ at the end of $t+1$. The central agent and all the drivers finishing relocation receive immediate rewards $r^A(\boldsymbol{s}_t, \mathbf{a}_t, \boldsymbol{s}')$ and $r^m(\boldsymbol{s}_t, \mathbf{a}_t, \boldsymbol{s}')$, respectively, by following the system dynamics. The iteration continues until the end of the horizon is reached (i.e., $t = T$).

In this model setting, there are underlying interactions between the AVs and CVs. CVs affect AVs in that the distribution of the CVs is an element of the state the supervisor agent perceives, based on which the supervisor agent relocates the AVs. Besides, in the sequential decision-making problem, the supervisor agent has no prior information on drivers' distribution dynamics resulting from drivers' joint behavior. Our two-sided approach enables the supervisor agent to anticipate the potential dynamics of drivers in the future given the current system state through continuous interaction with the environment. Therefore, the relocation of AVs is largely affected by the current and potential future distribution of CVs, resulting from drivers' joint actions. Furthermore, AVs affect CVs indirectly through trip demand. The impact is two-fold. On the one hand, the distribution of AVs affects the distribution of served demand and, naturally, the distribution of unserved demand. On the other hand, the platforms' reputation is largely dependent on the system level of service, which is reflected in the system level of service. With the AVs relocated in a way that can complement the drivers' service in areas with a supply shortage, the order fulfillment rate can be improved, which potentially increases the trip demand in the long run.

4. A Two-Sided Multiagent DRL Approach

Solving the stochastic dynamic problem is not trivial because the space of the state space and action is

extremely large, and the agents do not have an exact model of the environment. The DRL algorithm address the curse of dimensionality through evaluating and modifying current policies. The agents interact with their environment in discrete time steps in a model-free way.

We propose a two-sided multiagent DRL approach for mixed-fleet management optimization. Our approach incorporates the complex interactions between the platform and two types of fleets but does not rely on an accurate predefined model of the system. The structure of our algorithm is illustrated in Figure 2. The operator is a supervisor agent, and drivers act as individual agents. They interact in a common environment simultaneously, and their decisions will be influenced by each other. It is noted that Shou and Di (2020) focus on the noncooperative decision making of human drivers using multiagent RL. Most of the existing studies use multiagent RL to examine the drivers' behavior, which is similar to the objective shown on the right side of Figure 2. Nevertheless, our problem also considers the centralized decision making of the operator on the left side in Figure 2 and the complex interactions among drivers, AVs, and the operator.

Two types of RL algorithms are studied in the literature, namely, the value-based algorithm and the policy-based algorithm. Because the state space and action space are huge, value-based methods like Q-learning are not applicable in this case. Instead, we focus on the A2C method, a policy-based technique, which is much more efficient than the pure value-based or policy-based methods as it takes advantage of both of them. We formulate a two-sided multiagent A2C algorithm to simultaneously train the two types of agents. To our best knowledge, this is the first paper that formulates multiagent A2C for mixed-fleet MoD systems. The operator who acts as a supervisor agent learns its optimal strategy in a "centralized" manner, and the driver agents who have their own objectives learn their optimal strategy in a "decentralized" manner. Both types of agents consist of a critic and an actor. The value network (critic) accesses the value of current policies through examining the environment's reaction, and then the policy network (actor) takes the responsibility to generate actions according to the observation (i.e., the operator acts based on the system state, and the idle driver agent acts based on the partial observation). In this study, we pick DNNs to represent both the critics and actors for their flexibility. DNNs are kept updated during training by learning from past experience. However, in our problem formulation, the operator makes two types of decisions, namely, the dynamic CR, which influences drivers' behavior, and the relocation decisions for optimizing AV operations. To address this multitask problem, we formulate a two-head network to enable efficient sequential decision making for the operator.

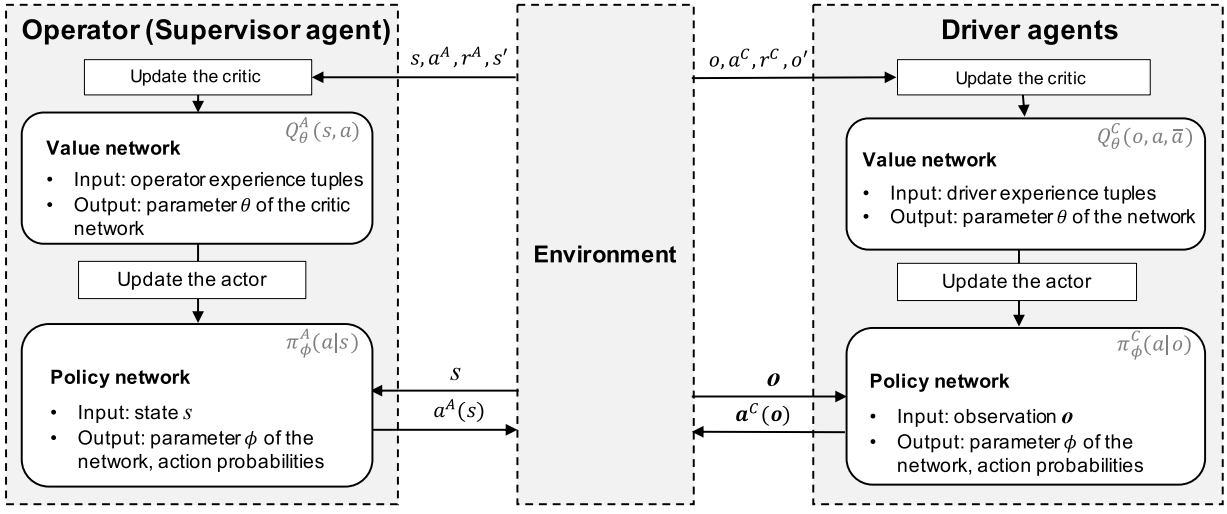
Following the notations in Section 3, we discuss the A2C algorithm in Section 4.1 firstly. Then, we describe the learning algorithm for the operator and drivers in Sections 4.2 and 4.3, respectively. We also bring out efforts we made to adapt the algorithm to carry out proper actions. Finally, the algorithm is detailed in Section 4.4.

4.1. Advantage Actor Critic Method

We devise a two-sided A2C method to train the supervisor agent and the driver agents for making optimal sequential decisions. We first present the A2C algorithm in a single-agent setting in this section, followed by the multiagent version.

4.1.1. A2C Algorithm. There are two types of RL methods in the literature, namely, the value-based method and the policy-based method. The A2C method (Konda and Tsitsiklis 2000, Mnih et al. 2016) combines the strengths of value-based and policy-based methods

Figure 2. Illustration of the Two-Sided Multiagent DRL Framework



while avoiding their drawbacks. The actor recommends action by providing probabilities over the action space, and the critic accesses the value of states and actions. Next, we explain the critic and actor and then present the A2C method.

4.1.1.1. Critic. To approximate the state-action (or observation-action for the case of drivers) values in the A2C method, we use a deep neural network, which is called a Deep Q Network (DQN). The DQN $Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})$ (value network) is parameterized by θ. Similar to the supervised learning by neural networks, DQN updates its parameters by minimizing the loss function $\mathcal{L}(\theta)$:

$$\mathcal{L}(\boldsymbol{\theta}) = (r(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') + \gamma max_{\boldsymbol{a}'} Q_{\boldsymbol{\theta}'}(\boldsymbol{s}',\boldsymbol{a}') - Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}))^2. \quad (8)$$

The Q value function is fit by minimizing the loss using the stochastic gradient descent method. Note that we calculate the target Q value of the current state by the target network $Q_{\boldsymbol{\theta}'}$. We keep updating $Q_{\boldsymbol{\theta}}$ during the training while only copying it to $Q_{\boldsymbol{\theta}'}$ after several steps to stabilize the learning procedure.

4.1.1.2. Actor. The actor is a policy that maps an input state to an output of a probability distribution in feasible action space in the current state. The idea is similar to the policy gradient methods. We also use a DNN $\pi_{\boldsymbol{\phi}}$ (policy network) parameterized by $\boldsymbol{\phi}$ to approximate the policy, similar to the value network. We calculate the advantage value by the difference between the Q value and the state value to know how good is a specific action compared with the rest. The policy gradient is written as

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi}) &= \mathbb{E}_{s,a}\left(\nabla_{\boldsymbol{\phi}} log \pi_{\boldsymbol{\phi}}(\boldsymbol{a}\,|\,\boldsymbol{s})\left(Q_{\boldsymbol{\theta}'}(\boldsymbol{s},\boldsymbol{a}) - V_{\boldsymbol{\theta}}(\boldsymbol{s})\right)\right)\\ &= \mathbb{E}_{s,a}\left(\nabla_{\boldsymbol{\phi}} log \pi_{\boldsymbol{\phi}}(\boldsymbol{a}\,|\,\boldsymbol{s})\left(r(\boldsymbol{s},\boldsymbol{a}) + \gamma V_{\boldsymbol{\theta}'}(\boldsymbol{s}') - V_{\boldsymbol{\theta}}(\boldsymbol{s})\right)\right).\end{aligned} \quad (9)$$

Note that we update the policy network by maximizing the policy gradient, which is equivalent to minimizing the negative policy gradient.

In summary, the A2C algorithm maintains value network $Q_{\boldsymbol{\theta}}$ and a policy network $\pi_{\boldsymbol{\phi}}$. We update $Q_{\boldsymbol{\theta}}$ by minimizing the loss function in Equation (8) and $\pi_{\boldsymbol{\phi}}$ is updated by the gradient in Equation (9). These two networks are updated simultaneously so that the policy gradient estimation can directly use the value approximated by $Q_{\boldsymbol{\theta}}$.

4.1.2. Multiagent Reinforcement Learning. MARL refers to learning problems in systems where multiple agents behave and learn simultaneously while sharing a common environment. The objective of each driver is to derive an optimal policy to maximize his or her cumulative reward. Each agent optimizes its policy by the A2C algorithm. For the drivers, because they only have partial observations instead of the system state, and the action value depends on the joint action of other agents, the value network of agent m becomes $Q^m_{\theta}(\boldsymbol{o}^m,\mathbf{a})$ and the policy network becomes π^m_{ϕ}. For a real-world problem with hundreds or even thousands of agents, it is computationally intractable to maintain a large number of DNNs, so we assume that all the drivers are homogeneous and share one value network, policy network, and reward function.

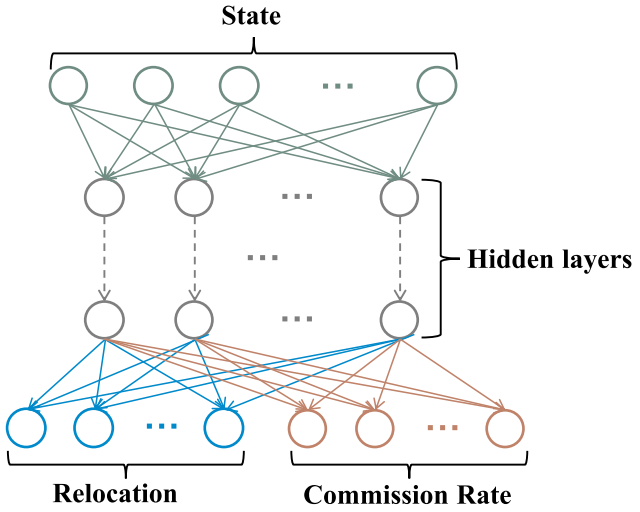
4.2. Centralized Decision Making for the Operator

The operator makes system-view decisions of AV relocation and rate of CF, following the A2C method elaborated in the previous subsection.

The state space and the action space of the original single-agent (i.e., the operator) decision-making problem become extremely large when there are hundreds or even thousands of grids in the study area. To develop a more scalable and efficient algorithm, we partition the complex problem into a multiagent RL problem. Each grid in the service region is assigned with an agent, indexed by $n \in \mathcal{N}$, where $\mathcal{N} = \{n\}_{n=1}^{N}$. The state of each agent includes the elements in Equation (1) and the one-hot encoding of the current grid index. The action of agent n is denoted by (HTML translation failed), which also includes two parts: the relocation action $\boldsymbol{a}_t^{n,r}$ and the CF action $\boldsymbol{a}_t^{n,c}$. For agent n, the reward function is written as the total profit of the platform averaged on each grid: $r_t^n = (g_t^A + CF_t^A + \omega f_t^A)/N$. This makes the agents act cooperatively to optimize the platform profit and the order fulfillment rate. We denote the value network for the operator as $Q^A_{\boldsymbol{\theta}}$ and the policy network as $\pi^A_{\boldsymbol{\phi}}$. To make the A2C method work fluently in our problem, we modify the policy network and formulate a two-head policy network.

4.2.1. Two-Head Policy Network. Because the operator policy network $\pi^A_{\boldsymbol{\phi}}$ needs to decide two types of actions, whose action probabilities cannot be given in a neural network with a single output layer, we formulate a two-head policy network to differentiate the two types of actions while keeping the common information in the hidden layers. The structure of a policy network with two heads (i.e., the relocation head and the CF head) is illustrated in Figure 3. The relocation head outputs the probability distribution of the relocation actions, whereas the CF end outputs the probability distribution of all possible values of c_{it}. They both have a ReLU activation function combined with a soft-max layer to ensure the action probabilities are added up to one. The action selector selects actions according to the probability distribution output by the policy network. Specifically, to make AV relocation decisions, each grid agent chooses the next location of each idle vehicle in the grid at the current time according to the relocation probabilities given by the relocation head. Specifically, denote the action probability values output by the relocation head of the policy network as p_{ijt} for each grid i,

Figure 3. (Color online) The Two-Head Operator Policy Network



then the number of vehicles relocated from grid i to its neighbor grid is calculated by the action probabilities: j: $x_{ijt} = v_{it} \times p_{ijt}$. For the commission rate decision, each grid agent selects one value of CR following the distribution given by the CF head in the policy network. Note that the selected commission rate is a mean value, whereas the real commission rate is randomly sampled from a normal distribution with mean value c_{it} and variance σ.

4.2.1.1. The Entropy Loss.

To balance the exploration and exploitation and avoid the policy being trapped in a local optimum, we subtract an entropy loss from the loss function (policy gradient) of the policy network to prevent the policy from being too certain over the action space. The entropy of information is defined to be $H(\pi_{\boldsymbol{\phi}}) = -\sum_a \pi_{\boldsymbol{\phi}}(a \mid \boldsymbol{s}) log \pi_{\boldsymbol{\phi}}(a \mid \boldsymbol{s})$ for discrete action distributions. As we have two types of discrete actions (i.e., relocation action and CF rate action), we compute the entropy as

$$H(\pi^A_{\boldsymbol{\phi}}) = -\sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} \pi^A_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{s}) \pi^A_{\boldsymbol{\phi}}(\boldsymbol{c} \mid \boldsymbol{s}) log(\pi^A_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{s}) \pi^A_{\boldsymbol{\phi}}(\boldsymbol{c} \mid \boldsymbol{s})). \tag{10}$$

Then the loss function is

$$\mathcal{L}^A = \nabla_{\boldsymbol{\phi}} J^A(\boldsymbol{\phi}) - \beta H(\pi^A_{\boldsymbol{\phi}}), \tag{11}$$

where $\nabla_{\boldsymbol{\phi}} J^A(\boldsymbol{\phi})$ is as given in Equation (9), and β is the entropy coefficient.

Different from the existing literature, the supervisor agent indirectly learns the decision behavior of drivers because the drivers' joint decision impacts the future system state, which in turn affects the optimal actions and rewards of the supervisor agent.

4.2.2. Two Separate Policy Networks.

We benchmark the performance of our two-head policy network against the performance when maintaining two separate policy networks (actors) to give the correspondent actions (Yang et al. 2017). Though we have multiple actors, we do not need to add new value networks (critics) because a single critic can properly guide the actors' work. As such, the critic needs to give multiple state-action values corresponding to each actor. The loss function is given as follows instead of the loss function in Equation (8)

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{K} (r(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') + \gamma max_{\boldsymbol{a}'} Q^k_{\boldsymbol{\theta}'}(\boldsymbol{s}', \boldsymbol{a}') - Q^k_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}))^2, \tag{12}$$

where k is the identity number of the actors and K is the total number of actors, which equals two in our case. With the critic, we update the actor k using the following gradient:

$$\nabla_{\boldsymbol{\phi}_k} J_k(\boldsymbol{\phi}_k) = \mathbb{E}_{\boldsymbol{s}, \boldsymbol{a}_k} \big(\nabla_{\boldsymbol{\phi}_k} log \pi_{\boldsymbol{\phi}_k}(\boldsymbol{a}_k \mid \boldsymbol{s}) \big(r(\boldsymbol{s}, \boldsymbol{a}) + \gamma V^k_{\boldsymbol{\theta}'}(\boldsymbol{s}') - V^k_{\boldsymbol{\theta}}(\boldsymbol{s}) \big) \big), k = 1, 2, \tag{13}$$

where $\boldsymbol{a}_1$ is the relocation action $\boldsymbol{x}$ and $\boldsymbol{a}_2$ is the commission rate action $\boldsymbol{c}$.

4.2.2.1. The Entropy Loss.

To balance the exploration and exploitation and avoid the policy being trapped in a local optimum, we subtract an entropy loss from the policy gradient $\nabla_{\boldsymbol{\phi}_k} J_k(\boldsymbol{\phi}_k)$ to prevent the policy from being too certain over the action space. The entropy of information is defined as $H(\pi_{\boldsymbol{\phi}}) = -\sum_{\boldsymbol{a}} \pi_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{s}) log \pi_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{s})$ for discrete action distributions, so that the entropy loss for each actor k is calculated by

$$H(\pi_{\boldsymbol{\phi}_k}) = -\sum_{\boldsymbol{a}_k} \pi_{\boldsymbol{\phi}}(\boldsymbol{a}_k \mid \boldsymbol{s}) log \pi_{\boldsymbol{\phi}_k}(\boldsymbol{a}_k \mid \boldsymbol{s}). \tag{14}$$

Therefore, the loss function of actor k is

$$\mathcal{L}^A_k = \nabla_{\boldsymbol{\phi}} J^A(\boldsymbol{\phi}_k) - \beta H(\pi^A_{\boldsymbol{\phi}_k}). \tag{15}$$

4.3. Decentralized Decision Making with a Mean-Field Approximation for Drivers

All drivers act strategically and evaluate their policies simultaneously according to the joint actions of all other agents. Consequently, the learning becomes intractable when the number of agents expands essentially owing to the curse of dimensionality and the exponential expansion of agent interactions. To address these issues, we articulate the Q-functions using the pairwise local interactions to approximate the interaction between one driver agent and all other driver agents:

$$Q^m(\boldsymbol{o}^m, \boldsymbol{a}) = \frac{1}{|\mathcal{K}_m|} \sum_{k \in \mathcal{K}(m)} Q^m(\boldsymbol{o}^m, a^m, a^k), \tag{16}$$

where $\mathcal{K}(m)$ consists of indexes of neighboring agents

of agent m. Following the mean field theory in Yang et al. (2018) and Li et al. (2019), the local interactions can further be approximated by mean action $\overline{a}$, which gives $Q^m(\boldsymbol{o}^m, \mathbf{a}) \approx Q^m(\boldsymbol{s}, a^m, \overline{a}_m)$. The mean action $\overline{a}$ can be represented by the demand-supply ratio at each grid. We further assume that the drivers are homogeneous and share the same value network and one policy network π^C_ϕ, so that the experience of all the drivers can contribute to the policy update. In the multiagent setting, the loss function of the critic becomes

$$\mathcal{L}^C(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{o}^m, a^m, \boldsymbol{o}^{m'}}\left(r(\boldsymbol{o}^m, a^m, \boldsymbol{o}^{m'}) + \gamma max_{a'}\mathbb{E}_{\overline{a}'}\left(Q_{\boldsymbol{\theta}'}(\boldsymbol{o}', a', \overline{a}')\right) - Q_{\boldsymbol{\theta}}(\boldsymbol{o}^m, a^m, \overline{a}')\right)^2, \tag{17}$$

and the policy gradient estimate becomes

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} J^C(\boldsymbol{\phi}) &= \mathbb{E}_{\boldsymbol{o}^m, a^m}\left(\nabla_{\boldsymbol{\phi}} log \pi_{\boldsymbol{\phi}}(\boldsymbol{o}^m \mid a^m)\left(Q_{\boldsymbol{\theta}'}(\boldsymbol{o}^m, a^m, \overline{a}^{m'}) - V(\boldsymbol{s})\right)\right) \\ &= \mathbb{E}_{\boldsymbol{o}^m, a^m}\left(\nabla_{\boldsymbol{\phi}} log \pi_{\boldsymbol{\phi}}(\boldsymbol{o}^m \mid a^m)\left(Q_{\boldsymbol{\theta}'}(\boldsymbol{o}^m, a^m, \overline{a}^{m'})\right.\right. \\ &\quad \left.\left. - \mathbb{E}_{a^m, \overline{a}^m}\left(Q_{\boldsymbol{\theta}'}(\boldsymbol{o}^m, a^m, \overline{a}^m)\right)\right)\right).\end{aligned} \tag{18}$$

The loss function for the policy network also includes the entropy: $\mathcal{L}^C(\boldsymbol{\phi}) = -\nabla_{\boldsymbol{\phi}} J^C(\boldsymbol{\phi}) - H(\pi^C_{\boldsymbol{\phi}})$, where $H(\pi^C_{\boldsymbol{\phi}}) = -\sum_{\boldsymbol{a}\in\mathcal{A}} \pi^C_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{o}) log \pi^C_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{o})$.

4.4. The Two-Sided Multiagent Deep Reinforcement Learning Algorithm

We summarize the two-sided DRL algorithm in Algorithm 1. First, we initialize the value networks, policy networks, and the algorithm parameters that need to be initialized, such as the learning rate, updating batch size, and the neural network hyperparameters. Two experience buffers are established to store past experience. Then, we have two loops: the outer loop counts the number of episodes, and the inner loop iterates over time in one episode. Each episode has two stages, and the details of the two stages are given as follows.

Algorithm 1 (The Two-Sided Multiagent DRL Algorithm)

Initialization:
Initialize the driver value networks $Q^C_{\boldsymbol{\theta}}(\boldsymbol{o}, a, \overline{a})$, driver policy network $\pi^C_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{o})$
Initialize the operator value networks $Q^A_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$, operator policy network $\pi^A_{\boldsymbol{\phi}}(\boldsymbol{a} \mid \boldsymbol{s})$
Initialize replay buffers B^C and B^A
Set the iteration counter $n = 1$, and set the maximum number of iteration N
for $1 \le n \le N$ **do**
  Reset environment and get the initial state $\boldsymbol{s}_0$; each agent draws an initial private observation $\boldsymbol{o}^m_0$
  Stage 1: Collect experience
  for $0 \le t \le T$ **do**
    Order assignment:
    Match orders to vehicles in the current grid and its neighbor grids following the matching heuristic
    Action selection:
    Sample actions $\boldsymbol{a}^C_t$ according to the policy $\pi^C_{\boldsymbol{\phi}}$ for each idle driver agent
    Sample actions $\boldsymbol{a}^A_t$ according to the policy $\pi^A_{\boldsymbol{\phi}}$ for the supervisor agent
    Action execution:
    The operator executes action $\boldsymbol{a}^A_t$; each available driver agents takes its action $\boldsymbol{a}_i$
    The operator observes r^A_t and $\boldsymbol{s}_{t+1}$; each driver agent observes $r_{i,t}$, $\overline{a}_{i,t}$ and $\boldsymbol{o}_{i,t+1}$
    Store the experience tuple $\{\boldsymbol{o}^m_t, a^m_t, r^m_t, \overline{a}^m_t, \boldsymbol{o}^m_{t+1}\}$ into the replay buffer B^C and $\{\boldsymbol{s}_t, \boldsymbol{a}^A_t, r^A_t, \boldsymbol{s}_{t+1}\}$ into B^A
  end for
  Stage 2: Update parameters
  for $m_1 = 1$ to M_1 **do**
    Sample a batch of experience tuples from buffer B^C
    Update the driver value network $Q^C_{\boldsymbol{\theta}}$
  end for
  for $m_2 = 1$ to M_2 **do**
    Sample a batch of experience tuples from buffer B^C
    Update the driver policy network $\pi^C_{\boldsymbol{\phi}}$
  end for
  for $m_3 = 1$ to M_3 **do**
    Sample a batch of experience tuples from buffer B^A
    Update the operator value network $Q^A_{\boldsymbol{\theta}}$
  end for
  for $m_4 = 1$ to M_4 **do**
    Sample a batch of experience tuples from buffer B^A
    Update the operator policy network $\pi^A_{\boldsymbol{\phi}}$
  end for
  Decrease the exploration parameter ϵ
  Decrease the learning rate α
end for
Return the value network and the policy network for drivers and the policy network for the operator

In the first stage, the agents explore the environment and collect experience. At the beginning of each time period, orders are matched with vehicles in a greedy manner. We describe the matching algorithm as follows. We match orders with available vehicles following a four-step heuristic algorithm similar to the two-step heuristic algorithm in Lin et al. (2018b). This heuristic naturally ensures the priority of drivers to take orders. We assume that customers are matched by flowing a first-come-first-serve rule. We first sort all orders aggregated in the current period departing from the same grid in ascending order by the trip request time.

Multiple orders with the same request time are further sorted by their trip prices in descending order. In the first step, orders are assigned to available CVs in the same grid following the sorted order. In the second step, the remaining orders are assigned to available AVs in the current grid. In the third step, the remaining requests are assigned to available CVs in the neighbor grids. In the last step, the remaining requests are assigned to available AVs in the neighbor grids. If there are no more waiting orders after a step, the next step will not be implemented. Passengers not matched in the current period will wait for the next round of matching if their maximum endurable waiting times are not reached. Otherwise, they will abandon their orders and leave the system. This matching procedure allows for cross-regional matching, which simulates the commonly observed relocation activities in practice. It also reflects the priority in drivers to ensure their income. After the assigning procedure, idle drivers and the operator act according to the current policy, then the operator and the drivers record their experience tuples $\{\boldsymbol{o}_t^m, a_t^m, r_t^m, \overline{a}_t^m, \boldsymbol{o}_{t+1}^m\}$ and $\{\boldsymbol{s}_t, \boldsymbol{a}_t, r_t^A, \boldsymbol{s}_{t+1}\}$ into two replay buffers after each action. The operator and idle drivers keep taking action and observing reward until T is reached.

Note that our matching scheme also avoids the famous "wild-goose chasing (WGC)" phenomenon in ride-hailing platforms without a maximum dispatch radius (MDR), which is the maximum distance within which a driver and a passenger can be matched, or the MDR is relatively far. It is because when the demand volume is significantly higher than the number of available drivers, the available drivers may be spread across the service region scarcely. Under this circumstance, matches between vehicles and customers will be far away from each other on average. Far matching distance leads to significant time spent on the way to hot spots to pick up passengers (Castillo, Knoepfle, and Weyl 2017). Instead, our matching scheme has a small MDR, which prevents the WGC. Also, the drivers are less likely to chase certain CF changes because the commission rate decisions made by the RL agent consider the long-term effect of the decisions and the time required for drivers to react.

In the second stage, we randomly sample a batch of experiences to update the four networks. Actions of all the agents are executed in the simulation environment, and such procedure repeats until the end of the operation period. In addition, we maintain training networks and target networks and copy the weights from training networks to target networks every τ episode to ensure the stability of the training process. Finally, the algorithm terminates when $n = N$ and outputs the trained policy networks. After the training, the value networks are no more needed when executing the trained agents.

5. Numerical Experiments

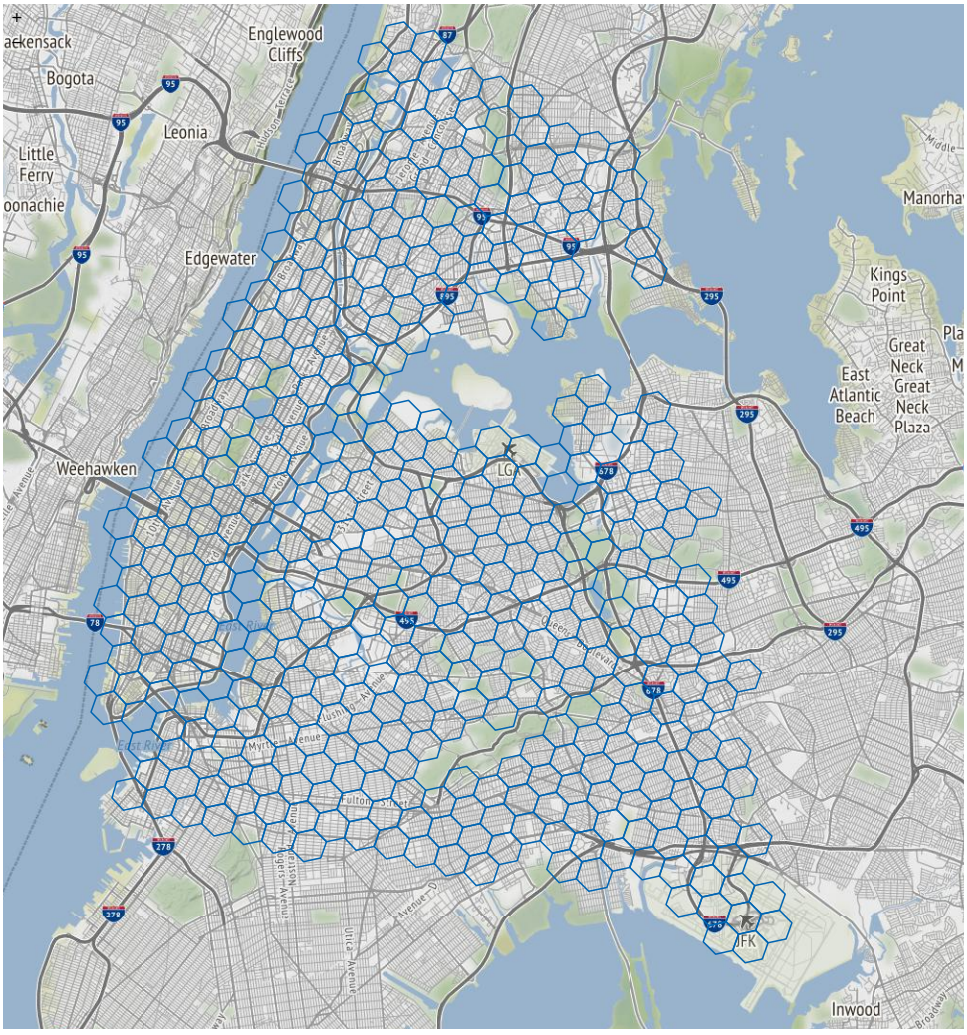
To test the performance of the proposed two-sided multiagent DRL optimization approach, we conduct numerical experiments in a realistic setting in New York City and discuss the results in this section. First, we present our simulator and experiment settings in Section 5.1. We then test the performance of our algorithm in Section 5.2 and vary the AV penetration rate in Section 5.3, followed by the benefit of CF when incorporating different fleet components in Section 5.4. Finally, we conduct sensitivity analysis by varying the demand volume and AV penetration in Section 5.5.

5.1. Experiment Settings

This subsection gives the details about the simulation settings in our numerical experiments, including how we incorporate the bounded rationality of drivers. Since our problem optimizes the MoD system operation considering stochastic future demand, which is unknown to the supervisor agent and driver agents, and the agent makes decisions for the current period based on its experience and current information, it is crucial to capture this in the simulation. In 5.1.1, we present the simulator settings to simulate the entire MoD system operations, and in 5.1.2, we give the details of the taxi trip data set to generate the orders in the simulation. Finally, 5.1.3 demonstrates how we model the bounded rationality of drivers.

5.1.1. Simulator Settings. For training and testing the proposed algorithm in Section 4, an interactive environment is an intrinsic part in our two-sided DRL framework. An effective solution to this is to build a simulation environment in transportation studies (Lin et al. 2018b, Ke et al. 2020). We design a simulator that simulates a large-scale MoD system providing service with a mixed taxi fleet of CVs and AVs in the New York City area. We use a hexagonal grid to discretize the map as depicted in Figure 4. There are 436 grids with an edge length of 0.46 kilometers covering areas with order records in New York, including Manhattan and two airports. The advantage of hexagonal discretization of the area is that all neighbor grids are equidistant for hexagons, simplifying vehicle movement analysis. The fleet size is 6,000, which is set by authors manually. The simulator models the dynamics of an MoD system, including the order generation, movement of drivers and AVs, and order assigning procedure. In this simulation environment, different grids may have different numbers of accessible neighbor grids as there are lakes or parks. To address this issue, we adopt the policy context embedding process proposed in Lin et al. (2018b) to recalculate the valid logits for both the operator agent and driver agents.

Figure 4. (Color online) Discretization of the Study Area



Next, we state the main activities in one time period (e.g., 10 minutes) in the simulator. First, new orders are generated by randomly sampling from the taxi trip data set to mimic unknown and stochastic demand that can be revealed along the time steps. The details of the data set and procedure for sampling random trip requests are given in Section 5.1.2. With order information that is revealed and aggregated in one period, the simulator assigns orders to vehicles according to the matching heuristic stated in Section 4.4. Vehicles matched with orders in the same grid pick up the passenger at the same period and transfer into the "serve" status. The status of vehicles matched with orders in the neighbor grids becomes "picking up." It takes one time period to pick up the passenger and then transfer into the serve status in the next period. Then, the spatial and temporal features to describe the system states are perceived in each grid. The operator (supervisor agent) decides the relocation plan of idle AVs and determines the value of CR taking effect in the next period, and each idle driver makes movement decisions based on their observed information. After the execution of decisions, the reward is collected (note that the reward collected at the current time period corresponds to the action taken in the last period) and the system states will be updated accordingly. The decision-making and experience collection procedures are given in Algorithm 1.

We also take account of the heterogeneity of passengers' maximum endurable waiting time. In our experiments, 80% customers' maximum waiting time is one time step (i.e., 10 minutes) and 20% of them have a maximum waiting time of two time steps (i.e., 20 minutes). This setting reflects that most of the passengers are impatient; if they are not served within one interval, they will cancel the request and abandon the system. Therefore, we assume that the maximum waiting time to follow a Bernoulli distribution (i.e., a discrete probability

distribution for a Bernoulli trial) with a success probability of 0.8. When each trip request is randomly generated from the NYC taxi data set, the maximum waiting time for the trip request is also randomly generated from the Bernoulli distribution. We calculate the trip profit as the trip fare minus the travel cost. We assume the travel cost equals the travel distance times the unit travel cost for both AVs and CVs. The Manhattan distance between the pickup and drop-off points of each trip is calculated and used as the travel distance. The unit travel cost c equals $0.6\$/km$ for both vehicles. The relocation cost for both AVs and CVs is estimated using the travel cost between the center of two neighbor grids, which is 0.92 kilometers in our hexagonal-grid network.

5.1.2. Sampling Trip Requests. We acquire a real-world taxi trip data set from the New York City Taxi & Limousine Commission (https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page). It is an open-source data set commonly used in MoD literature. We select one-month data of yellow taxis during April 2014. The trip data columns we used include pickup and drop-off dates and times, pickup and drop-off latitudes and longitudes, and trip fares. A data sample is presented in Table 1. The pickup and drop-off locations are mapped into our grid indexes from latitudes and longitudes.

The heat map of trip order locations is depicted in Figure 5. There is one hotspot near central park in the Manhattan area and two local hotspots at two airports. Figure 6 demonstrates the aggregated hourly departure trips on different days of the week. It can be observed that the real demand is highly asymmetric both spatially and temporally, and the demand pattern on weekends is slightly different from that on weekdays. We focus on weekdays, so the trip request data on weekdays is used in the experiments. After preprocessing the data and removing the trips outside our study area, a rich data set with 10 million weekday trip records is used for simulating stochastic trip requests and travel times. We divide the operational horizon into 10-minute intervals. We consider the operation period between 4 p.m. and 8 p.m., which includes 24 intervals. The period is selected to capture the congestion evolution and demand patterns that gradually transit from afternoon off-peak hours into evening peak hours. The average demand volume is around 90,000 trips per day during this period. Note that the trip requests in one day are generated by randomly sampling from the one-month data set by a fixed proportion during simulation. The sampling proportion is set to be 1/28 initially and is varied in Subsection 5.5 to analyze the system sensitivity to demand volume. The trip duration and trip fare of each order is also obtained from the sampled records. Ridesharing is not allowed in our system. We further assume that at the beginning of each operation horizon, the number of drivers and AVs at each grid in the service area is proportional to the grid's average demand volume. Given enough computational resources, our approach can be generalized to any network and any length of operation period. All the experiments are carried out on a computer with Intel Core i8086K CPU@4.00 GHz, 12 GB RAM, and NVIDIA Geforce RTX 2060 GPU.

5.1.3. Boundedly Rational Drivers. In the numerical experiments, we consider that drivers are not perfectly rational when they make decisions because of imperfect information and different cognitive ability. They are boundedly rational and may choose actions that are suboptimal. Here, we consider heterogeneous drivers in terms of their degree of bounded rationality (Di and Liu 2016, Di et al. 2017b, Li and Liu 2021). We assume that there exists a probability threshold that if the probability of some actions given by a policy exceeds the threshold (policy threshold), the boundedly rational drivers will be satisfied with those actions. We define an indifferent action set $\mathcal{A}_t^{m,-}$ for idle driver m at time t:

$$\mathcal{A}_t^{m,-} = \{a_{t,i}^m : \pi(a_{t,i}^m) \geq \zeta^m\},$$

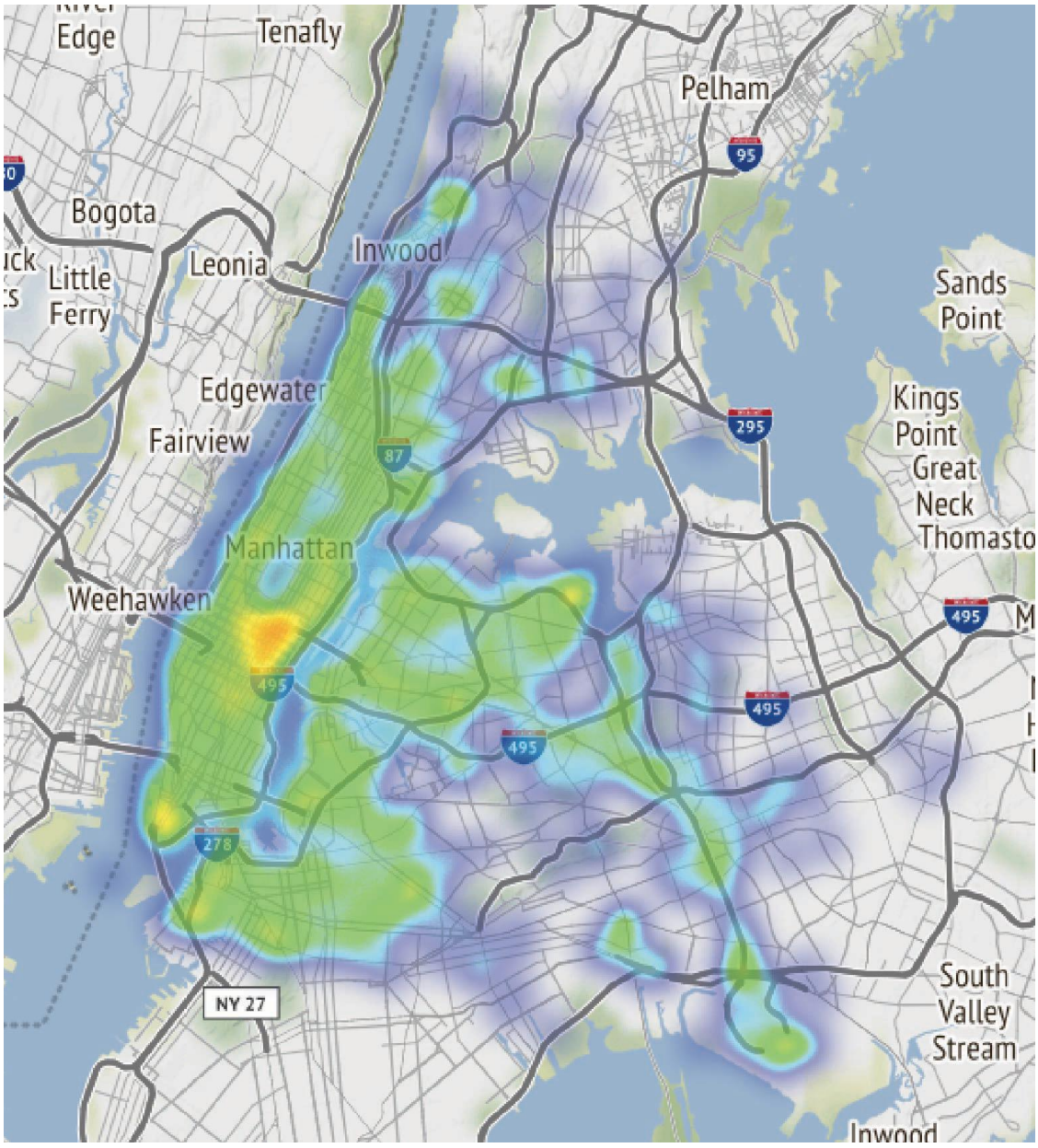
where ζ^m is the policy threshold corresponding to driver m. It is the set of actions with action probability that exceeds ζ^m. If the indifferent action set is not empty, the boundedly rational drivers will choose the next actions by selecting among the indifferent action set with equal probability. Otherwise, the driver agent will choose the action with the highest probability.

We also consider heterogeneous drivers by involving different types of drivers with different policy thresholds. To make insights sharp, we focus on the case where drivers are divided into two distinct classes, namely, high rationality (HR) and low rationality (LR). HR drivers have a higher policy threshold ζ_h, and LR drivers have a lower policy threshold ζ_l. We let $\zeta_h = 0.5$ and $\zeta_l = 0.4$ in our numerical experiments, and the ratio

Table 1. Taxi Trip Data Sample

Pickup time	Drop-off time	Pickup longitude	Pickup latitude	Drop-off longitude	Drop-off latitude	Fare
2014-04-08 08:59:39	2014-04-08 09:28:57	−73.958848	40.763585	−73.986284	40.752034	18.0
2014-04-08 08:00:20	2014-04-08 08:11:31	−73.973726	40.750095	−73.976889	40.755623	8.0

Figure 5. (Color online) Heat Map of Order Requests in New York



of LR drivers and HR drivers is 1:1. We assume that the supervisor agent has prior knowledge of the rationality type of each driver. To consider the drivers' heterogeneous characteristics, we add an indicator that specifies the type of driver in the observation tuple of each driver and modifies the state tuple of the supervisor agent:

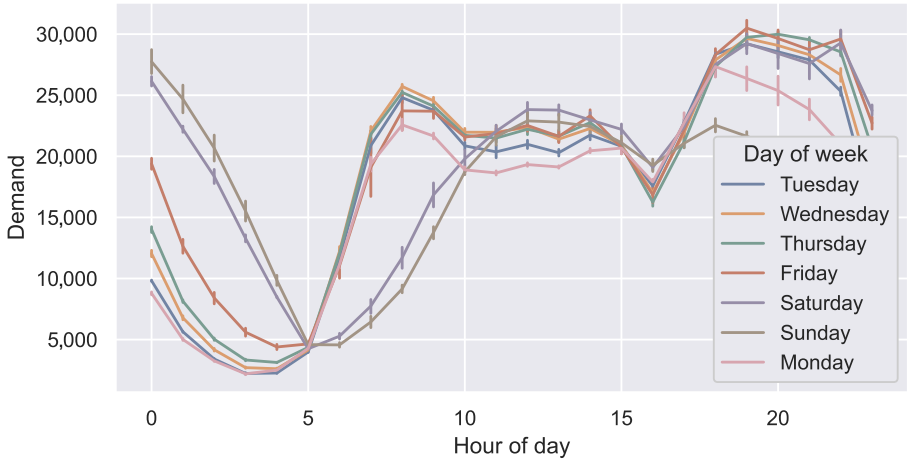
$$\mathcal{S} := \{(\boldsymbol{V}_t^a, \boldsymbol{V}_t^{c,h}, \boldsymbol{V}_t^{c,l}, \boldsymbol{D}_t, t) : t \in \mathcal{T}, \boldsymbol{D}_t \in \mathbb{R}_+^{N\times N}, \boldsymbol{V}_t^a \in \mathbb{R}_+^N, \boldsymbol{V}_t^c \in \mathbb{R}_+^N\}, \tag{19}$$

where $\boldsymbol{V}_t^{c,h}$ and $\boldsymbol{V}_t^{c,l}$ are the distribution of HR drivers and LR drivers, respectively.

5.2. Performance of the Two-Sided Multiagent DRL Algorithm

In this subsection, we carry out extensive numerical experiments to access the performance of our two-sided multiagent DRL algorithm. The following four policies, including three benchmark policies and our proposed policy, are tested in the simulation environment. (1) *Simulation policy*: The simulation benchmark adopts a multinomial logit model to describe drivers' relocation decisions. Following Wong, Szeto, and Wong (2014), we define $s_i = min(O_i/A_i, 1)$ as the probability of successfully being matched with an order for a vehicle at

Figure 6. (Color online) Number of Order Requests in New York



grid i, where O_i is the total recorded occupied trip originating from grid i and A_i is the total number of recorded idle drivers in grid i. Specifically, let $p_{ij}(j \in \mathcal{B}_i)$ denote the probability of a driver located at grid i selecting its neighbor grid j as a relocation target. The probability can be described in the following multinomial logit function:

$$p_{ij} = \frac{e^{\beta_1 s_j + \beta_2 RC_{ij}}}{\sum_{n \in \mathcal{B}_i} e^{\beta_1 s_n + \beta_2 RC_{in}}}, \ \forall i \in \mathcal{N}, \ \forall j \in \mathcal{B}_i, \tag{20}$$

where $RC_{ij} = u$ if $i \neq j$, else zero. We assume that the idle vehicles move one step for relocation at each time step, and the relocation decisions are independent of the previous decisions so that we do not consider the case with subsequent choices. In the numerical experiments, we set β_1 as 0.08, which is calibrated by Wong et al. (2014), and $\beta_2 = 0.1$. For the AVs, because they are fully cooperative and not intelligent, we assume that they are relocated to their neighbor grids where the supply number is greater than the number of requests. The probability of repositioning to a candidate neighbor grid is proportional to the demand-supply ratio of that grid. (2) *Multi-agent (MA) policy*: The two-sided multiagent DRL approach without charging dynamic CF scheme from drivers. (3) *Multi-agent-commission-fee-2 (MACF-2) policy*: In Section 4.2, we introduce the benchmark case with two separate policy networks, which give two types of policies. We name it as MACF-2 policy. (4) *Multi-agent-commission-fee-1 (MACF-1) policy*: Our two-sided multiagent DRL approach with dynamic CF scheme and the two-head policy network proposed in Section 4.

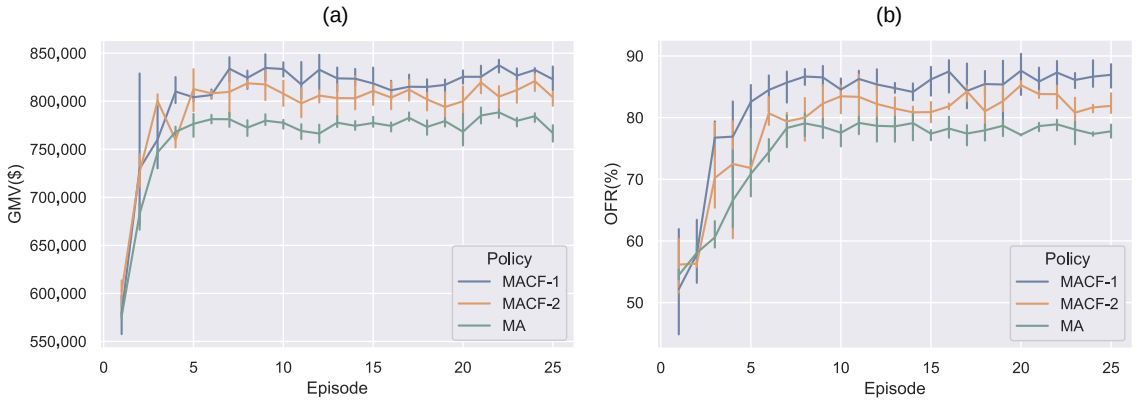
The general settings of our neural networks are the same throughout all the experiments. The policy network π_ϕ^A for the operator and the value networks Q_θ^C and Q_θ^A for drivers and the operator are all parameterized by a four-hidden-layer network with 512, 256, 128, and 32 nodes each layer. The policy network π_ϕ^C for drivers is a three-hidden-layer network of 256, 128, and 64 nodes. We use the ReLU (Li and Yuan 2017) activation function between hidden layers and transform the final output of driver policy network and operator policy network with ReLU and sigmoid function, respectively.

At the end of each episode, the neural networks are updated 4,000 times by bootstrapping from the replay buffer ($M1 = M2 = M3 = M4 = 4,000$). The batch size of one update is 256 over the first 10 episodes and is increased to 512 over the remaining episodes to stabilize the training further. An AdamOptimizer with a learning rate of $1e-3$ is used, the discount factor γ is 0.95, and β is 0.001. The value of ω in Equation (4) is set to be 10.

To simplify the discussion, we confine the demand level to be one (i.e., the mean value of sampled order in one episode is equal to that of the real data set) and the C-A ratio (ratio between the CV fleet size and the AV fleet size) to be 1:1, and the AV fleet size and CV fleet size are both 3,000 (i.e., the total fleet size is 6,000). We select the gross merchandise value (GMV) and order fulfillment rate as the evaluation metrics. GMV is calculated by adding up the total trip fare of served orders, and OFR is calculated by dividing the total number of fulfilled requests by the total number of requests during one episode. Higher GMV and OFR values ensure the income of drivers and platform as well as the system level of service, which helps attract more customers in the long run.

Figure 7 depicts the training curves of the MACF-1 policy, MACF-2 policy, and MA policy. The horizontal axis represents the number of training episodes, and the curves demonstrate the value of metrics (i.e., GMV and OFR) measured at the episode taking the average of seven runs. The vertical segments demonstrate the standard deviations of the results. To further show the training procedure of the two types of agents, we plot

Figure 7. (Color online) Training Curves of Three RL Approaches

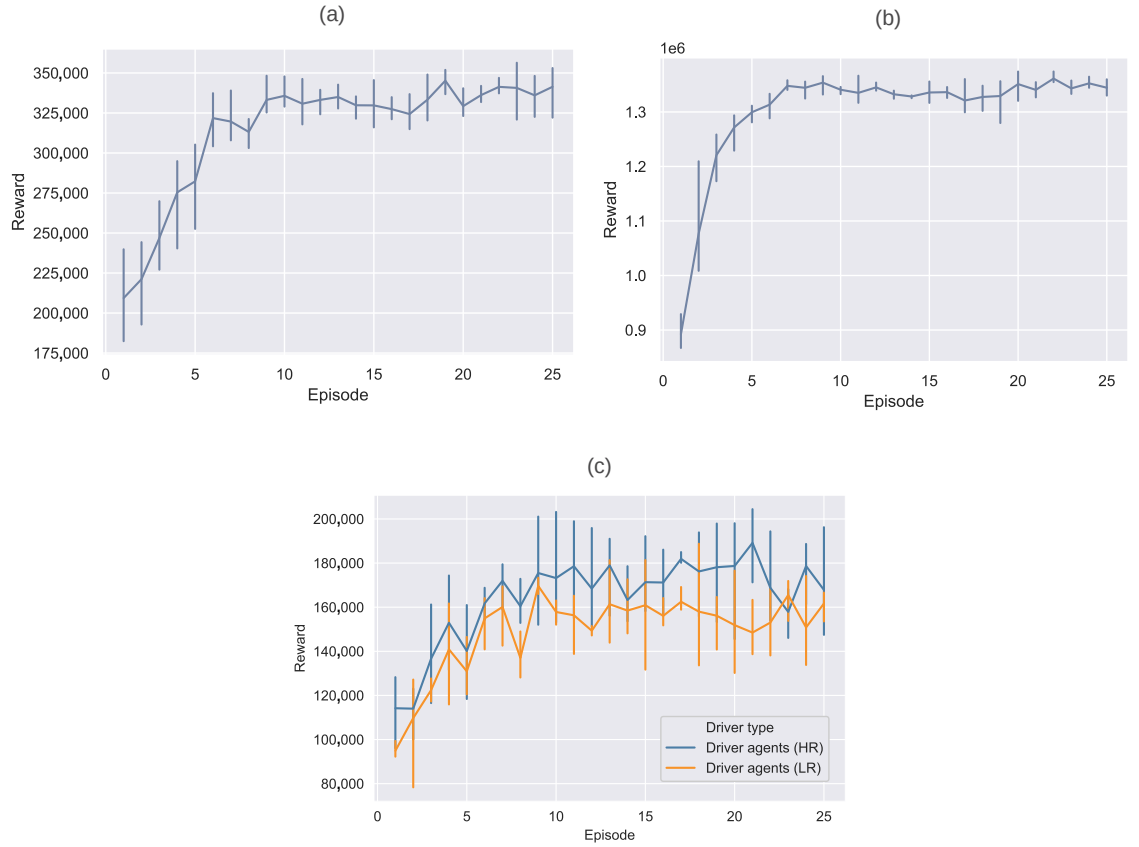


Note. (a) Gross merchandise value (GMV), (b) order fulfillment rate (OFR).

the total episode reward of driver agents and the supervisor agent in Figure 8(a) and (b), respectively. Both policies are improved significantly in the first five iterations when exploring the environment and gradually converge after seven episodes. Hence, the convergence of our algorithm is validated. The training curves of HR drivers and LR drivers are also presented in Figure 8(c), from which we observe that the reward obtained by HR drivers is slightly greater than that of LR drivers, which is not surprising because HR drivers tend to make decisions with better quality.

We then run the trained policy networks and simulation policy using the test demand data set while other settings remain the same as the training procedure. The testing results are close to the convergence values in Figure 7, demonstrating the ability of the trained policy

Figure 8. (Color online) Episode Reward During Training of Two Types of Agents and HR and LR Drivers



Note. (a) Episode reward of driver agents, (b) episode reward of the supervisor agent, (c) episode reward of HR and LR drivers.

to be seamlessly applied in online operations. The benchmark results under three policies are summarized in Table 2 by taking the average of 10 runs in each scenario. The performance metrics include GMV, driver earning, operator profit, average waiting time (AWT, i.e., the average waiting time until an order is matched), general OFR, average vehicle utilization rate (AUR), OFR and AUR for drivers (OFR-C and AUR-C) and AVs (OFR-A and AUR-C), and the averaged computational time of training and evaluation. Operator profit equals the AV profit plus the total CF. AUR measures the utilization rate of vehicles, and the utilization rate of one vehicle is the ratio of total time in service to the episode duration.

Next, we focus on the results from the scenarios when the C-A ratio equals 1 : 1. Comparing the performance of the four policies, three RL policies outperform the simulation policy significantly, whereas our MACF-1 policy achieves the best performance. The GMV and OFR of the MACF-1 policy are improved for 6.81% and 8.92% compared against the MA policy and 31.66% and 37.95% compared against the simulation policy. It reflects that the MACF-1 policy can learn from practical experience and relocate the idle AVs to areas suffering from supply shortage compared with the simulation policy with anticipation of future demand and drivers' behavior. Also, the AUR of the MACF-1 policy is 8.89% higher than that of the MA policy. It indicates that the MA policy is not enough when uncooperative human drivers exist in the system, and embedding dynamic CF is essential to affect the drivers' profit-seeking behavior to cover more trip requests and improve the system level of service.

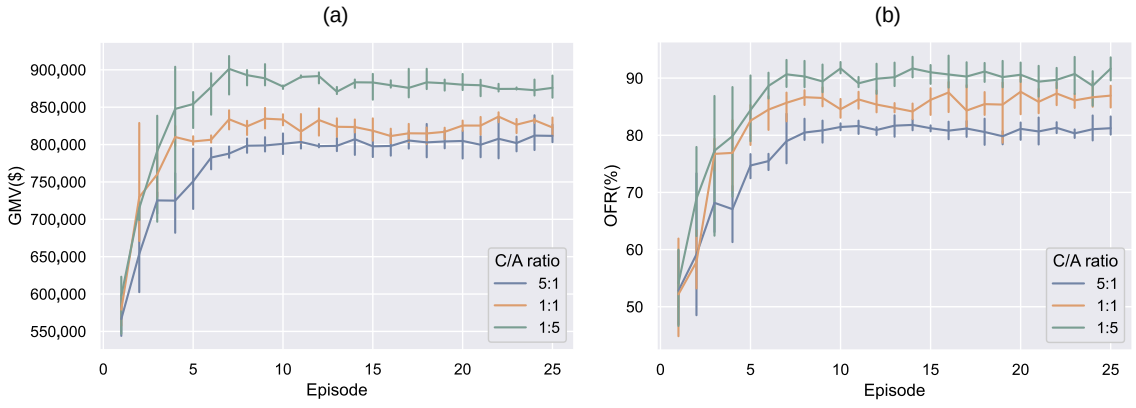
Comparing the performance of MACF-1 and MACF-2 provides insights into the benefit of the two-head policy network. The MACF-1 policy outperforms the MACF-2 policy in terms of several metrics. In the meanwhile, the MACF-1 policy requires less computational time. The two-head network structure enables a certain degree of information sharing within the shared network layers to make two types of decisions effectively and efficiently. The lightweight network structure also significantly reduces the number of weights in the neural networks and leads to higher learning efficiency.

The computational times for training and evaluation are also summarized in the last two rows in Table 2, which reveals that training takes much longer time than evaluation. In fact, the simulation and the neural network updating are the most time-consuming parts of the training process. The evaluation time is around 50 seconds per episode, which means the decision time at each step only takes a few seconds. It validates that, as the agent is well trained, our algorithm has the capability of generating decisions quickly and effectively leveraging real-time data. On the other hand, the supervisor agent could keep improving its strategy after a certain amount of experience is collected when the policy is carried out in real environments. With the online updates, our two-sided multiagent DRL approach has the capability to adapt to the uncertainties in practical applications. Therefore, our approach is highly transferable and adaptable in practical implementations.

5.3. Impact of AV Penetration Rate

This subsection examines the impact of the AV penetration rate in the mixed fleet. The system performances for two additional C-A ratios, that is, 5:1 and 1:5, are evaluated to analyze the system's sensitivity to various AV penetration levels. The demand rate and other training specifications are the same as in the previous section. Figure 9(a) and (b) compare the training curves of our MACF-1 policy of two performance metrics under three scenarios of the C-A ratio. The green, orange, and blue curves here correspond to the C-A ratio 1:5, 1:1, and 5:1, respectively. The figure shows that as the C-A ratio decreases, both the converged GMV and OFR increase.

Figure 9. (Color online) Training Curves of the MACF-1 Policy Under Various C-A Ratios



Note. (a) Gross merchandise value, (b) order fulfillment rate.

Table 2. Performance Benchmark and Computational Times of Three Policies Under Various C-A Ratios

Policy	MACF-1 policy			MACF-2 policy			MA policy			Simulation policy		
C-A ratio	5:1	1:1	1:5	5:1	1:1	1:5	5:1	1:1	1:5	5:1	1:1	1:5
GMV ($)	794,864.79	828,968.21	878,707.39	784,961.71	813,808.31	866,338.97	740,774.43	776,083.13	838,540.69	610,829.70	629,651.75	616,167.13
Driver earning ($)	533,258.20	347,521.29	111,062.47	538,081.05	346,510.84	129,250.47	488,383.44	316,461.33	102,242.14	439,321	271,790	118,004
Operator profit ($)	272,880.31	485,940.12	781,433.19	518,476.28	464,309.51	112,896.80	230,290.73	459,392.56	728,743.14	183,472.56	348,837.39	496,203.92
AWT (min)	8.08	7.39	6.64	7.82	7.32	7.69	8.23	8.06	7.86	9.37	8.44	8.66
OFR-C (%)	69.29	73.84	97.71	67.35	71.60	87.41	64.50	68.86	81.21	48.28	55.85	72.56
OFR-A (%)	67.81	70.33	73.70	63.76	69.42	73.93	62.03	65.36	66.55	45.09	48.66	43.53
OFR (%)	69.05	72.08	77.70	66.75	70.51	76.18	64.09	67.11	69.00	47.75	52.25	48.37
AUR-C (%)	85.49	88.84	93.21	82.15	84.60	104.41	81.70	84.86	87.21	62.88	72.85	83.57
AUR-A (%)	82.81	85.33	91.40	85.76	88.42	90.93	78.03	79.36	90.55	44.09	47.66	48.53
AUR (%)	85.05	87.08	91.70	82.75	86.51	93.18	81.09	82.11	90.00	59.75	60.25	54.37
Training time(h)	8.38	7.14	7.14	8.71	7.71	7.29	7.62	7.05	6.86	—	—	—
Evaluation time(s)	55.67	53.75	55.00	57.50	53.75	51.92	55.33	56.75	57.00	33.19	29.62	31.71

Note. AUR, average vehicle utilization rate; AWT, average waiting time.

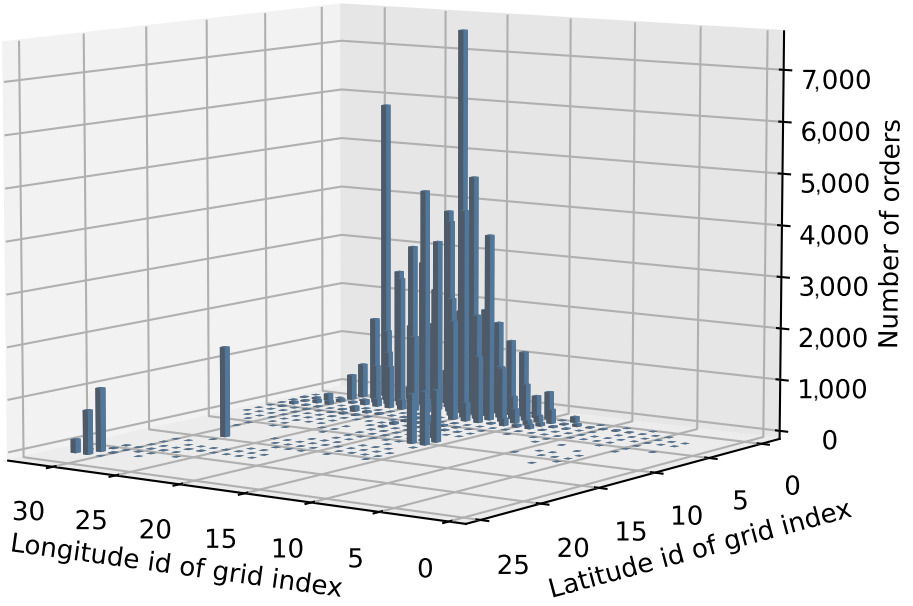
We further vary the C-A ratio and test with three benchmark policies. The test results are also summarized in Table 2. The results suggest that introducing AVs is beneficial for both passengers and the operator. Under our MACF-1 policy with a 1:5 C-A ratio (i.e., 1,000 CVs and 5,000 AVs), the system GMV and OFR increase 6.34% and 5.36%, respectively, compared with the 1:1 C-A ratio scenario. When the C-A ratio is 5:1 (i.e., 5,000 CVs and 1,000 AVs), system GMV and OFR decrease by 4.99% and 6.82%, respectively. Such an observation validates the advantage of fully cooperative AVs to fleet management in large-scale MoD systems. In the foreseeable future, as the AV penetration rate gradually increases in the market, a smaller fleet size will be needed to maintain a satisfying level of service.

Meanwhile, we observe higher AUR of drivers in scenarios with smaller CV fleet size, which can be explained by the less competitive environment for drivers when the total number of drivers reduces. The MACF-1 policy consistently achieves advantages across various metrics compared against the MA policy under different C-A ratios. An in-depth discussion on the benefit of dynamic CF will be given in the next subsection.

5.4. Benefit of Dynamic Commission Fee

In this subsection, we would like to demonstrate the effectiveness of our dynamic CF scheme combined with AV fleet management in addressing the unbalanced spatial distribution of demand and supply. For better illustration, we plot the aggregated demand at each grid in one episode where the demand rate is still one in Figure 10. To better present our method on alleviating the demand-supply gap, we plot the average demand-supply gap at each grid in three-dimensional bar plots in Figure 11. Grids with more demands than supplies have positive values and negative values otherwise. Figure 12(a) and (b) reflect the spatial demand-supply gap with high and low AV penetration, respectively; each of them presents the results under two policies, namely, with and without CF scheme. The blue bars show the gap under the MA policy, and the orange bars show the gap under the MACF-1 policy. In both subfigures, the demand-supply gap is narrowed when CF is incorporated. This is expected because CF discourages drivers from oversupplied grids and seeks riders in undersupplied grids, such as the center Manhattan area and two airports. Comparing two subfigures, the worst case is revealed to be the scenario with low AV penetration and without CF. Drivers gather in crowded areas so that many customers are left unserved because of vehicle shortages. However, the CF scheme and centralized AV reposition primarily mitigate the problem by sending vehicles to oversupplied areas. The major takeaway from Figure 11 is that the MACF-1 policy broadly fills the demand-supply gap in the metropolitan area during

Figure 10. (Color online) The Aggregated Demand at Each Grid in One Episode



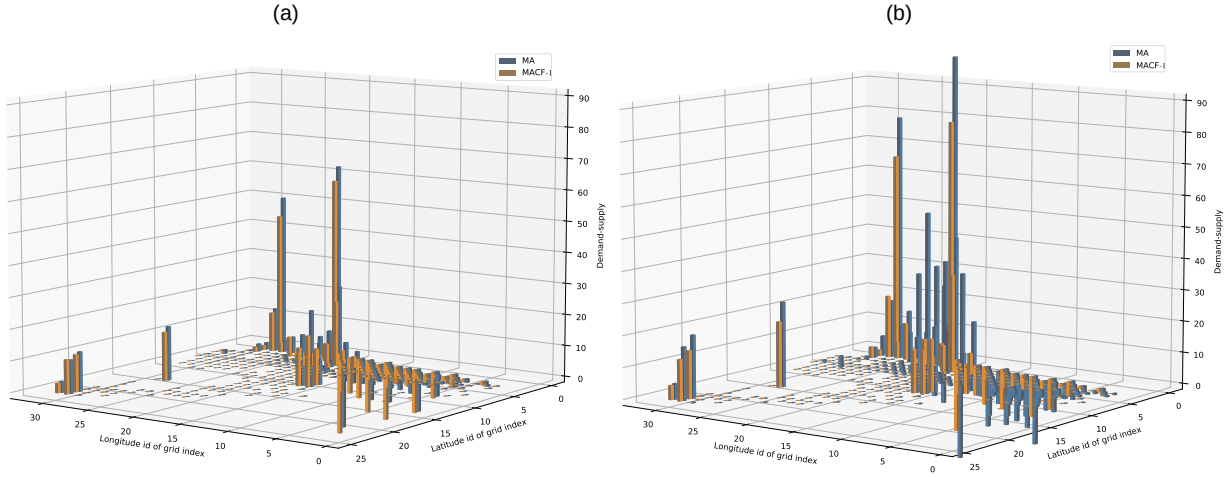
evening peaks, whereas drivers without being charged CF fail to take orders heading to undersupplied destinations, thus leaving many hotspots with unserved orders.

5.5. Sensitivity Analysis on Demand Level

Next, we conduct sensitivity analysis by varying the demand level in the simulation environment. To synthesize a lower-demand-density scenario, for each record in the one-month trip data set, we shrink the sampling rate to two-thirds, resulting in the mean demand volume equal to around 60,000 per episode. Similarly, we magnify the demand by one-third and two-thirds to simulate high-density scenarios, corresponding to mean episode demand volumes 120,000 and 15,000. All the tests are run using the trained policy networks under the scenario with the demand rate as one and the C-A ratio as 1:1. Throughout the tests in this subsection, we fix the C-A ratio to be 1:1 as well.

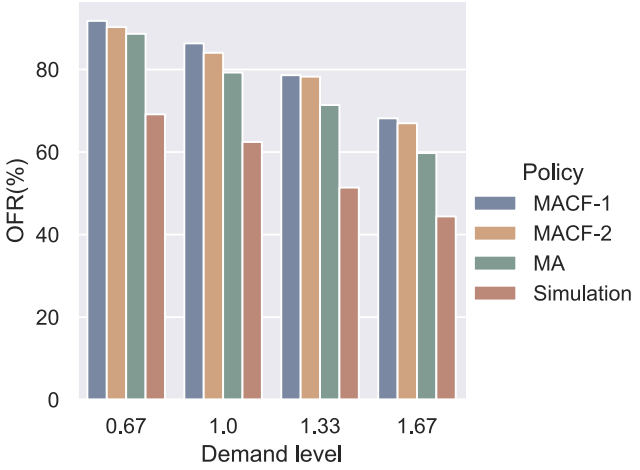
Figure 12 and 13 show the average episode OFR and GMV, respectively. The horizontal axis indicates the various demand rate compared with the base case. The superiority of MACF-1 in improving the OFR when demand is rather dense can be seen from the figures. Specifically, our MACF-1 policy exhibits the greatest advantage in GMV and OFR comparing to the benchmarks. The MACF-1 policy conquers the MACF-2 and

Figure 11. (Color online) Demand-Supply Gap Throughout the City During Evening Peak, Grid with More Demand Show Positive Values and Negative Otherwise



Note. (a) C-A ratio = 1:5, (b) C-A ratio = 5:1.

Figure 12. (Color online) Order Fulfillment Rate of Three Policies Under Various Demand Levels



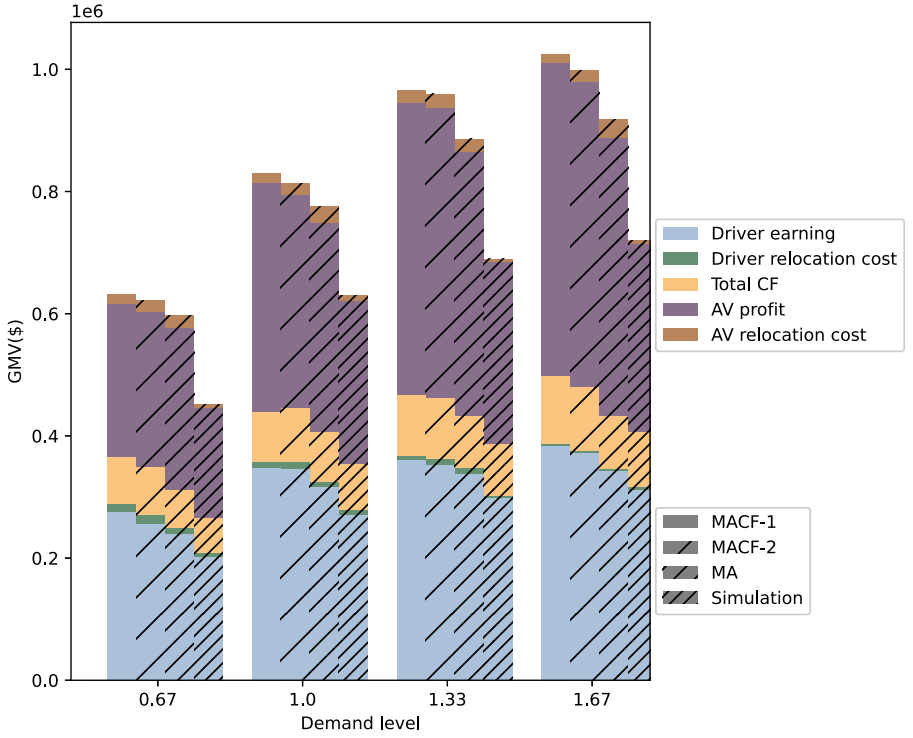
the MA policy, especially when the demand pattern is crowded. In addition, it maintains satisfactory OFR except for the extreme demand rate, for the reason we discussed in Section 5.4.

We further discuss the profits and the costs for AVs and CVs in Figure 13, including the profits and relocation costs of drivers and AVs and the amount of CF charged from drivers. The height of each bar is the GMV throughout the operation horizon. The profit of the platform consists of the profit earned by AVs and the CF charged from drivers. When the supply is sufficient in low-demand scenarios, the amount of AVs' profit is relatively low compared with drivers' earning, which is due to the priority given to drivers to retain drivers. When the demand level changes from 0.67 to 1.67, the total profits of AVs in the three policies increase rapidly because the driver fleet is exploited as demand increases. Regardless of the demand level, the dynamic CF scheme is beneficial in increasing the profit of both drivers and platforms compared with the other two benchmark policies, and the positive effect is more significant when the demand volume is extremely large. Also, the amounts of relocation cost for both the drivers and AVs are relatively low, indicating that the total relocation distance of vehicles is acceptable, because the relocation cost is proportional to the relocation distance.

6. Conclusion

This study investigates the online operation problem of an MoD system with a mixed fleet of AVs and CVs. We propose a two-sided multiagent DRL approach to capture different behavior and objectives of the operator and human drivers as well as to adapt fluidly to a dynamic environment with fluctuations in demands. Focusing on the supply side problem, the operator makes centralized decisions to relocate the AVs and commission rate aiming at improving the system performance, considering the reward-seeking behavior of

Figure 13. (Color online) The GMV Values Under Scenarios with Different Demand Volumes with Various Components



drivers. The numerical experiments in NYC validate the effectiveness of our approach under a mixed autonomy setting, where the demand is generated from a real-world data set. The trained policy can make decisions online in a rapid manner and easily tackle the system uncertainty. It is worth mentioning that the GMV and the OFR of the system are increased by 31.66% and 37.95%, respectively, compared with the simulation policy. Comparing the system performance under various AV penetration rates, we discover that introducing more AV fleet can significantly improve the system objective. The dynamic location-based CF scheme is also beneficial to both the drivers and the platform. The results also validate the advantage of the two-head policy network.

This study opens a door for several future directions. First, the study can be extended by investigating how the MoD system with a mixed fleet will impact traffic congestion. Second, with the advancement in technology, electrification has become an inevitable trend in future intelligent mobility systems. We can take the electric AV charging problem into account and develop an optimal charging strategy for the AVs under such a setting. Third, our paper uses dynamic CF to regulate drivers' relocation decisions. However, in practice, high spatiotemporal volatility of the trip price is often observed, leading to commission fees with high volatility. Explicitly providing incentives to drivers as long as they enter an area can tackle this issue, which is left for future study. Fourth, our four-step matching heuristic does not perfectly capture the matching procedure in practice. Therefore, integrating a more sophisticated matching algorithm into our proposed framework is another promising direction. Fifth, there is only one platform in our problem. Another possible direction is to consider multiple platforms. Each platform may operate a mixed fleet or a pure fleet, which could be autonomous or conventional. Comparing the effects of platforms' competition and cooperation is also an interesting topic. Finally, we focus on the supply side of the MoD system in this study. Another possible direction is to incorporate surge pricing or dynamic pricing to influence the demand side to balance the supply and demand. Because passengers may have different attitudes toward trip price and vehicle type, learning passengers' preferences is also an interesting topic.

References

Afeche P, Liu Z, Maglaras C (2018) Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance. Preprint, submitted February 12, https://dx.doi.org/10.2139/ssrn.3120544.

Ahmed S, Muhammod R, Khan ZH, Adilina S, Sharma A, Shatabda S, Dehzangi A (2021) Acp-mhcnn: An accurate multi-headed deep-convolutional neural network to predict anticancer peptides. *Sci. Rep.* 11(1):1–15.

Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc. Natl. Acad. Sci. USA* 114(3):462–467.

Arik SÖ, Jun H, Diamos G (2018) Fast spectrogram inversion using multi-head convolutional neural networks. *IEEE Signal Processing Lett.* 26(1):94–98.

Banerjee S, Johari R, Riquelme C (2015) Pricing in ride-sharing platforms: A queueing-theoretic approach. Roughgarden T, Feldman M, Schwarz M, eds. *Proc. Sixteenth ACM Conf. Econom. Comput.* (ACM, New York), 639–639.

Buşoniu L, Babuška R, De Schutter B (2010) Multi-agent reinforcement learning: An overview. Srinivasan D, Jain LC, eds. *Innovations in Multi-Agent Systems and Applications–1: Studies in Computational Intelligence* (Springer, Berlin, Heidelberg), 183–221.

Cachon GP, Daniels KM, Lobel R (2017) The role of surge pricing on a service platform with self-scheduling capacity. *Manufacturing Service Oper. Management* 19(3):368–384.

Castillo JC, Knoepfle D, Weyl G (2017) Surge pricing solves the wild goose chase. Daskalakis C, Babaioff M, Moulin H, eds. *Proc. 2017 ACM Conf. Econom. Comput.* (ACM, New York), 241–242.

Chen X, Di X (2021) Ridesharing user equilibrium with nodal matching cost and its implications for congestion tolling and platform pricing. *Transportation Res. Part C: Emerging Tech.* 129:103233.

Chen TD, Kockelman KM (2016) Management of a shared autonomous electric vehicle fleet: Implications of pricing schemes. *Transportation Res. Record* 2572(1):37–46.

Chen Y, Liu Y (2022) Integrated optimization of planning and operations for shared autonomous electric vehicle systems. *Transportation Sci.*, ePub ahead of print August 24, https://doi.org/10.1287/trsc.2022.1156.

Chen L, Mislove A, Wilson C (2015) Peeking beneath the hood of Uber. Cho K, Fukuda K, eds., Pai V, Spring N, Program Chairs, *Proc. 2015 Internet Measurement Conf.* (ACM, New York), 495–508.

Chen XM, Zheng H, Ke J, Yang H (2020a) Dynamic optimization strategies for on-demand ride services platform: Surge pricing, commission rate, and incentives. *Transportation Res. Part B: Methodological* 138:23–45.

Chen C, Wei H, Xu N, Zheng G, Yang M, Xiong Y, Xu K, Li Z (2020b) Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. *Proc. Conf. AAAI Artificial Intelligence* 34(4):3414–3421.

Chow Y, Yu JY, Pavone M (2015) Two phase Q-learning for bidding-based vehicle sharing. Preprint, submitted September 29, https://arxiv.org/abs/1509.08932.

Coppola P, Silvestri F (2019) Autonomous vehicles and future mobility solutions. Coppola P, Esztergar-Kiss D, eds. *Autonomous Vehicles and Future Mobility* (Elsevier, Amsterdam), 1–15.

Di X, Ban XJ (2019) A unified equilibrium framework of new shared mobility systems. *Transportation Res. Part B: Methodological* 129:50–78.

Di X, Liu HX (2016) Boundedly rational route choice behavior: A review of models and methodologies. *Transportation Res. Part B: Methodological* 85:142–179.

Di X, Liu HX, Ban XJ (2016) Second best toll pricing within the framework of bounded rationality. *Transportation Res. Part B: Methodological* 83:74–90.

Di X, Liu HX, Ban X, Yang H (2017) Ridesharing user equilibrium and its implications for high-occupancy toll lane pricing. *Transportation Res. Record* 2667(1):39–50.

Di X, Liu HX, Zhu S, Levinson DM (2017) Indifference bands for boundedly rational route switching. *Transportation* 44(5):1169–1194.

Duan L, Wei Y, Zhang J, Xia Y (2020) Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode. *Transportation Res. Part C: Emerging Tech.* 111:397–420.

Flet-Berliac Y, Preux P (2019) MERL: Multi-head reinforcement learning. Preprint, submitted September 26, https://arxiv.org/abs/1909.11939.

Furuhata M, Dessouky M, Ordóñez F, Brunet M-E, Wang X, Koenig S (2013) Ridesharing: The state-of-the-art and future directions. *Transportation Res. Part B: Methodological* 57:28–46.

Gao Y, Jiang D, Xu Y (2018) Optimize taxi driving strategies based on reinforcement learning. *Internat. J. Geographical Inform. Sci.* 32(8):1677–1696.

Godfrey GA, Powell WB (2002a) An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times. *Transportation Sci.* 36(1):21–39.

Godfrey GA, Powell WB (2002b) An adaptive dynamic programming algorithm for dynamic fleet management, ii: Multiperiod travel times. *Transportation Sci.* 36(1):40–54.

Guériau M, Dusparic I (2018) SAMoD: Shared autonomous mobility-on-demand using decentralized reinforcement learning. *21st Internat. Conf. Intelligent Transportation Systems* (IEEE, New York), 1558–1563.

Haliem M, Mani G, Aggarwal V, Bhargava B (2020) A distributed model-free ride-sharing approach for joint matching, pricing, and dispatching using deep reinforcement learning. Preprint, submitted October 5, https://arxiv.org/abs/2010.01755.

Haydari A, Yilmaz Y (2020) Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Trans. Intelligent Transportation Systems* 23(1):11–32.

He F, Wang X, Lin X, Tang X (2018) Pricing and penalty/compensation strategies of a taxi-hailing platform. *Transportation Res. Part C: Emerging Tech.* 86:263–279.

Hu M, Zhou Y (2020) Price, wage, and fixed commission in on-demand matching. Preprint, submitted September 1, https://dx.doi.org/10.2139/ssrn.2949513.

Hu S, Dessouky MM, Uhan NA, Vayanos P (2021) Cost-sharing mechanism design for ride-sharing. *Transportation Res. Part B: Methodological* 150:410–434.

Karamanis R, Angeloudis P, Sivakumar A, Stettler M (2018) Dynamic pricing in one-sided autonomous ride-sourcing markets. *21st IEEE Internat. Conf. Intelligent Transportation Systems* (IEEE, New York), 3645–3650.

Ke J, Xiao F, Yang H, Ye J (2019) Optimizing online matching for ride-sourcing services with multi-agent deep reinforcement learning. Preprint, submitted February 17, https://arxiv.org/abs/1902.06228.

Ke J, Yang H, Li X, Wang H, Ye J (2020) Pricing and equilibrium in on-demand ride-pooling markets. *Transportation Res. Part B: Methodological* 139:411–431.

Kim B, Kim J, Huh S, You S, Yang I (2020) Multi-objective predictive taxi dispatch via network flow optimization. *IEEE Access* 8:21437–21452.

Konda VR, Tsitsiklis JN (2000) Actor-critic algorithms. Solla S, Leen T, Muller K, eds. *Advances in Neural Information Processing Systems* (NIPS, Denver), 1008–1014.

Lei C, Jiang Z, Ouyang Y (2020) Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers. *Transportation Res. Part B: Methodological* 132:60–75.

Li Y (2017) Deep reinforcement learning: An overview. Preprint, submitted January 25, https://arxiv.org/abs/1701.07274v1.

Li Q, Liao F (2020) Incorporating vehicle self-relocations and traveler activity chains in a bi-level model of optimal deployment of shared autonomous vehicles. *Transportation Res. Part B: Methodological* 140:151–175.

Li Y, Liu Y (2021) Optimizing flexible one-to-two matching in ride-hailing systems with boundedly rational users. *Transp. Res. Part E: Logist. Transportation Rev.* 150:102329.

Li Y, Yuan Y (2017) Convergence analysis of two-layer neural networks with relu activation. Preprint, submitted May 28, https://arxiv.org/abs/1705.09886v1.

Li Y, Liu Y, Xie J (2020) A path-based equilibrium model for ridesharing matching. *Transportation Res. Part B: Methodological* 138:373–405.

Li M, Di X, Liu HX, Huang H-J (2020) A restricted path-based ride-sharing user equilibrium. *J. Intelligent Transportation Systems* 24(4):383–403.

Li M, Qin Z, Jiao Y, Yang Y, Wang J, Wang C, Wu G, Ye J (2019) Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning, *World Wide Web Conf.* (Association for Computing Machinery, New York), 983–994.

Lin K, Zhao R, Xu Z, Zhou J (2018a) Efficient collaborative multi-agent deep reinforcement learning for large-scale fleet management. Preprint, submitted February 18, https://arxiv.org/abs/1802.06444v1.

Lin K, Zhao R, Xu Z, Zhou J (2018b) Efficient large-scale fleet management via multi-agent deep reinforcement learning. Guo Y, Farooq F, eds. *Proc. 24th ACM SIGKDD Internat. Conf. Knowledge Discovery & Data Mining* (Association for Computing Machinery, New York), 1774–1783.

Litman T (2017) *Autonomous Vehicle Implementation Predictions* (Victoria Transport Policy Institute, Victoria, BC, Canada).

Liu Y, Li Y (2017) Pricing scheme design of ridesharing program in morning commute problem. *Transportation Res. Part C: Emerging Tech.* 79:156–177.

Liu Y, Xie J, Chen N (2022) Stochastic one-way carsharing systems with dynamic relocation incentives through preference learning. *Transportation Res. Part E: Logist. Transportation Rev.* 166:102884.

Lokhandwala M, Cai H (2018) Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC. *Transportation Res. Part C: Emerging Tech.* 97:45–60.

Luo Q, Saigal R (2017) Dynamic pricing for on-demand ride-sharing: A continuous approach. Preprint, submitted October 23, https://dx.doi.org/10.2139/ssrn.3056498.

Ma J, Xu M, Meng Q, Cheng L (2020) Ridesharing user equilibrium problem under OD-based surge pricing strategy. *Transportation Res. Part B: Methodological* 134:1–24.

Mao C, Liu Y, Shen Z-JM (2020) Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. *Transportation Res. Part C: Emerging Tech.* 115:102626.

Miao F, Han S, Lin S, Stankovic JA, Zhang D, Munir S, Huang H, He T, Pappas GJ (2016) Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach. *IEEE Trans. Automation Sci. Engrg.* 13(2):463–478.

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. *Internat. Conf. Machine Learning* (PMLR, New York), 1928–1937.

Mo D, Chen XM, Zhang J (2022) Modeling and managing mixed on-demand ride services of human-driven vehicles and autonomous vehicles. *Transportation Res. Part B: Methodological* 157:80–119.

Nazari M, Oroojlooy A, Snyder LV, Takáč M (2018) Reinforcement learning for solving the vehicle routing problem. Preprint, submitted February 12, https://arxiv.org/abs/1802.04240v1.

Noruzoliaee M, Zou B, Liu Y (2018) Roads in transition: Integrated modeling of a manufacturer-traveler-infrastructure system in a mixed autonomous/human driving environment. *Transportation Res. Part C: Emerging Tech.* 90:307–333.

Nourinejad M, Roorda MJ (2016) Agent based model for dynamic ridesharing. *Transportation Res. Part C: Emerging Tech.* 64:117–132.

Ordóñez F, Dessouky MM (2017) Dynamic ridesharing. Batta R, Peng J, eds. *Leading Developments from INFORMS Communities*, INFORMS Tutorials in Operations Research (INFORMS, Catonsville, MD), 212–236.

O'Keeffe K, Anklesaria S, Santi P, Ratti C (2021) Using reinforcement learning to minimize taxi idle times. *J. Intelligent Transportation Systems* 26(4):1–16.

Pakusch C, Meurer J, Tolmie P, Stevens G (2020) Traditional taxis vs automated taxis–Does the driver matter for millennials? *Travel Behav. Soc.* 21:214–225.

Pang J-S, Zhang M, Dessouky MM, Gu W, Center MT, Region PS (2020) Modeling e-hailing and car-pooling services in a coupled morning-evening commute framework, Technical report, California Department of Transportation, Division of Research and Innovation, Sacramento, CA.

Powell WB (2007) *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 703 (John Wiley & Sons, New York).

Powell WB (2009) What you should know about approximate dynamic programming. *Naval Res. Logist.* 56(3):239–249.

Qin Z, Tang J, Ye J (2019) Deep reinforcement learning with applications in transportation. Teredesai A, Kumar V, Li Y, Rosales R, Terzi E, Karypis G eds. *Proc. 25th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 3201–3202.

Qin Z, Tang X, Jiao Y, Zhang F, Xu Z, Zhu H, Ye J (2020) Ride-hailing order dispatching at didi via reinforcement learning. *INFORMS J. Appl. Analytics* 50(5):272–286.

Ramezani M, Nourinejad M (2018) Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach. *Transportation Res. Part C: Emerging Tech.* 94:203–219.

Sayarshad HR, Chow JYJ (2017) Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. *Transportation Res. Part E: Logist. Transportation Rev.* 106:60–77.

Shetty S (2020) Uber's self-driving cars are a key to its path to profitability. Accessed November 1, 2022, https://www.cnbc.com/2020/01/28/ubers-self-driving-cars-are-a-key-to-its-path-to-profitability.html

Shojaeighadikolaei A, Ghasemi A, Jones KR, Bardas AG, Hashemi M, Ahmadi R (2020) Demand responsive dynamic pricing framework for prosumer dominated microgrids using multiagent reinforcement learning. Preprint, submitted September 23, https://arxiv.org/abs/2009.10890.

Shou Z, Di X (2020) Reward design for driver repositioning using multi-agent reinforcement learning. *Transportation Res. Part C: Emerging Tech.* 119:102738.

Shou Z, Di X, Ye J, Zhu H, Hampshire R (2019) Where to find next passengers on e-hailing platforms? A Markov decision process approach. Preprint, submitted May 23, https://arxiv.org/abs/1905.09906v1.

Simao HP, Day J, George AP, Gifford T, Nienow J, Powell WB (2009) An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Sci.* 43(2):178–197.

Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).

Tang X, Li M, Lin X, He F (2020) Online operations of automated electric taxi fleets: An advisor-student reinforcement learning framework. *Transportation Res. Part C: Emerging Tech.* 121:102844.

Torreño A, Onaindia E, Komenda A, Štolba M (2017) Cooperative multi-agent planning: A survey. *ACM Comput. Surveys* 50(6)84:1–84:32.

Turan B, Pedarsani R, Alizadeh M (2020) Dynamic pricing and fleet management for electric autonomous mobility on demand systems. Preprint, submitted October 1, https://arxiv.org/abs/1909.06962.

Ulmer MW, Goodson JC, Mattfeld DC, Hennig M (2019) Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Sci.* 53(1):185–202.

Van Seijen H, Fatemi M, Romoff J, Laroche R, Barnes T, Tsang J (2017) Hybrid reward architecture for reinforcement learning. Submitted June 13, https://arxiv.org/abs/1706.04208v1.

Vinsensius A, Wang Y, Chew EP, Lee LH (2020) Dynamic incentive mechanism for delivery slot management in e-commerce attended home delivery. *Transportation Sci.* 54(3):567–587.

Vosooghi R, Puchinger J, Jankovic M, Vouillon A (2019) Shared autonomous vehicle simulation and service design. *Transportation Res. Part C: Emerging Tech.* 107:15–33.

Wang X, He F, Yang H, Gao HO (2016) Pricing strategies for a taxi-hailing platform. *Transportation Res. Part E: Logist. Transportation Rev.* 93:212–231.

Wang X, Liu W, Yang H, Wang D, Ye J (2020a) Customer behavioural modelling of order cancellation in coupled ride-sourcing and taxi markets. *Transportation Res. Part B: Methodological* 132: 358–378.

Wang C, Zhang J, Xu L, Li L, Ran B (2019) A new solution for freeway congestion: Cooperative speed limit control using distributed reinforcement learning. *IEEE Access* 7:41947–41957.

Wang E, Ding R, Yang Z, Jin H, Miao C, Su L, Zhang F, Qiao C, Wang X (2020b) Joint charging and relocation recommendation for e-taxi drivers via multi-agent mean field hierarchical reinforcement learning. *IEEE Trans. Mobile Comput.* 21(4): 1274–1290

Wei Q, Rodriguez JA, Pedarsani R, Coogan S (2019) Ride-sharing networks with mixed autonomy. *2019 Amer. Control Conf.* (IEEE, New York), 3303–3308.

Wollenstein-Betech S, Paschalidis IC, Cassandras CG (2020) Joint pricing and rebalancing of autonomous mobility-on-demand systems. *59th IEEE Conf. Decision Control (CDC)* (IEEE, New York), 2573–2578.

Wong R, Szeto W, Wong S (2014) A cell-based logit-opportunity taxi customer-search model. *Transportation Res. Part C: Emerging Tech.* 48:84–96.

Xie J, Yang Z, Lai X, Liu Y, Yang XB, Teng T-H, Tham C-K (2022) Deep reinforcement learning for dynamic incident-responsive traffic information dissemination. *Transportation Res., Part E: Logist. Transportation Rev.* 166:102871.

Xu H, Pang J-S, Ordóñez F, Dessouky M (2015) Complementarity models for traffic equilibrium with ridesharing. *Transportation Res. Part B: Methodological* 81:161–182.

Xu Z, Li Z, Guan Q, Zhang D, Li Q, Nan J, Liu C, Bian W, Ye J (2018) Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. Guo Y, Farooq F, eds. *Proc. 24th ACM SIGKDD Internat. Conf. Knowledge Discovery & Data Mining* (ACM, New York), 905–913.

Yang H, Leung CW, Wong SC, Bell MG (2010) Equilibria of bilateral taxi–customer searching and meeting on networks. *Transportation Res. Part B: Methodological* 44(8–9):1067–1083.

Yang Z, Merrick KE, Abbass HA, Jin L (2017) Multi-task deep reinforcement learning for continuous action control. *IJCAI (U. S.)* 17:3301–3307.

Yang H, Shao C, Wang H, Ye J (2020a) Integrated reward scheme and surge pricing in a ride sourcing market. *Transportation Res. Part B: Methodological* 134:126–142.

Yang K, Tsao MW, Xu X, Pavone M (2021) Real-time control of mixed fleets in mobility-on-demand systems, *2021 IEEE Internat. Intelligent Transportation Systems Conf. (ITSC)* (IEEE, New York), 3570–3577.

Yang Y, Wang X, Yuanbo X, Huang Q (2020b) Multiagent reinforcement earning-based taxi predispatching model to balance taxi supply and demand. *J. Adv. Transportation* 2020:1–12.

Yang Y, Luo R, Li M, Zhou M, Zhang W, Wang J (2018) Mean field multi-agent reinforcement learning. Preprint, February 15, https://arxiv.org/abs/1802.05438v1.

Zha L, Yin Y, Du Y (2017) Surge pricing and labor supply in the ride-sourcing market. *Transportation Res. Procedia* 23:2–21.

Zha L, Yin Y, Du Y (2018) Surge pricing and labor supply in the ride-sourcing market. *Transportation Res. Part B: Methodological* 117:708–722.

Zha L, Yin Y, Xu Z (2018) Geometric matching and spatial pricing in ride-sourcing markets. *Transportation Res. Part C: Emerging Tech.* 92:58–75.

Zha L, Yin Y, Yang H (2016) Economic analysis of ride-sourcing markets. *Transportation Res. Part C: Emerging Tech.* 71:249–266.

Zhang R, Pavone M (2016) Control of robotic mobility-on-demand systems: A queueing-theoretical perspective. *Internat. J. Robotics Res.* 35(1–3):186–203.

Zhang D, Liu Y, He S (2019) Vehicle assignment and relays for one-way electric car-sharing systems. *Transportation Res. Part B: Methodological* 120:125–146.

Zhang C, Odonkor P, Zheng S, Khorasgani H, Serita S, Gupta C (2020) Dynamic dispatching for large-scale heterogeneous fleet via multi-agent deep reinforcement learning. Preprint, submitted August 24, https://arxiv.org/abs/2008.10713.

Zhu Z, Ke J, Wang H (2021) A mean-field Markov decision process model for spatial-temporal subsidies in ride-sourcing markets. *Transportation Res. Part B: Methodological* 150:540–565.