An adaptive agent-based approach for instant delivery order dispatching
Incorporating task buffering and dynamic batching strategies

Lu, Miaojia; Yan, Xinyu; Azadeh, Shadi Sharif; Wang, Pengling

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Research Paper

An adaptive agent-based approach for instant delivery order dispatching: Incorporating task buffering and dynamic batching strategies

Miaojia Lu [a,b], Xinyu Yan [c], Shadi Sharif Azadeh [d], Pengling Wang [a,b,*]

[a] *College of Transportation Engineering, Tongji University, China*
[b] *The Key Laboratory of Road and Traffic Engineering, Ministry of Education, 4800 Cao'an Road, Shanghai 201804, China*
[c] *Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, 11 Yuk Choi Rd, Hung Hom, Kowloon 999077, Hong Kong, China*
[d] *Transport & Planning Department, Civil Engineering and Geosciences, Delft University of Technology, Mekelweg 5, Delft, South Holland 2628, Netherlands*

ABSTRACT

The volume of instant delivery has witnessed a significant growth in recent years. Given the involvement of numerous heterogeneous stakeholders, instant delivery operations are inherently characterized by dynamics and uncertainties. This study introduces two order dispatching strategies, namely task buffering and dynamic batching, as potential solutions to address these challenges. The task buffering strategy aims to optimize the assignment timing of orders to couriers, thereby mitigating demand uncertainties. On the other hand, the dynamic batching strategy focuses on alleviating delivery pressure by assigning orders to couriers based on their residual capacity and extra delivery distances. To model the instant delivery problem and evaluate the performances of order dispatching strategies, Adaptive Agent-Based Order Dispatching (ABOD) approach is developed, which combines agent-based modelling, deep reinforcement learning, and the Kuhn-Munkres algorithm. The ABOD effectively captures the system's uncertainties and heterogeneity, facilitating stakeholders learning in novel scenarios and enabling adaptive task buffering and dynamic batching decision-makings. The efficacy of the ABOD approach is verified through both synthetic and real-world case studies. Experimental results demonstrate that implementing the ABOD approach can lead to a significant increase in customer satisfaction, up to 275.42%, while simultaneously reducing the delivery distance by 11.38% compared to baseline policies. Additionally, the ABOD approach exhibits the ability to adaptively adjust buffering times to maintain high levels of customer satisfaction across various demand scenarios. As a result, this approach offers valuable support to logistics providers in making informed decisions regarding order dispatching in instant delivery operations.

© 2023 Tongji University and Tongji University Press. Publishing Services by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Instant delivery services provide on-demand delivery within two hours by delivering goods like fresh products, takeout, and urgent documents via a digital platform (Dablanc et al., 2017). The volume of instant delivery has grown at a fast pace in recent years. The market size of the Chinese fresh e-commerce industry in 2021 has reached 46.4 billion dollars, 18.2% higher than in 2020 (iMedia Research, 2022).

However, the instant delivery industry has encountered various challenges that impede its expansion. Firstly, the exponential growth in instant delivery demand has led to increased complexity and difficulty in addressing order dispatching issues. The instant delivery platform possesses extensive data on interactions among stakeholders, including operators, customers, and couriers. Effectively utilizing and transforming this information into decision-making criteria is crucial for efficient order dispatching. Furthermore, order dispatching in instant delivery system is inherently characterized by uncertainties. Instances of such abrupt incidents encompass a sudden surge in order demand or a rider encountering a traffic accident. An unfavourable outcome of these unforeseen events is that, while the allocation of orders to couriers may be judicious at the time of assignment, the evolving statuses of couriers and orders over time diminish the optimality of these assignments. Consequently, assigning orders to riders whose suitability has waned can lead to order delays, necessitating additional waiting time for riders to fulfil orders. This, in turn, adversely affects delivery efficiency and impedes the enhancement of user experience.

Previous studies have dedicated substantial efforts to represent the complexity of instant delivery and address the inherent uncertainty in order dispatching. In contrast to the previous studies where the instant delivery problem was proposed as a mathematical programming model (Liu et al., 2018, Du et al., 2019, Zhen et al., 2023), we use Agent-based modelling (ABM) to describe the complex order dispatching problem. As ABM's flexibility and effectiveness of design in scenarios where interactions among the actors of a system are complex, stochastic, dynamic, and heterogeneous, as well as when agents' position in space plays a crucial role (Zhang et al., 2015). The utilization of ABM has gained widespread recognition as a suitable technique for modelling intricate systems, such as urban logistics, by addressing problems at a microscopic level (Hofmann et al., 2017, Fikar et al., 2018, Poeting et al., 2019). The use of ABM in the previous studies facilitated simulations that integrated complex behaviour rules, which can incorporate optimization algorithms to agents (Bonabeau, 2002) and support dynamic decision-making based on real-time information tracking (Turhanlar et al., 2022).

To address the uncertainties inherent in instant delivery, the majority of previous studies model the order dispatching process as a Markov Decision Process (MDP). An MDP is employed to handle decision-making challenges characterized by sequential interactions over discrete time steps between an agent and an environment. The conventional modelling methods and algorithms are inadequate in addressing the dynamic nature of the instant delivery problem (Holler et al., 2019, Kuhnle et al., 2019). Deep reinforcement learning (DRL) learning policies that optimize long-term rewards is more adept to solve MDP problems. It has been demonstrated that DRL outperforms the state-of-the-art order assignment algorithms to solve the order dispatching problem in logistics (Kuhnle et al., 2019, Voccia et al., 2019, Malus and Kozjek, 2020, Mo and Ohmori, 2021, Zou et al., 2021, Chen et al., 2022, Jahanshahi et al., 2022, Kavuk et al., 2022, Bozanta et al., 2022, Shi et al., 2022).

Among existing order dispatching research, most of them applied DRL for decision making such as routing planning and order selection or rejection, there are few studies that specifically employs DRL to devise strategies like task buffering in order dispatching. And few researchers have discussed the adaptive performances of DRL in different scenarios. Intelligent dispatching systems, designed to learn from historical data and dynamically adapt to evolving conditions, represent an important development direction for the future instant delivery studies (Liao et al., 2020). In this study we proposed two order dispatching strategies: task buffering and dynamic batching. Task buffering refers to optimizing the assignment time of orders to couriers for delivery, which means seeking the best assignment time to improve efficiency under the dynamic and uncertain environment. Based on our best knowledge, there is only one studies considering task buffering in the crowd sourcing dynamic pickup and delivery problem (Mo and Ohmori, 2021), but it assumed that the drivers delivered only one task per route. Dynamic batching refers to the system allocate the order to the optimum couriers based on the real-time information such as the delivery distance of orders with the same route and the distance between the courier's current location and the depot. The strategy of dynamic batching not only can find the optimum matching between couriers and orders, but also alleviate the delivery pressure of the couriers when there is a large amount of accumulated orders. As for the order batching studies, Li et al. (2022) combined normal meal orders by the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm and optimized the meal delivery problem at the next stage. Thus, few studies optimize the order batching problem considering the routing planning of the delivery vehicles.

The aim of this study is to develop an adaptive Agent-Based Order Dispatching (ABOD) approach, capable of capturing system uncertainties and heterogeneity, thereby facilitating adaptive decision-making. In contrast to prior research on order dispatching, this work makes the following contributions: 1) Using ABM formulating the model: Instant delivery order dispatching poses challenges due to its complex spatiotemporal dynamics. The performances of order dispatching are influenced by individual stakeholders' behaviours and their interactions (e.g., order placement, order allocation, and order pickup and delivery), which cannot be effectively addressed using analytical methods. In this research, we employ ABM instead of mathematical models such as mixed-integer linear programming to describe the order dispatching problem. ABM enables the incorporation of dynamics and interactions, allowing agents to make adaptive decisions based on real-time information; 2) Innovative order dispatching strategies: We introduce two innovative strategies, namely task buffering

and dynamic batching, to mitigate delivery pressure and address high uncertainties. These strategies, which are seldom discussed in existing instant delivery studies, offer novel approaches to improving the order dispatching performances; 3) highlighting ABOD's adaptive efficacy: We investigate the relationships between order time, buffering time, and customer satisfaction at the individual customer level. This analysis highlights the adaptivity of the ABOD approach across diverse demand and fleet sizes. To the best of our knowledge, this aspect has not been explored in previous studies on instant delivery problems, making it of significant practical importance.

The rest of the paper is organized as follows. Previous studies related to order dispatching are presented and discussed in section 2. Section 3 describes the order dispatching problem. The methodology incorporating task buffering and dynamic batching is detailed in Section 4. Section 5 provides synthetic and real-world case studies, with different policies simulated and compared. Finally, Section 6 concludes the paper.

2. Related work

Here we review recent research examining the order dispatching problem, including order dispatching in logistics and order dispatching in online ride-hailing. The methods used in the previous studies investigating order dispatching mainly focus on reinforcement learning. Remarkable progress has been observed in the application of reinforcement learning algorithms to address the complexities and uncertainties associated with order dispatch in logistics and online ride-hailing.

2.1. Order dispatching in logistics

Kuhnle et al. (2019) studied adaptive order dispatching in job shop manufacturing systems based on a reinforcement learning approach. An artificial neural network was embedded in reinforcement learning, and the reward value was taken as the weight of the neural network to optimize the order dispatching process. The results showed that the method can optimize overall performance in terms of both machine utilization and delivery time. Voccia et al. (2019) presented a formal MDP model for the same-day delivery problem and demonstrated that how uncertain future requests are modelled has the most impact on solution quality. Ulmer et al. (2019) integrated dynamic requests into same-day delivery routes with consideration of preemptive depot returns. The current value of a subset selection decision and its impacts on future rewards were quantified based on approximate dynamic programming. Malus and Kozjek (2020) adopted a multi-agent reinforcement learning approach to schedule the material flow in a production system. Autonomous mobile robots were agents taking actions based on their individual observations of the environment. The presented model had a better performance compared to the model based on the closest-first rule. Chen et al. (2022) studied the same-day delivery problem with fleets of drones and vehicles and proposed a deep Q-learning approach to address the order dispatch problem and compute the values for combinations of state and action features. Jahanshahi et al. (2022) tailored DRL algorithms to solve the meal delivery problem considering the effect of order rejection and courier repositioning. Mo and Ohmori (2021) studied the crowd sourcing dynamic pickup and delivery problem with consideration of task buffering and driver rejection and used multi-agent reinforcement learning to solve it. The actions of the task agent consist of assignment and waiting for a driver, but it assumed that the drivers delivered only one task per route. Kavuk et al. (2022) applied DRL to the order dispatching problem in ultrafast delivery service. The centralized warehouses in the regions decide whether an incoming order should be served or cancelled depending on their couriers' shifts and status. Two deep Q-networks were designed with two different rewards, and the results outperformed the rule-based heuristic employed in practice. Zou et al. (2021) proposed a Double Deep Q Network based reinforcement learning framework for O2O order dispatching using ABM, with different state encoding schemes designed and tested to improve the performance of the Double-DQN based dispatcher. Kronmueller et al. (2021) focused on the multi-depot vehicle routing problem and allowed robots to perform depot returns prior to being empty. Bozanta et al. (2022) applied the Double Deep Q-Networks to solve the courier routing and assignment problems of food delivery service with the objective of maximize the total expected reward. They have "reject" action in the action space. Wu et al. (2021) combined Floyd's algorithm and the particle swarm optimization algorithm for the task assignment and path planning of unmanned ground vehicles (UGVs). They proposed a distributed logistic controller that enables UGVs to achieve the objectives of minimizing the maximum time needed to complete all tasks. Guo et al. (2021) introduced a Time-Constrained Actor-Critic Reinforcement Learning-based concurrent dispatch system with the aim of augmenting long-term cumulative revenue while mitigating the overdue rate. A time-constrained action pruning module and a Deep Matching Network with a variable action space were designed to improve the DRL performances in order dispatching. Huang et al. (2023) presented a fleet management approach for the Green Logistic System, employing deep reinforcement learning. The proposed method facilitated integrated decision-making for order dispatching, route selection, and charging, with the overarching goal of optimizing operational profits. Spatial and temporal variations in charging costs were explicitly considered in the optimization process.

2.2. Order dispatching in online ride-hailing

Tong et al. (2017) defined the "Flexible Two-side Online task Assignment problem" and applied a two-step framework integrating offline prediction and online task assignment to solve it. The method was validated via experiments on both syn-

thetic datasets and real-world datasets from a large-scale taxi-calling platform. Xu et al. (2018) proposed an online dispatching model with a learning and planning approach. Based on real-time order data, a state-action value function was constructed through reinforcement learning, and the global optimal matching was solved through combinatorial optimization. Tang et al. (2019) incorporated a temporal factor into the state-action value function, and further adopted deep neural networks, specifically Cerebellar Value Networks, to learn the matching value of future orders. Holler et al. (2019) presented a DRL approach for tackling full fleet management and dispatching problems. The researchers treated the drivers as individual agents and considered the problems from driver-centric and system centric perspectives. Yao et al. (2020) tested the hybrid operations of human driving vehicles and automated vehicles using a data-driven multi-agent simulation platform and applied an Kuhn-Munkres (KM) algorithm to find the optimum match between passengers and vehicles. Lv et al. (2021) employed random forest classification to distinguish commuting private cars from other travel vehicles and identified ride-sharing opportunities among commuting private cars using reinforcement learning. These methods resulted in an approximately 21% reduction in the number of commuting private cars during both morning and evening peak hours following ride-sharing matches. Wang and Guo (2021) modelled the dispatching problem of shared autonomous electric vehicles according to a MDP and two optimization models from short-sighted view and farsighted view based on combinatorial optimization theory, respectively. The recharging and repositioning processes were also taken into consideration. Ge et al. (2021) proposed a traffic assignment framework for optimal matching and routing of shared autonomous vehicle (SAV) trips by considering the congestion effect of SAV operations in a mixed traffic environment. Tong et al. (2021) proposed an approach for large-scale taxi order dispatching that allows synergic integration of reinforcement learning and combinatorial optimization. Noruzoliaee and Zou (2022) incorporated the ride sharing and network equilibrium into addressing the autonomous ridesharing order dispatching problem. Liu et al. (2022) proposed a DRL approach for vehicle dispatching through an online ride-hailing platform based on industrial-scale real-world data. The vacant vehicles are reallocated to regions with large demand gaps in advance and the problem of high concurrency of dispatching requests is addressed by sorting the actions as a recommendation list. Wang et al. (2023) introduced the Courier Displacement Reinforcement Learning (CDRL) framework, based on centralized multi-agent actor-critic, addressing challenges specific to cross-region courier displacement in on-demand delivery, resulting in a notable improvement of 47.97% in supply–demand balance. Yan et al. (2023) addressed the charging and order scheduling issue for an online hailing vehicle fleet, modelled it as a Markov decision process, and introduced a novel online approximation algorithm to optimize platform profits in a dynamic stochastic environment.

3. Problem description

In this section, we give formal definitions relevant to the heterogenous stakeholders and formulate the order dispatching problem in a dynamic environment.

3.1. Definitions

Definition 1. Instant delivery: Instant delivery services provide on-demand delivery within two hours by connecting delivery depot, couriers, and customers via a digital platform (Dablanc et al., 2017). Goods typically provided via instant delivery include but are not limited to fresh products, takeout, and urgent documents. In this study, the instant delivery we discuss refers to the online fresh food delivery.

Definition 2. Couriers: The couriers discussed in this study are self-logistics of the fresh food e-commerce platform, distinct from crowdsourced logistics. The attributes of couriers mainly include quantity, location, speed, capacity, occupancy, list of loaded orders, and list of assigned orders. The list of assigned orders includes not only the list of loaded orders, but also includes the orders have been assigned to the courier but have not been picked by the vehicle. The loaded orders are the orders will be sent to the customers in the current route. The capacity is the maximum number of orders the courier can load, and the occupancy is the number of loaded orders on the vehicle in the current route. The major behaviour of a courier is to plan the delivery sequence of the loaded orders, which also refers to the vehicle routing planning of the courier. The courier's vehicle routing problem is not the primary focus of this study. Given that the delivery range for online fresh food services typically spans 3–5 km, considerably shorter than ride-hailing distances, and with a single depot, the routing problem for each courier can be viewed as a variation of the Traveling Salesman Problem (TSP).

Definition 3. Customers: The customers are the online fresh food customers located in a certain service area charged by one depot. The attributes of customer include location, order time, and maximum acceptable delivery time. The major behaviour of the customer is to report his or her satisfaction. The customer satisfaction is calculated as follows (see Eq. (1)):

$$S_{ci} = min\left(\frac{D_i}{A_{ji} - T_i}, 1.0\right) \tag{1}$$

in which T_i is the order time of the customer i, A_{ji} is the actual delivery time of the courier j to the customer i, D_i is the maximum acceptable delivery time for the customer i. The customers' satisfaction S_{ci} is 1.0 when the order is sent within the maximum acceptable delivery time, otherwise it is less than 1.0. Customer satisfactions are dynamically visualized in real-time via the ABM interface. The representation employs a color-coded scheme, where the color red signifies a satisfac-

tion level below 1.0. The intensity of the red hue correlates with decreasing satisfaction, such that a deeper shade of red indicates a lower satisfaction level. Conversely, the color green is indicative of a satisfaction level precisely at 1.0.

Definition 4. The depot: There is only one depot in the model. The depot is not only the origin/destination of instant delivery, but also is regarded as a dispatcher in the system. The main attribute of the depot agent is the task assignment list. The major behaviours of the depot are task buffering and dynamic batching, which are the strategies implemented to solve the order dispatching problem in instant delivery.

Definition 5. The Environment: The environment includes physical information such as the actual road network and residential buildings, and temporal information such as the time and days. Table 1 summarizes the frequently used notations in this paper.

3.2. Problems

In our instant delivery system, customers first place the orders to the depot. Secondly, the depot decides whether to assign the orders at this time step, and if so, to update the task assignment list. Thirdly, the depot selects one order from the task assignment list and decides to assign the order to which courier. Fourthly, the courier picks up the order at the depot, and fifthly, the courier plans the delivery sequence of the orders loaded in the vehicle. Sixthly, the courier records the delivery distance. Finally, the customer reports his or her satisfaction after receiving the order. The main process of the instant delivery is shown in Fig. 1. The interactions among customers and the depot, the depot and couriers, and couriers and customers are depicted by the arrows in Fig. 1. This iterative cycle continues until the completion of the one-day instant delivery service. This study primarily focuses on the second and third steps of this cycle: 1) whether to assign the orders at this time step; 2) assign the order to which courier, which are solved with two strategies: task buffering and dynamic batching.

It should be acknowledged that the objective of this study is not to optimize the customer satisfaction or delivery distance, but to understand the performances of the task buffering and dynamic batching strategies on customer satisfaction and delivery distance in different scenarios with various customer demand and courier fleet size.

This study is based on the following assumptions: 1) All couriers initiate their routes from the depot and conclude by returning to the same depot; 2) Each customer is exclusively served by a designated vehicle; 3) All couriers possess identical capacities; 4) Couriers have the capability to deliver multiple orders per route, but the combined occupancy must not exceed the load capacity; 5) Couriers refrain from returning to the depot for new orders until all loaded orders are delivered; 6) The depot is stocked with a comprehensive range of fresh products to fulfil the daily requirements of all customers, eliminating the necessity for transferring fresh products from another depot.

Table 1
Summary of important notations.

| Notation	Description
T_i	The order time of the customer i
A_{ji}	The actual delivery time of the courier j to the customer i
D_i	The maximum acceptable delivery time for the customer i
S_{ci}	Customer satisfaction value of customer i
T	Time of the decision point
l_{co}	The location list of all the couriers
O	The task assignment list
l_i	The location of customer i
n	The number of the orders
ETA_i	The estimated time of arrival of customer i's order
n_{co}	The number of working couriers
δ	A large negative number
C	Current time
$s_{t(G)}^k$	Global state
$s_{t(L)}^k$	Local state
$X_t^\pi(S_t)$	The action selected by policy π at state S_t
S_0	The current state
A_0	The current action
γ	The discount factor
$Q_\pi(s, a)$	Q function or action-value function
r_t	The reward that occurs when takes action a_t at the given state of s_t in the future
θ_i^-	The target network parameter
θ_i	The current network parameter
$dist(i, j)$	The matching degree between order i and courier j
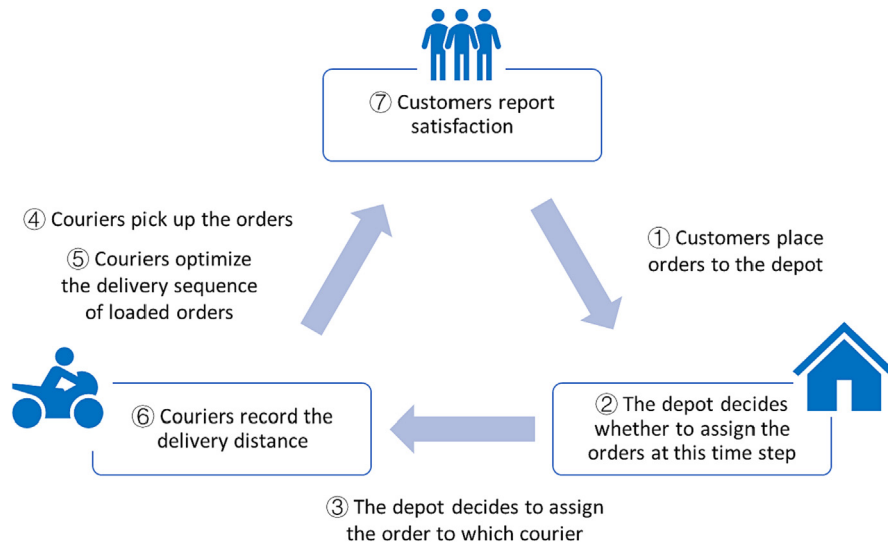
Fig. 1. The main process of instant delivery.

4. Adaptive agent-based order dispatching approach

4.1. Framework overview

We propose the ABOD approach to solve the above two order dispatching problems: 1) whether to assign the orders at this time step; 2) assign the order to which courier.

We use ABM to describe the order dispatching problem to better mimic the real-world instant delivery settings and take adaptive task buffering and dynamic batching decisions. The ABM model first initializes the spatial distributions of residential buildings and the road network, then generates the agents including the courier agents, the customer agents, and the depot agent. The interface for the ABOD is shown in Fig. 2.

The research framework is shown in Fig. 3. As for the integration of ABM, DRL and KM algorithm, the DRL first collects and processes information on the current state of the system from the ABM and, based on this information, determines the best action of the order (wait or to assign) based on the task buffering strategy. The task buffering strategy is implemented by DRL from a farsighted view. The implemented DRL algorithm in this study adopts a Multi-Agent Reinforcement Learning framework characterized by Centralized Training and Decentralized Execution. Specifically, the depot agent functions as the central management entity, assimilating behavioural information from customer agents and courier agents to inform its decision-making processes related to task buffering and dynamic batching. Subsequently, the customer agents and courier agents receive and execute decisions emanating from the depot. The dynamic batching strategy is considered in the order matching process. The matching value between an order and a courier with residual capacity is quantified based on the pick-up distance and delivery distance of the order considering other orders delivered along the same route.. The matching problem of orders and couriers is solved by KM algorithm in a global view. And the delivery time ETA_i is estimated as the basis of the reward if the action related to order i is to assign. Then the ABM changes the state of the system by applying the action and emerges the corresponding rewards and sends the rewards to the DRL. GAMA was used as an agent-based platform to simulate the application of the ABOD to the study area.

4.2. Task buffering

We model the task buffering strategy as a sequential decision process, a sequence of states connected by actions and transitions.

State. The state of the customer k at the time step t, s_t^k is composed by global state and local state, global state is information shared by all couriers in the system, and local state is information exclusively belonging to customer k itself. The global state $s_{t(G)}^k$ consists of the time of the decision point T, the task assignment list O, and the locations of all the couriers l_{co}. The local state $s_{t(L)}^k$ concerns the customer k waiting for order dispatching, comprised of the location l_k of the customer k, the maximum acceptable delivery time D_k of the customer k, and the order time T_k of customer k. Mathematically, the states are represented in Table 2.
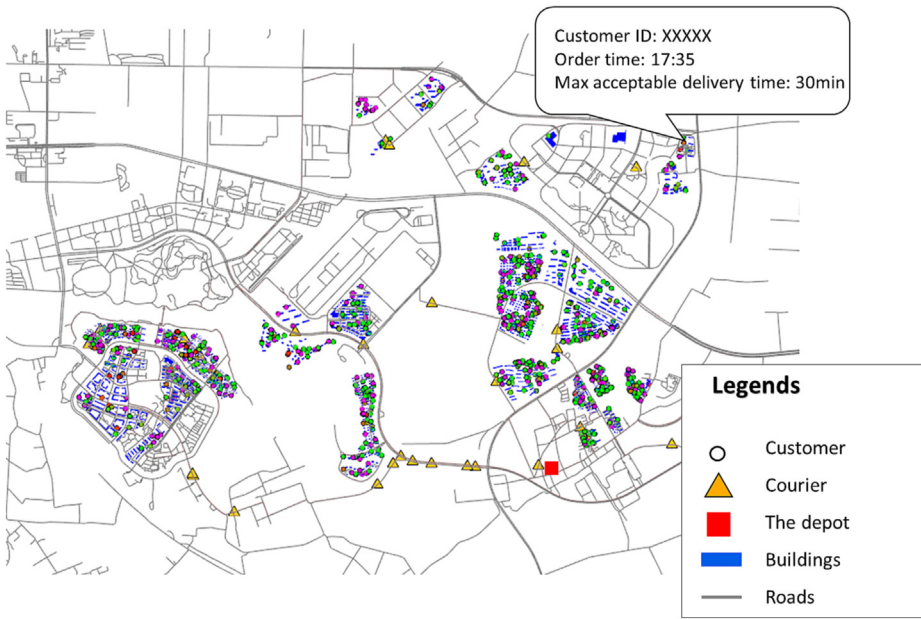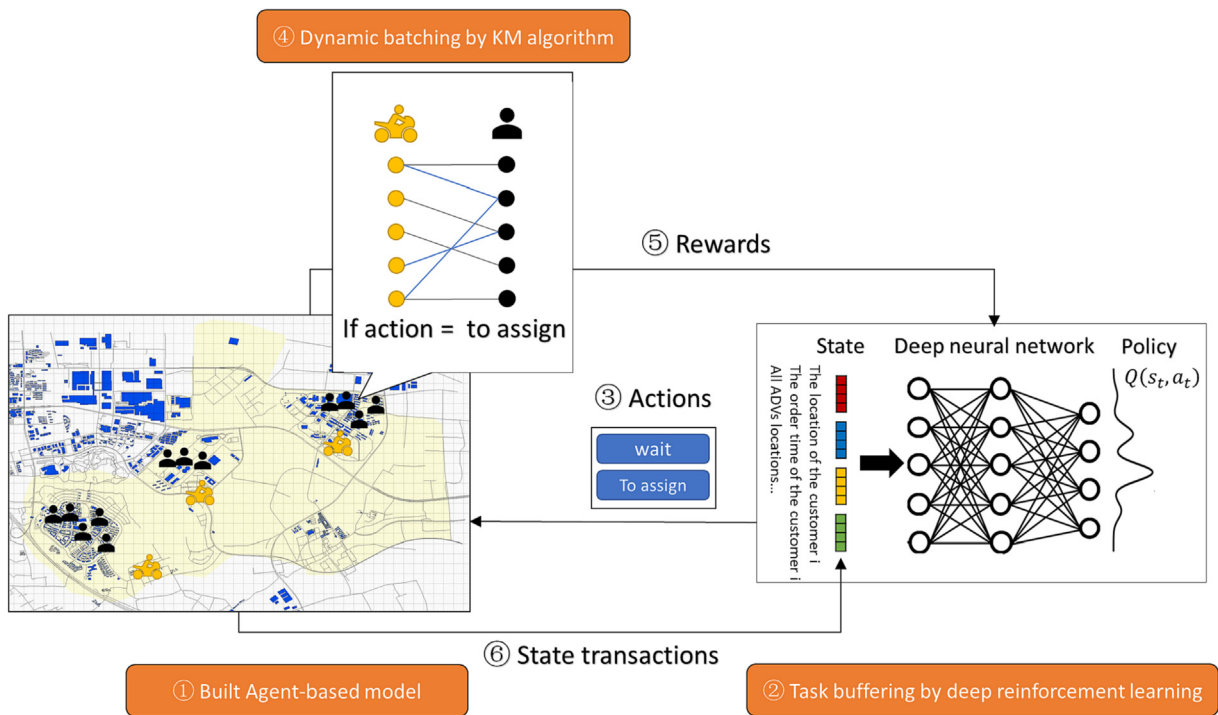
Fig. 2. The interface of ABOD.

Fig. 3. The research framework of ABOD.

Action. There are two types of actions in the action setting: to assign or wait, the action of "to assign" involves adding the order to the task assignment list, and the action of "wait" entails remaining the orders continue waiting. The buffering time of an order represents the duration between customer placement and its inclusion in the task assignment list O. For instance, immediate addition results in a buffering time of zero.

Reward. There are three types of rewards (refer to Eq. (2)). The first reward accounts for customer satisfaction, quantified as the average time difference between the maximum acceptable delivery time and the estimated time of arrival for the

Table 2
The components of the states.s_t^k

States	Index	Explanations
Global state $s_{t(G)}^k$	T	Time of the decision point
	l_{co}	The location list of all the couriers
	O	The task assignment list
Local state $s_{t(L)}^k$	l_k	the location of customer k
	D_k	the maximum acceptable delivery time of customer k
	T_k	The order time of customer k

orders to be assigned. The estimation of the time of arrival (ETA_i) for order i is determined upon its assignment to a specific courier in the subsequent stage of dynamic batching. The first reward may take a negative value if the estimated time of arrival exceeds the maximum acceptable delivery time. If an order is added to the assignment list after the current time surpasses the maximum acceptable delivery time, the reward becomes a significant negative value (δ); otherwise, the reward is set to zero. These configurations are designed to mitigate the accumulation of prolonged waiting times before orders are eventually assigned.

$$R_i = \begin{cases} \frac{1}{n}\sum_{i\in O}(T_i + D_i - ETA_i) \ (to\ assign) \\ 0 \ (wait\ and\ C \leq T_i + D_i) \\ \delta \ (wait\ and\ C > T_i + D_i) \end{cases} \tag{2}$$

State Transition. The state s_t^k is changed to $s_t^{k'}$ after performing the action. If the action is "wait", there is no change about the order. If the action is "to assign", the order is added to the task assignment list, and the order will be assigned to one of the couriers. Whether an order is assigned successfully or not depends on the second stage of dynamic batching. If assigned successfully, the order list of the courier will be updated, and the delivery sequence will be adjusted. The estimated delivery time of the orders will be re-estimated. If the order is not assigned successfully, the order will return to the task assignment list.

Objective Function. A solution to the task buffering problem is a policy $\pi \in \Pi$ that assigns an action to each state. The optimal solution is a policy π^* that maximizes the expected sum of long-term rewards when taking an action a in the state s and following a specific policy π thereafter. Satisfying the Bellman optimality equation, the optimal policy can be represented as choosing the action maximizing the expected sum of long-term rewards at each state, as defined as follows:

$$Q_{\pi^*}(s, a) = \max_\pi(Q_\pi(s, a))$$

$$= \max_\pi E\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a, \pi\right] \tag{3}$$

$$= \sum_{s'} p(s'|s, a)\left[r(s, a, s') + \gamma \max_{a'} Q_{\pi^*}(s', a')\right]$$

We use Deep Q-networks (DQN) to solve the task buffering problem. DQN is a value-based, off-policy DRL algorithm. The Q-network estimates action-values, generating numeric estimates to the expected future return for each action available in the current state. A neural network (NN) is created to approximate the value of the corresponding actions at certain states. The state and action spaces in the sequential decision process model are selected as features and approximate the value for each feature vector by DQN. All features are normalized using min–max normalization before being inputted into the NNs. For each iteration i, the DQN optimizes the objective function:

$$\mathscr{L}(\theta_i) = E_{(s,a,r,s')}\left[\left(R + \gamma \max_{a'}(Q(s', a'; \theta_i^-) - Q(s, a; \theta_i))^2\right)\right] \tag{4}$$

where (s, a, r, s') are the states, actions, rewards and next states sampled from the simulation environment. $Q(s, a; \theta_i)$ is a neural network parameterized by θ_i approximating the action-value function, θ_i^- is the target network parameter, and θ_i is the current network parameter. The DQN algorithm gathers experiences from the environment, stores them in a replay buffer, and updates the current network parameters through stochastic gradient descent on the loss function (as outlined in Algorithm 1). The target network computes an action-value estimate for the next state, s'. Target network parameters are synchronized with the current network every 360-time steps (equivalent to 6 hours). Action selection involves taking the argmax of the Q-network output, with the exception of a probability exploration ε, which decays from 1 to 0.01 over the training epochs.

Algorithm 1 (*DQN*).

1	Given: historical transitions pool P, a constant C.
2	Initialize replay memory M to capacity N and insert the terminal transitions set.
3	Initialize the state-action value network Q with random weights θ_0.
4	Initialize the target state-action value network Q^- with weights $\theta_0^- = \theta_0$.
5	For episode = 1, 2, $\cdots$, X do
6	For t = 1, 2, $\cdots$, T do
7	With probability ε select a random a_t.
8	Otherwise select $a_t = \arg\max\limits_{a} Q(s_{j+1}, a\|\theta_0)$.
9	Execute action a_t in simulation and observe reward r_t and s_{t+1}.
10	Store (s_t, a_t, r_t, s_{t+1}) in P.
11	Remove a transition sample (s_0, a, r, s_1) from P and store (s_0, a, r, s_1) in M.
12	End For
13	Sample a random mini-batch $\{(s_j, a_j, r_j, s_{j+1})\}$ from M.
14	$y_j = \begin{cases} r_j, & \text{if } s_{j+1} \text{ is a terminal state} \\ r_j + \gamma \max\limits_{a'} Q^-(s_{j+1}, a'\|\theta_0^-), & \text{otherwise} \end{cases}$
15	Perform a gradient descent step on $\left(y_j - Q(s_j, a_j)\|\theta_0\right)^2$ with respect to θ_0.
16	Every C steps set $Q^- = Q$
17	End For

4.3. Dynamic batching

The dynamic batching problem is considered in the real-time process of order matching, which is solved using one of combinational optimization algorithm, KM algorithm (also known as the Hungarian method) (Kuhn, 1955). KM algorithm is used to solve the maximum weight matching problem of the bipartite graph, which is widely applied in order dispatching for ride-hailing services (Tang et al., 2019, Tong et al., 2021, Wang and Guo, 2021, Yao et al., 2020). The KM algorithm exhibits notable computational efficiency in achieving an exact solution within polynomial time. Conceptually, the couriers with residual capacity and the orders can be construed as two subsets of a bipartite graph, with the orders representing those already incorporated into the task assignment list. At each time step, the objective is akin to identifying the optimal match between orders and couriers to optimize global gain, as elaborated in Algorithm 2. Building upon the real-time order dispatching algorithm introduced by Xu et al. (2018), the objective function of the centralized order dispatch algorithm is articulated as follows:

$$argmin \sum_{i=0}^{m} \sum_{j=0}^{n} dist(i,j) a_{ij} \tag{5}$$

s.t. $\sum_{i=0}^{m} a_{ij} = 1, j = 1, 2, 3 \cdots n$
$\sum_{j=0}^{n} a_{ij} = 1, i = 1, 2, 3 \cdots m$

where $a_{ij} = \begin{cases} 1 & (\text{if } order_i \text{ is assigned to } courier_j) \\ 0 & (\text{if } order_i \text{ is not assigned to } courier_j) \end{cases}$

Here, $i \epsilon [1, \cdots, m]$ corresponds to all available orders to be assigned at this time step, while $j \epsilon [1, \cdots, n]$ corresponds to couriers with residual capacity. When i = 0, it represents a virtual order, and when j = 0, it represents a virtual courier. $a_{0j} = 1$ represents that courier j has been assigned a virtual order indicating that the number of couriers is greater than the number of orders, and courier j has not been assigned an order. $a_{i0} = 1$ represents that order i has been assigned a virtual courier indicating the number of orders is greater than the number of couriers, and order i has not been assigned a courier. a_{00} indicates the situation where both orders and couriers are virtual and does not exist in reality. $dist(i,j)$ calculation is categorized into two scenarios. In the initial scenario, when the order is the first in the route, $dist(i,j)$ is the sum of the courier's pickup distance and the delivery distance of the order i. If the order is not the first in the route, $dist(i,j)$ represents the additional delivery distance of other orders in the same route resulting from the insertion of order i and the delivery distance of the order i. $dist(i,j)$ acts as the matching degree between orders and couriers with residual capacity, which is a specific value between different couriers and orders and updated every time step. And the estimated time of arrival of the order i, delivered by courier j, is determined through calculations based on $dist(i,j)$.

We illustrate the dynamic batching process in Fig. 4. The service time here refers to the pickup time and delivery time of the order. The new orders of next turn will be assigned to the couriers every time step when the couriers are serving the current turn orders. But the couriers will not back to the depot until they finish all the delivery of the current turn orders.
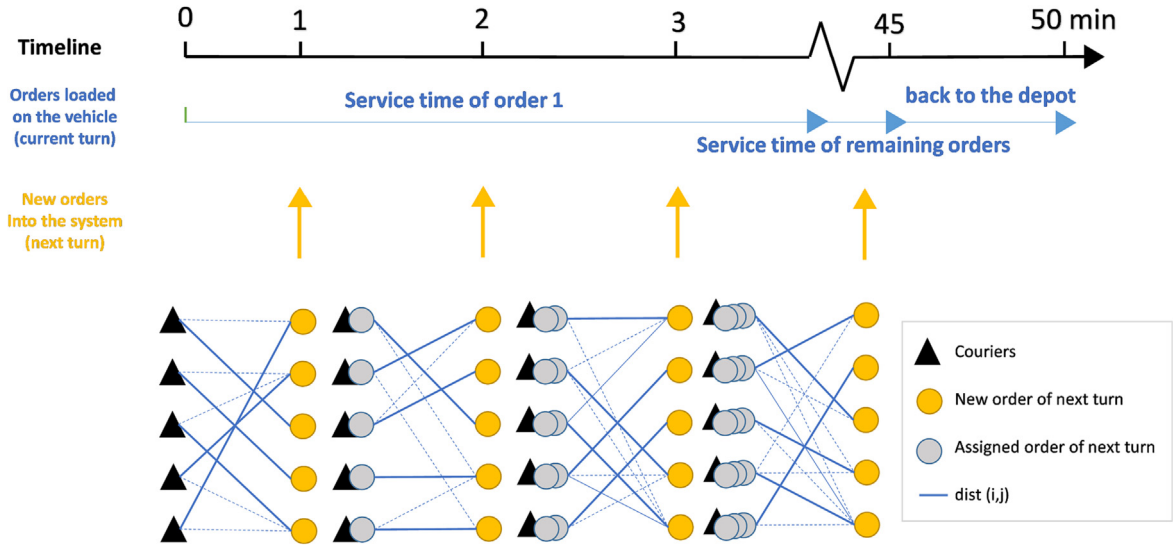
Fig. 4. The workflow of dynamic batching in a global view.

The occupancy of the courier depends on its capacity and the maximum acceptable delivery time of the orders loaded in the vehicle. The dynamic batching strategy is capable of responding to the real-time dynamics of the environment, aligning with the inherent characteristics of instant delivery.

Algorithm 2 (*KM Algorithm*).

1	Input: the set of orders to be assigned O, the set of available couriers C, and the distance d_{ij}.
2	Build a bipartite graph $G = (O \bigcup C, E)$, and set weight $w(o_i, c_j) = d_{ij}$.
3	Generate initial labelling $l_{o_i} = \max \{ w(o_i, c_j) \mid c_j \text{ in } C \}$, $l_{c_j} = 0$ and an empty matching M in G_l.
4	If M perfect, stop. Otherwise, pick free vertex $u \epsilon O$. Set $S = u, T = \varnothing, J_l(S) = \{ y \text{ in } C \mid s \text{ in } S, (s,y) \text{ in } M \}$
	If $J_l(S) = T$ do
5	$\alpha_l = \min\limits_{s \epsilon S, y \notin T} \{l(s) + l(y) - w(s,y)\}$
6	$\widehat{l}(v) = \begin{cases} l(v) - \alpha, v \in S \\ l(v) + \alpha, v \in T \\ l(v), otherwise \end{cases}$
7	If $J_l(S) \neq T$ do
8	choose $y \in J_l(S) - T$
9	If y free do
10	$u - y$ is an augmentation path. Augment M and go to step 3.
11	If y matched z do
12	extend alternating tree: $S = S \bigcup z, T = T \bigcup y$. Go to step 4.

5. Experiments

5.1. Experimental setup

5.1.1. Datasets

We use both synthesized and real-world datasets to evaluate the ABOD approach. The synthesized datasets are generated on instant delivery layouts with different demand and fleet sizes. The real-world dataset is derived based on historical delivery records from one fresh food e-commerce located in Shanghai. Table 3 shows the summary of the synthesized and real-world datasets. Snapshots of the synthesized and real cases are shown in Fig. 5(a) and Fig. 5(b).

Synthetic case. We create a synthetic region consisting of 25×25 grids. In this region, the depot is placed in the upper middle cell (i.e., cell [9, 4]). Customers are located in any cell with a uniform probability distribution. Specifically, the orders

Table 3
Summary of datasets.

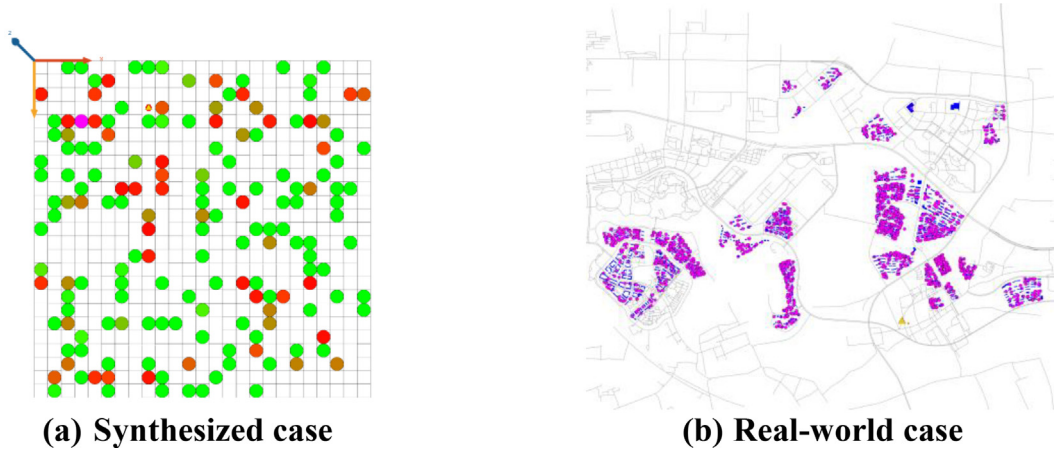| Name	Scope	Demand	Fleet size
Synthesized	25 grids × 25 grids	200, 400, 600, 800, 1000	5, 10, 15, 20
Real-world	8 km × 5 km	858	10

(a) Synthesized case   **(b) Real-world case**

Fig. 5. Snapshots of the synthesized and real-world cases.

are generated using a Poisson distribution with parameter λ_h for each hour h, which are predetermined in synthetic dataset. Let N be the daily order count, and α_h percentage of daily order at hour h. Then the hourly arrival rate can be defined as

$$\lambda_h = \alpha_h \times N \tag{6}$$

The probability P that between two orders being placed it will take a time interval t at hour h can be obtained from the transformation relationship between the Poisson distribution and the exponential distribution as follows:

$$P(X_h > t) = \frac{(\lambda_h t)^0 e^{-\lambda_h t}}{0!} = e^{-\lambda_h t} \tag{7}$$

where X_h is the time interval between two orders being placed at hour h. Then the random time interval t between two orders can be obtained from a random number rnd between 0 and 1 that fits a uniform distribution as follows:

$$t = \frac{\ln rnd}{-\lambda_h} \tag{8}$$

We conduct a sensitivity analysis to evaluate the robustness of our Synthetic model concerning the daily order count and courier fleet size. The daily order counts N are varied within the set {200, 400, 600, 800, 1000}. It is important to mention that the randomly generated orders are distributed to replicate observed patterns in real orders, including peak and non-peak hours. In the synthesized case, the courier fleet size is varied from 5 to 20, with an interval of 5. The capacity of each courier is set at 10 units.

Real-world case. We use one-day online fresh food delivery as the case study to validate and evaluate the ABOD's performance on the real case. The real data sources from different stakeholders were collected to guarantee the realism of the case study. The location of the case study is in Jiading, Shanghai. A fresh food e-commerce enterprise operates in this region. The distribution area covers 40 km², operated by one depot, which serves 14 residential communities with altogether more than six thousand households. The depot operates from 6:00 to 24:00, and the average daily demand on working days is approximately 840 orders, with a cutoff time for the latest order at 21:00. The time unit in the system is "minute", and the system updates the collected information of orders and couriers and making decisions every minute. There are 858 customers, whose maximum acceptable delivery times are summarized in Fig. 6. Fig. 7 shows the order times of the customers. There are two demand peak-hours, morning peak-hour (10:00–11:00) and evening peak-hour (19:00–20:00). The customers' maximum acceptable delivery time and order time are derived from survey data obtained from online fresh food customers. The spatial distribution of the orders is visualized in Fig. 8.
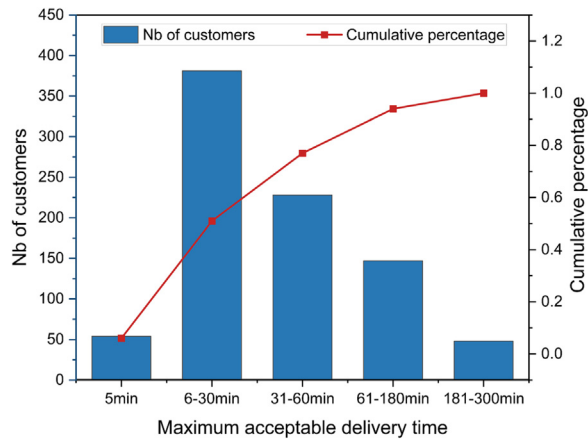
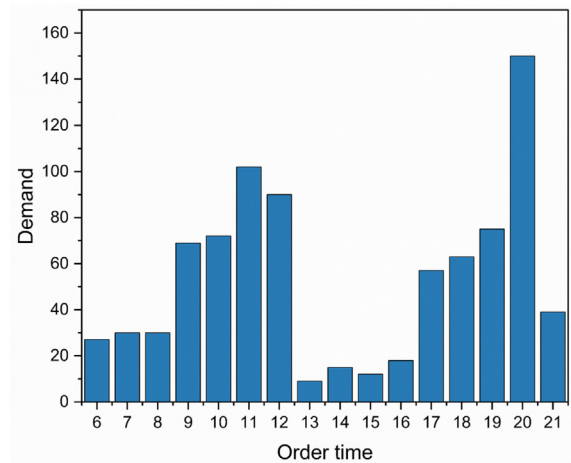**Fig. 6.** Customers' maximum acceptable delivery time.

Fig. 7. Daily demand distribution throughout the day.

Fig. 8. The spatial distribution of the order.

Table 4
The attributes of the couriers and depot.

Attributes	Value
Couriers' max driving speed (km/h)	40
Couriers' capacity (orders)	15
Depot location	(121.205, 31.266)
Depot working hours	6:00 – 24:00

The attributes of the courier agents were derived from the existing literature (Hu et al., 2020, Dimensions, 2014, Figliozzi and Jennings, 2020, Dimensions, 2020) (see Table 4), including maximum driving speed and capacity. The attributes of the depot agents are derived from the actual case study setting, including the location and the working hours.

5.1.2. Benchmark policies

Baseline. We compare the ABOD approach with two other dispatching strategies: (1) No buffer and No batching (NN); (2) No buffer but batching (NBA). We do not consider the strategy of "Buffer and No batching", which means the order waiting for assignment and the courier delivers only one order at one route. The strategy of "Buffer and No batching" is impractical for instant delivery as the large number of orders need to be sent with limited time.

No buffer and No batching (NN): The depot updates the task assignment list every one minute without task buffering. The depot assigns the orders to the couriers based on their time back to the depot, in other words, first back first assign. But the courier delivers only one order per route.

No buffer but batching (NBA): The depot updates the task assignment list every one minute without task buffering but with order batching. The depot assigns the orders to the couriers based on their time back to the depot, first back first assign, until the couriers are fully loaded.

Buffer and batching (ABOD)**:** The proposed ABOD approach in this study with order dispatching strategies of task buffering and dynamic batching.

Evaluation Metrics. We use four metrics to evaluate the strategies' effectiveness.

Customer satisfaction: It is the sum of the customer satisfaction after one day simulation.

Delivery distance: It is the sum of delivery distance of couriers in one day simulation.

Completion rate: It is the percentage of orders have been delivered to the corresponding customers successfully.

Hourly working couriers: It is the number of working couriers every hour in one day simulation. The courier who are not assigned orders are excluded.

All strategies and case studies are implemented in GAMA platform and the experiments are run on Intel(R) Core (TM) i7-8700 CPU @ 3.20 GHz 3.19 GHz.

5.2. Experimental results

5.2.1. Synthesized case

Fig. 9 illustrates a performance comparison between the ABOD and baseline policies concerning customer satisfaction, delivery distance, and competition rate. Our ABOD approach consistently achieves the highest level of customer satisfaction across various scenarios, surpassing the NN and NBA policies by 0.94% to 275.42% and 1.20% to 84.85%, respectively. Regarding delivery distance, the ABOD approach exhibits an average reduction of 11.38% and 2.19% compared to the NN and NBA policies, respectively. Despite an increase in demand, the ABOD approach maintains a completion rate above 80% even with only 5 couriers in the system. In contrast, the completion rates of the NN and NBA policies are unstable, with minimum values of 31.10% and 54.60%, respectively.

Furthermore, we observed that the delivery distance of the ABOD approach exhibits similar variability to that of the NBA policy, but which is noticeably less than that of the NN policy. This indicates that the presence or absence of task buffering does not significantly affect the delivery distance, but dynamic batching can reduce the delivery distance greatly.

The ABOD approach demonstrates strong adaptability for two primary reasons: Firstly, based on the performance of the NBA and NN policies, it can be inferred that ABOD strategy not only enhances customer satisfaction but also reduces delivery distance, particularly in high-demand scenarios. Secondly, completion rates remain stable with our ABOD approach, while other baseline policies exhibit significant variability. This suggests that our approach can adaptively make order dispatching decisions and maintain operational stability despite fluctuations in demand and supply.

5.2.2. Shanghai case

We use shanghai case to validate our model. We compare the simulated number of working couriers at different times-of-the-day with real-world records. Overall, the simulated number of working couriers matches the actual labour input without large deviation (see Fig. 10).

The total customer satisfaction results for ABOD, NBA and NN are 827.07, 762.02, and 629.49, respectively. Our approach achieves the highest total customer satisfaction, 8.53% and 31.39% higher than that in the NBA and NN, respectively. Fig. 11 plots customer satisfaction at different hours of the day according to the three dispatching policies. We found ABOD
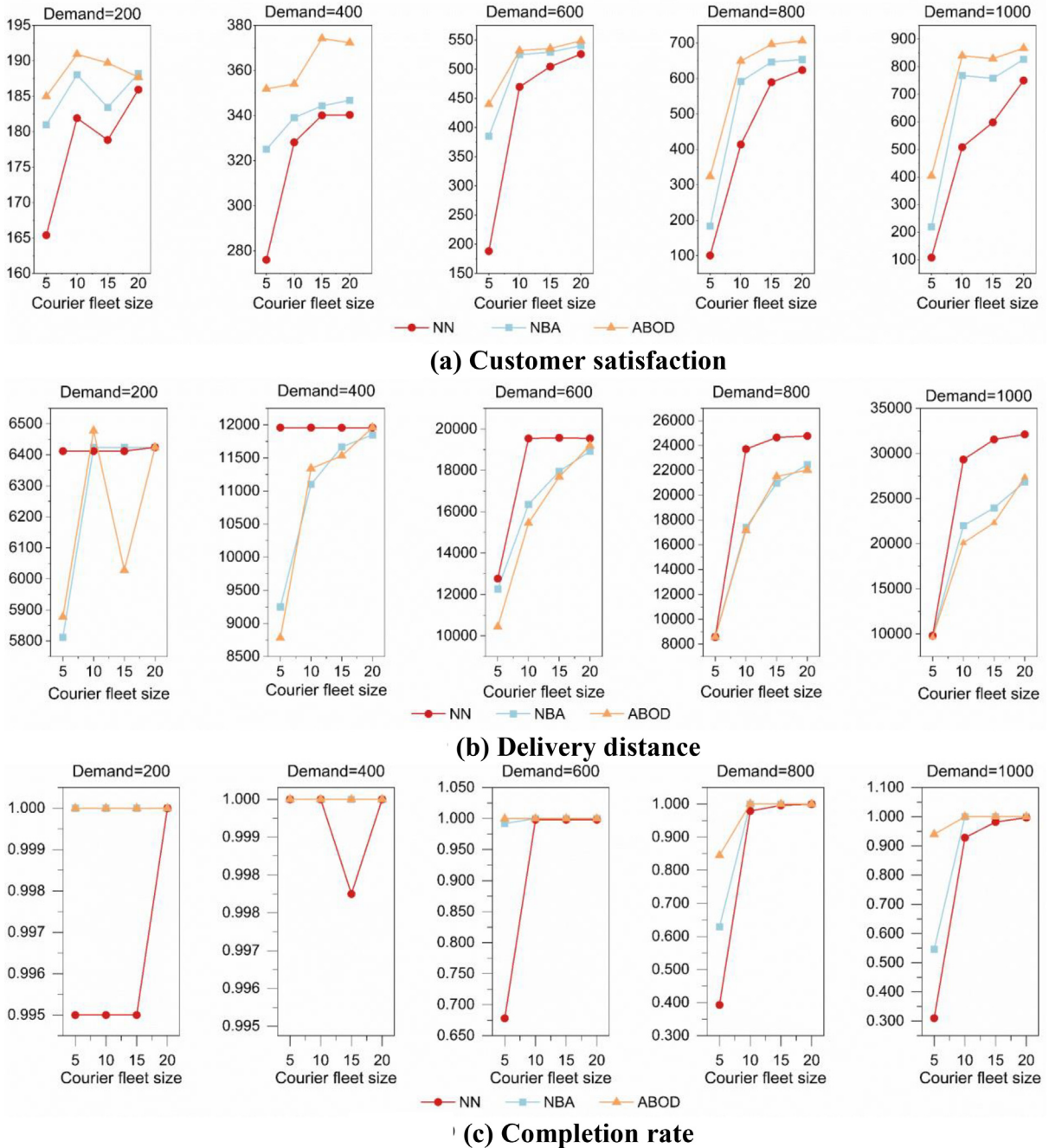
(a) Customer satisfaction

(b) Delivery distance

(c) Completion rate

Fig. 9. Performance comparison in Synthesized case.

approach had the best performances in the peak hours (11:00–12:00 and 20:00–21:00). These results show that our task buffering strategy can learn the optimum buffering time to ensure the customers' satisfaction based on the current demand, but also suggest that our approach can be applicable to a peak hour scenario to improve the customer satisfaction.

We also compare the hourly working couriers with ABOD and other two baseline policies. Based on the hourly working couriers at different hours of the day, we can find there are full fleet size couriers working around peak hours (9:00–13:00 and 17:00–21:00), but the hourly working couriers is reduced by almost half in non-peak hours (14:00–16:00). ABOD approach requires the lowest number of working couriers in most of the time (see Fig. 12). The average number of couriers operated per hour in our approach is nearly same with NBA, and 16.28% less than that in NN policies. These findings are consistent with the objectives of the dynamic batching strategy aiming to alleviate the delivery pressure and reduce the labour cost accordingly.
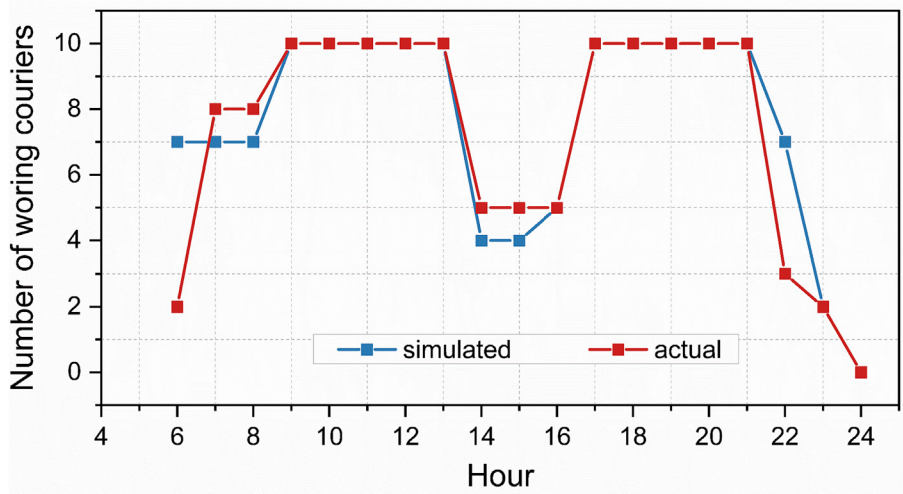
Fig. 10. Actual number of working couriers vs simulated number of working couriers.
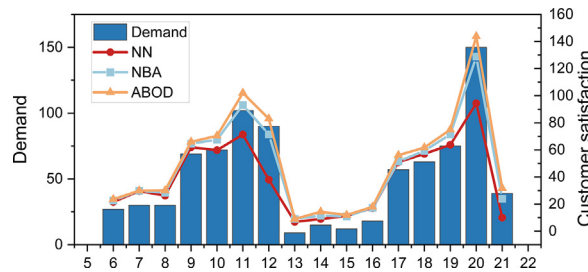

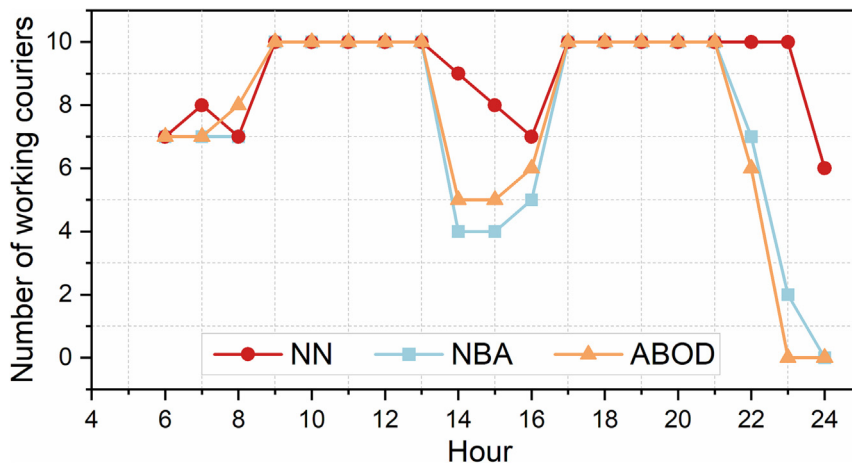**Fig. 11.** Customer satisfaction at different hours of the day.

Fig. 12. hourly working couriers at different hours of the day.

We examine the correlations between average buffering time, average customer satisfaction and hourly demand. The average buffering time and average customer satisfaction refer to the buffering time and customer satisfaction at individual average level. Hourly demand means how many people placed orders during this hour. Fig. 13 shows that with the increase of the hourly demand, the average buffering time is reduced and vice versa. The longest average buffering time occurs in non-peak hour (14:00–16:00), which is 142 seconds. We conduct a regression analysis to analyse the impact of hourly
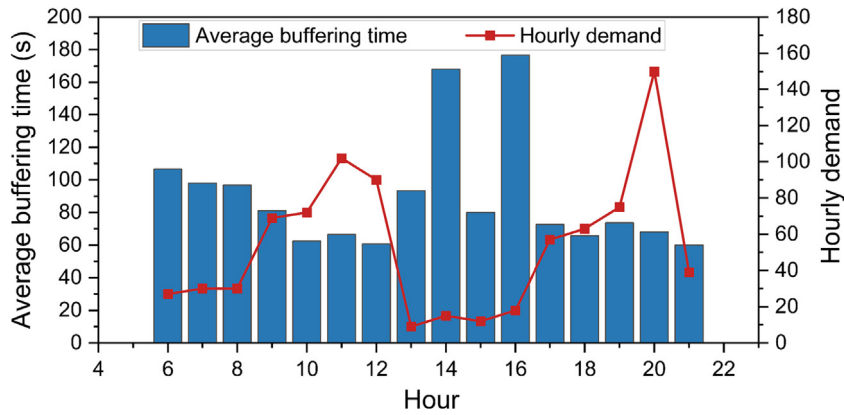
Fig. 13. The correlation between average buffering time and hourly demand.

Table 5
Impact of hourly demand on average buffering time.

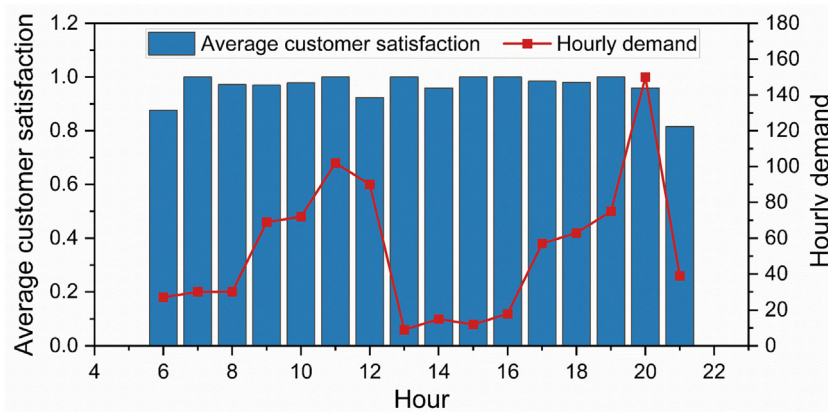| Items	Coefficients	standard error	t Stat	P-value
Intercept	117.36	13.13	8.93	0.00
Hourly demand	−0.52	0.20	−2.60	0.02
R Square	0.33			
Adjusted R Square	0.28			

Fig. 14. The correlation between average customer satisfaction and hourly demand.

demand on average buffering time (see Table 5). According to Table 5, the results support the observations in Fig. 13, the coefficient of hourly demand is significantly negative (−0.52, $p < 0.05$). It can be estimated that the adaptive task buffering decisions are made with ABOD approach, the orders in non-peak hour usually have longer buffering times than orders placed at other time. But the customer satisfactions in non-peak hours are not less than the customer satisfactions at other time (see Fig. 14). Based on the hourly working couriers in Fig. 12, it can be explained that there are less orders need to be served in non-peak hours, thus these orders are buffered and batched, and less couriers are needed to serve these orders. But the customer satisfactions of these orders are not influenced. As shown in Fig. 14, we can find there are no obvious correlations between average customer satisfaction and hourly demand. The customer satisfactions are maintained at a relatively stable and high level (>0.8). The ABOD approach demonstrates significant adaptability in maintaining high customer satisfaction levels across various demand scenarios.

6. Conclusions

In this study we propose ABOD approach that incorporates task buffering and dynamic batching strategies. The task buffering strategy optimizes the buffering time of an order before it is assigned to one courier, which can tackle the uncertain

delivery demand. The dynamic batching strategy enables the couriers to deliver more than one order per route, with an evaluation of which orders should be grouped together for efficient delivery, which aims to reduce the delivery pressure. Our approach combines DRL for long-term efficiency in task buffering and utilizes the KM algorithm for global optimization in dynamic batching between couriers and orders. We evaluate the performance of the ABOD approach using synthesized cases with varying demand and couriers, as well as a real-world online fresh food delivery case from Shanghai. We compared the ABOD approach to a NN policy and an NBA policy in customer satisfaction, delivery distance, completion rate, and hourly working couriers. The results show that our approach achieves the highest customer satisfaction, the highest completion rate and the least delivery distance among three policies and implements relatively minimum hourly working couriers to serve the same demand. It is also demonstrated that the ABOD approach can buffer the order assignment autonomously while keep the customer satisfaction at a high level across different demand scenarios.

This research is the first to utilize ABM rather than mathematical models in describing the instant delivery problem. ABM is particularly suitable for instant delivery as ABM and instant delivery share similar characteristics, such as dynamics, interactivity, and uncertainty. The proposed order dispatching strategies of dynamic batching and task buffering effectively alleviate delivery pressure and address high uncertainties, which have received limited attention in previous instant delivery studies. The simulation results provide specific recommendations to logistics providers in the instant delivery industry. Moreover, we demonstrate the adaptability of the ABOD approach in various demand and fleet size scenarios.

Limitations of the study of particular note is that the route planning of the couriers obeys the simple principle of "first order first serve", which is a compromise to adapt to dynamic environment and should be improved in the future study. Dynamic order dispatching (i.e., returning to the depot to load additional goods in the middle of a delivery route in progress) is not taken into account, which is because there is only one depot in case studies, and dynamic order dispatching is more often examined in the context of multiple depots (Kronmueller et al., 2021). In terms of future research directions, we are interested in dynamic order dispatching for food delivery with multiple depots. The current DRL algorithm employed in the paper relies on the DQN algorithm. Future research endeavours will focus on advancing DRL algorithms and conducting a comparative analysis of task buffering performances with the latest reinforcement learning methods in instant delivery.

Conflicts of Interest

The author would like to thank the editor and reviewers for their insightful comments to improve the quality of this paper. The authors declare that the contents of this article have not been published previously. All the authors have contributed to the work described, read and approved the contents for publication in this journal. All the authors have no conflict of interest with the funding entity and any organization mentioned in this article in the past three years that may have influenced the conduct of this research and the findings.

CRediT authorship contribution statement

Miaojia Lu: Conceptualization, Data curation, Funding acquisition, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Xinyu Yan:** Data curation, Methodology, Software, Writing – original draft. **Shadi Sharif Azadeh:** Conceptualization, Writing – review & editing. **Pengling Wang:** Conceptualization, Formal analysis, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

References

Bonabeau, E., 2002. Agent-based modeling: methods and techniques for simulating human systems. PNAS 99 (Suppl 3), 7280–7287.

Bozanta, A., Cevik, M., Kavaklioglu, C., Kavuk, E.M., Tosun, A., Sonuc, S.B., Duranel, A., Basar, A., 2022. Courier routing and assignment for food delivery service using reinforcement learning. Comput. Ind. Eng. 164, 107871.

Chen, X., Ulmer, M.W., Thomas, B.W., 2022. Deep Q-learning for same-day delivery with vehicles and drones. Eur. J. Oper. Res. 298, 939–952.

Dablanc, L., Morganti, E., Arvidsson, N., Woxenius, J., Browne, M., Saidi, N., 2017. The rise of on-demand 'Instant Deliveries' in European cities. Supply Chain Forum: An International Journal 18, 203–217.

DIMENSIONS. 2014. *Starship Robot* [Online]. Available: https://www.dimensions.com/element/starship-robot [Accessed June 17 2022].

DIMENSIONS. 2020. *Nuro R2* [Online]. Available: https://www.dimensions.com/element/nuro-r2 [Accessed June 17 2022].

Du, J., Guo, B., Liu, Y., Wang, L., Han, Q., Chen, C., Yu, Z., 2019. CrowDNet: Enabling a crowdsourced object delivery network based on modern portfolio theory. IEEE Internet Things J. 6, 9030–9041.

Figliozzi, M.A., Jennings, D., 2020. A study of the competitiveness of autonomous delivery vehicles in urban areas. Civil and Environmental Engineering Faculty Publications and Presentations 548.

Fikar, C., Hirsch, P., Gronalt, M., 2018. A decision support system to investigate dynamic last-mile distribution facilitating cargo-bikes. Int J Log Res Appl 21, 300–317.

Ge, Q., Han, K., Liu, X., 2021. Matching and routing for shared autonomous vehicles in congestible network. Transportation Research Part E: Logistics and Transportation Review 156.

Guo, B., Wang, S., Ding, Y., Wang, G., He, S., Zhang, D. & He, T. Concurrent Order Dispatch for Instant Delivery with Time-Constrained Actor-Critic Reinforcement Learning. 2021 IEEE Real-Time Systems Symposium (RTSS), 7-10 Dec 2021 2021. 176–187.

Hofmann, W., Assmann, T., Neghabadi, P.D., Cung, V.-D., Tolujevs, J., 2017.. A simulation tool to assess the integration of cargo bikes into an urban distribution system. The 5th International Workshop on Simulation for Energy, Sustainable Development & Environment (SESDE 2017).

Holler, J., Vuorio, R., Qin, Z., Tang, X., Jiao, Y., Jin, T., Singh, S., Wang, C. & Ye, J. Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. 2019 IEEE International Conference on Data Mining (ICDM), 2019. IEEE, 1090–1095.

Hu, J., Zhang, Y., Han, S., 2020. Research on distribution optimization of electric unmanned vehicles in urban logistics. Journal of Zhejiang Institute of Science and Technology 44, 124–133.

Huang, Y., Ding, Z.H., Lee, W.J., 2023. Charging Cost-Aware Fleet Management for Shared On-Demand Green Logistic System. IEEE Internet Things J. 10, 7505–7516.

IMEDIA RESEARCH. 2022. 2022 China's Fresh E-commerce Industry Development Trends: High cost-effectiveness and timely delivery drive the substantial growth of the fresh e-commerce sector [Online]. Available: https://www.iimedia.cn/c1020/85058.html [Accessed June 17 2022].

Jahanshahi, H., Bozanta, A., Cevik, M., Kavuk, E.M., Tosun, A., Sonuc, S.B., Kosucu, B., Başar, A., 2022. A deep reinforcement learning approach for the meal delivery problem. Knowl.-Based Syst. 243, 108489.

Kavuk, E.M., Tosun, A., Cevik, M., Bozanta, A., Sonuç, S.B., Tutuncu, M., Kosucu, B., Basar, A., 2022. Order dispatching for an ultra-fast delivery service via deep reinforcement learning. Appl. Intell. 52, 4274–4299.

Kronmueller, M., Fielbaum, A. & Alonso-Mora, J. On-demand grocery delivery from multiple local stores with autonomous robots. 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), 2021. IEEE, 29–37.

Kuhn, H.W., 1955. The Hungarian method for the assignment problem. Naval research logistics quarterly 2, 83–97.

Kuhnle, A., Schäfer, L., Stricker, N., Lanza, G., 2019. Design, implementation and evaluation of reinforcement learning for an adaptive order dispatching in job shop manufacturing systems. Procedia CIRP 81, 234–239.

Li, J., Yang, S., Pan, W., Xu, Z., Wei, B., 2022. Meal delivery routing optimization with order allocation strategy based on transfer stations for instant logistics services. IET Intelligent Transport System 16, 1108–1126.

Liao, W., Zhang, L., Wei, Z., 2020. Multi-objective green meal delivery routing problem based on a two-stage solution strategy. J. Clean. Prod. 258, 120627.

Liu, Y., Guo, B., Chen, C., Du, H., Yu, Z., Zhang, D., Ma, H., 2018. FooDNet: Toward an optimized food delivery network based on spatial crowdsourcing. IEEE Trans. Mob. Comput. 18, 1288–1301.

Liu, Y., Wu, F., Lyu, C., Li, S., Ye, J., Qu, X., 2022. Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform. Transportation Research Part E: Logistics and Transportation Review 161.

Lv, J., Zheng, L., Liao, L. & Chen, X. 2021. Ride-sharing matching of commuting private car using reinforcement learning. International Conference on Knowledge Science, Engineering and Management, Springer, 679–691.

Malus, A., Kozjek, D., 2020. Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. CIRP Ann. 69, 397–400.

Mo, J., Ohmori, S., 2021. Crowd sourcing dynamic pickup & delivery problem considering task buffering and drivers' rejection-application of multi-agent reinforcement learning. WSEAS Trans. Bus. Econ. 18, 636–645.

Noruzoliaee, M., Zou, B., 2022. One-to-many matching and section-based formulation of autonomous rideshare equilibrium. Transp. Res. B Methodol. 155, 72–100.

Poeting, M., Schaudt, S., Clausen, U., 2019. Simulation of an optimized last-mile parcel delivery network involving delivery robots. Advances in Production, Logistics and Traffic.

Shi, D., Tong, Y., Zhou, Z., Xu, K., Tan, W. & Li, H. 2022. Adaptive task planning for large-scale robotized warehouses. 2022 IEEE 38th International Conference on Data Engineering (ICDE), 2022. IEEE, 3327-3339.

Tang, X., Qin, Z., Zhang, F., Wang, Z., Xu, Z., Ma, Y., Zhu, H. & Ye, J. 2019. A deep value-network based approach for multi-driver order dispatching. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 1780-1790.

Tong, Y., Wang, L., Zimu, D., Ding, B., Chen, L., Ye, J. & Xu, K. 2017. Flexible online task assignment in real-time spatial data. Proceedings of the VLDB Endowment.

Tong, Y., Shi, D., Xu, Y., Lv, W., Qin, Z., Tang, X., 2021. Combinatorial optimization meets reinforcement learning: Effective taxi order dispatching at large-scale. IEEE Trans. Knowl. Data Eng.

Turhanlar, E.E., Ekren, B.Y., Lerher, T., 2022. Autonomous mobile robot travel under deadlock and collision prevention algorithms by agent-based modelling in warehouses. Int J Log Res Appl, 1–20.

Ulmer, M.W., Thomas, B.W., Mattfeld, D.C., 2019. Preemptive depot returns for dynamic same-day delivery. EURO Journal on Transportation and Logistics 8, 327–361.

Voccia, S.A., Campbell, A.M., Thomas, B.W., 2019. The same-day delivery problem for online purchases. Transp. Sci. 53, 167–184.

Wang, N., Guo, J., 2021. Modeling and optimization of multiaction dynamic dispatching problem for shared autonomous electric vehicles. J. Adv. Transp. 2021, 1–19.

Wang, S., Hu, S., Guo, B., Wang, G., 2023. Cross-Region Courier Displacement for On-Demand Delivery With Multi-Agent Reinforcement Learning. IEEE Trans. Big Data 9, 1321–1333.

WU, Y., DING, Y., DING, S., SAVARIA, Y. & LI, M. J. M. P. I. E. 2021. Autonomous Last-Mile Delivery Based on the Cooperation of Multiple Heterogeneous Unmanned Ground Vehicles. 2021.

XU, Z., LI, Z., GUAN, Q., ZHANG, D., LI, Q., NAN, J., LIU, C., BIAN, W. & YE, J. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 905-913.

Yan, P.Y., Yu, K.Z., Chao, X.L., Chen, Z.B., 2023. An online reinforcement learning approach to charging and order-dispatching optimization for an e-hailing electric vehicle fleet. Eur. J. Oper. Res. 310, 1218–1233.

Yao, F., Zhu, J., Yu, J., Chen, C., Chen, X., 2020. Hybrid operations of human driving vehicles and automated vehicles with data-driven agent-based simulation. Transp. Res. Part D: Transp. Environ. 86, 102469.

Zhang, W., Guhathakurta, S., Fang, J., Zhang, G., 2015. Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. Sustain. Cities Soc. 19, 34–45.

Zhen, L., Wu, J., Laporte, G., Tan, Z., 2023. Heterogeneous instant delivery orders scheduling and routing problem. Comput. Oper. Res. 157, 106246.

Zou, G., Tang, J., Yilmaz, L., Kong, X., 2021. Online food ordering delivery strategies based on deep reinforcement learning. Appl. Intell., 6853–6865