International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Spatio-temporal task pricing for shared electric micro-mobility battery-swapping platform with reinforcement learning

Minjeong Kim & Ilkyeong Moon

To cite this article: Minjeong Kim & Ilkyeong Moon (2025) Spatio-temporal task pricing for shared electric micro-mobility battery-swapping platform with reinforcement learning, International Journal of Production Research, 63:4, 1473-1494, DOI: 10.1080/00207543.2024.2379561

To link to this article: https://doi.org/10.1080/00207543.2024.2379561

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2025, VOL. 63, NO. 4, 1473–1494
https://doi.org/10.1080/00207543.2024.2379561

Taylor & Francis
Taylor & Francis Group

Spatio-temporal task pricing for shared electric micro-mobility battery-swapping platform with reinforcement learning

Minjeong Kim[a] and Ilkyeong Moon [a,b]

[a]Department of Industrial Engineering, Seoul National University, Seoul, Korea; [b]Institute of Engineering Research, Seoul National University, Seoul, Korea

ABSTRACT

Spatial crowdsourcing has emerged in shared electric micro-mobility platforms, compensating occasional drivers (ODs) per task of swapping micro-mobility batteries. As ODs autonomously select tasks only when satisfied with predetermined compensation and travel distance, a traditional uniform pricing strategy results in possible low task completion. To resolve the imbalance between ODs and tasks, this study introduces a spatio-temporal pricing strategy where task prices differ by region and time interval. Considering the daily variations in task distribution and OD availability, the goal is to minimise the platform costs equal to the sum of total OD wages and penalties for uncompleted tasks. The reinforcement learning approach with proximal policy optimisation (PPO) is implemented to generate real-time continuous task prices. A domain-specific masking technique is incorporated to improve the learning process by disregarding the data from inactive grids in loss calculations. Computational results show that the PPO agent strategically raises prices in regions with insufficient ODs according to the OD density level. Further comparison with the mixed integer programming model with perfect information on ODs' willingness-to-accept parameters demonstrates the superior capability of our algorithm in navigating the uncertainties of OD task acceptance. A sensitivity analysis provides insights into the decision of system parameters.

ARTICLE HISTORY
Received 13 March 2024
Accepted 2 July 2024

KEYWORDS
Spatial crowdsourcing; shared electric micro-mobility; occasional drivers; spatio-temporal pricing; reinforcement learning; proximal policy optimisation

SUSTAINABLE DEVELOPMENT GOALS
SDG 11: Sustainable cities andcommunities

1. Introduction

Spatial crowdsourcing has been booming with the emergence of the gig economy. It is a business trend to engage individuals through an open call via a digital platform to perform the company's location-dependent tasks. Currently, spatial crowdsourcing applications are pervasive: ride-sharing services (e.g. Uber; https://www.uber.com), food delivery (e.g. DoorDash; https://www.doordash.com/), parcel delivery (e.g. Shipt; https://www.shipt.com/), information collection (e.g. Waze; https://www.waze.com/), and micro-tasks (e.g. Gigwalk; https://www.gigwalk.com/). These platforms leverage the power of a distributed network of crowds to complete spatial tasks with flexible working schedules (Chen et al. 2017).

A recent application of spatial crowdsourcing has appeared in the shared electric micro-mobility (SEM) platform. SEMs are widely distributed across the city to facilitate first-mile and last-mile transportation (Kadri, Labadi, and Kacem 2015; Singgih and Kim 2020). To maintain a high battery level in SEMs, the platform utilises crowdsourced workers to swap batteries at night when SEM usage decreases (Osorio, Lei, and Ouyang 2021). The typical process is as follows. First, the platform displays the locations of low-battery SEMs and rewards workers per task through the app. Second, workers reserve tasks that meet satisfactory criteria, typically based on travel distance and reward, in a first-come, first-served basis. Once a booking is confirmed, the crowdsourced worker travels to the corresponding task location and then replaces the depleted battery with a fully charged one. Last, upon confirmation of task completion, the worker receives the reward assigned to the task. Two major e-scooter sharing platforms, Lime and Bird, utilise this programme with unique titles like 'Lime juicers' and 'Bird hunters,' then gather the crew online (https://www.li.me/, https://www.bird.co/).

In the relevant literature, crowdsourced workers who participate in this decentralised workforce are called 'occasional drivers' (ODs). First introduced in Archetti, Savelsbergh, and Grazia Speranza (2016), the term OD denotes in-store customers willing to deliver parcels on their way home for small compensation. However, they only agree if the delivery task falls within a certain distance threshold specific to an OD. In later research, task acceptance conditions have been extended to cover

CONTACT Ilkyeong Moon ikmoon@snu.ac.kr Institute of Engineering Research, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Korea

delivery time windows (Macrina et al. 2020) and the willingness of ODs to make a limited number of stops (Arslan et al. 2019). Meanwhile, the limitations of these studies lie in treating the task price as a given parameter rather than as a decision variable, even though it is a crucial factor in attracting ODs (Archetti, Savelsbergh, and Grazia Speranza 2016; Arslan et al. 2019; Gdowska, Viana, and Pedro Pedroso 2018). Alnaggar, Gzara, and Bookbinder (2021) attributes the lack of extensive research to the limitation that fixed price facilitates obtaining the optimal task assignment solution.

However, task pricing becomes crucial when ODs select tasks voluntarily, as it becomes the only inducement strategy to attract ODs. In particular, the flexibility of ODs and the distribution of tasks change daily in the SEM battery-swapping platform. Thus, a uniform task pricing strategy may not sufficiently motivate ODs to accept tasks requiring longer travel distances. Employing a spatio-temporal pricing approach is one effective strategy to prevent regional disparities in task fulfillment. This approach divides the target area into uniform hexagonal grids and breaks down the service time into intervals, dynamically assigning pricing for each grid and time interval (Chen et al. 2021; Yun et al. 2022). Tasks within the same grid are set at equal prices.

Dynamic pricing is well established in the ride-sourcing platform, which likewise employs self-scheduling contractors. However, pricing for the SEM battery-swapping platform presents distinct challenges. In the battery-swapping platform, incomplete tasks remain in the system, potentially blocking revenue the next day. This issue necessitates including a penalty cost in the objective function to ensure smooth operations. In contrast, the ride-sourcing platform allows unmatched orders to exit the system without any disruption, focussing on maximising profits (total order revenue minus driver wage) as its primary objective (Chen et al. 2021). Additionally, workers on the battery-swapping platform can view all task prices simultaneously, influencing their decision-making, unlike ride-sourcing drivers, who only see one order at a time and decide based on the order's individual price. This difference requires the battery-swapping platform to strategically offer price combinations that attract workers to select proper tasks, leading to optimal outcomes.

Therefore, the spatio-temporal pricing for the battery-swapping platform becomes a unique problem of deciding optimal task (grid) price combinations. The objective is to minimise the platform's daily operational cost, the sum of total OD wages and penalties for incomplete tasks. Based on the industry characteristics, the proposed strategy should address the following four challenges:

(i) *Unknown and heterogeneous task acceptance condition of ODs:* Only ODs satisfied with the price perform the task. However, the platform's pricing decision should be made before the OD's decision. Previous literature gained feedback by requiring ODs to submit their task acceptance conditions in advance or via an auction system to indirectly reveal an OD's task preference (Kafle, Zou, and Lin 2017; Mancini and Gansterer 2022). However, when ODs compete for task reservations in real time, gaining OD feedback is unsuitable. Instead, the strategy should predict ODs' task acceptance and set prices that enable ODs who are interested to secure tasks. A trade-off exists in setting task prices where too high a price can lead to increased cost, whereas too low a price risks the task being left uncompleted.

(ii) *Uncertainty of ODs' working schedule:* The platform has no information on when and where the ODs will participate in the platform. Therefore, if there are sufficient ODs in a specific region at a particular time, directing some ODs to regions with OD shortages can address potential future imbalances. As the pricing decision for an interval sequentially influences the task distribution in the next interval, including the relocation of ODs, this becomes a sequential optimisation problem. Thus, the price must be considered with a long-term effect.

(iii) *Dependency of ODs' selection:* ODs select tasks from the grid based on cost-effectiveness, following a first-come, first-served protocol. Thus, earlier in the queue, ODs may preempt the most desirable tasks, forcing subsequent ODs to choose their second-best options. As a result, the OD's selection process depends on the individual task prices and the combined pricing across all tasks. This dependency complicates determining the lowest acceptable price to attract ODs.

(iv) *Real-time solution of high dimensionality:* Price updates occur frequently within a few minutes. For practicality, the optimal grid price should be derived instantly according to the current participating ODs and task distribution. This fact requires the pricing algorithm to perform rapid computations for generating a high-dimensional solution, whose complexity escalates with the number of regions (Pan et al. 2019).

In pursuit of addressing these issues, this study introduces a reinforcement learning-based (RL-based) task pricing algorithm for the SEM battery-swapping platform. The objective is to minimise the platform's daily

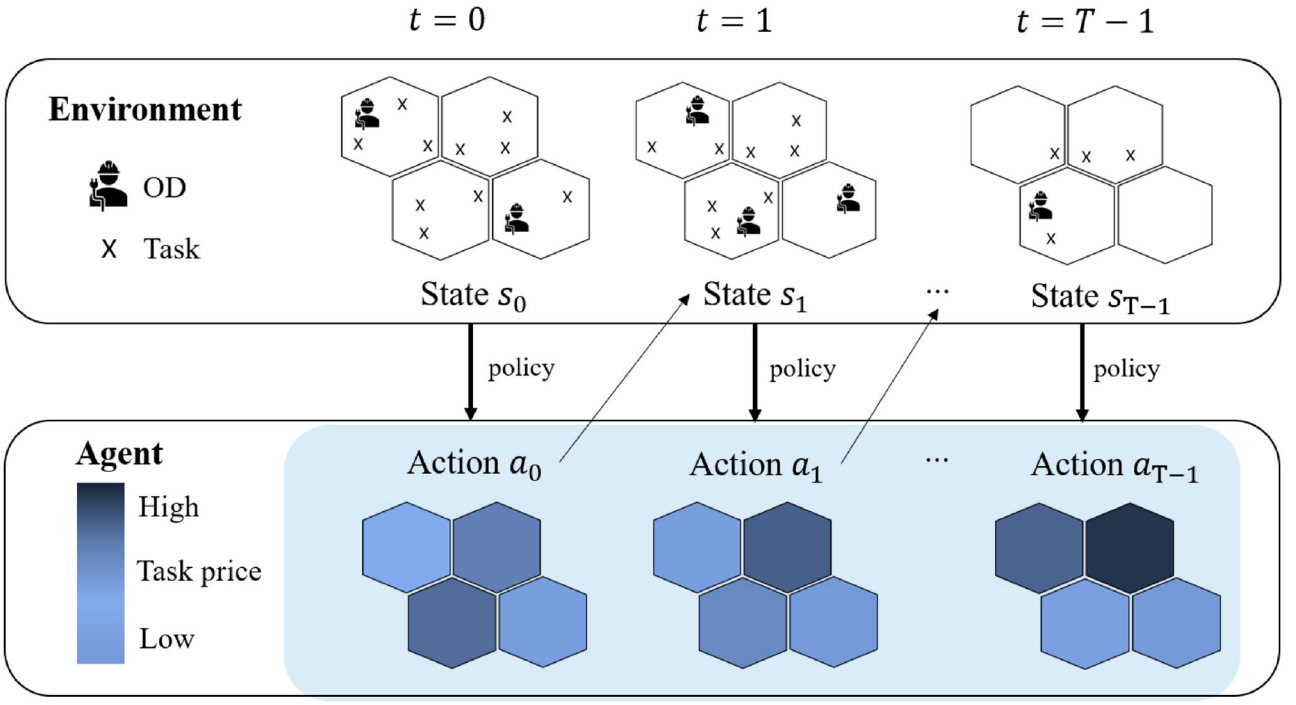


Figure 1. Overview of RL-based spatio-temporal task pricing algorithm.

operational cost, the sum of total OD wages and penalties for incomplete tasks. Figure 1 shows the overview of the spatio-temporal algorithm. The RL agent generates grid prices, while the environment simulates the real-world task reservations of ODs according to the price updates. For each time step, the agent observes the current state $\boldsymbol{s_t}$, representing the environment status. Then, according to the pricing policy $\pi(\boldsymbol{a_t} \mid \boldsymbol{s_t})$, which is a distribution of actions based on state, the RL agent samples the pricing action $\boldsymbol{a_t}$. This price update changes the environmental state, and the process repeats until the last step. As a model-free approach, the optimal pricing policy is learned through environmental interactions. Feedback on the previous actions is used to train the policy that consists of two deep neural networks. To the best of our knowledge, this study is the first attempt to apply a model-free reinforcement learning approach in the decision of spatio-temporal pricing on spatial crowdsourcing platforms, especially utilising ODs on SEM battery-swapping tasks.

The proposed RL-based algorithm effectively addresses the four challenges mentioned above. First, we utilise proximal policy optimisation (PPO) to determine the optimal policy within the continuous action space. The trial and error approach for exploring the actions and observing the consequences helps balance the trade-off outlined in the challenge (i). Second, RL is effective for sequential decision-making by considering the long-term rewards of the current action, addressing the challenge (ii). Third, the complex reservation process is handled inside the environment, and the agent merely recognises the outcome. This model-free approach lessens the burden pointed out in the challenge (iii). Fourth, once the learning process of RL algorithms is completed, they can rapidly generate actions for new input states (Hezewijk et al. 2023). Thus, it is practical to use in real-world scenarios, addressing the issue in the challenge (iv).

Computational experiments demonstrate that the proposed pricing algorithm resolves the spatio-temporal imbalance by strategically raising prices in OD shortage regions. The daily average reward successfully converges across large-scale episodes of a simulation environment designed based on real e-scooter data from Chicago. Comparing the cost obtained from a mixed integer programming (MIP) model with perfect information on ODs' task acceptance conditions, it is clear that PPO effectively learns near-optimal pricing policies even under uncertainty. Using the algorithm, a sensitivity analysis of key operational factors of the platform operation gives insights for algorithm training and system design.

The contributions of this study are threefold.

- This study presents an RL framework for spatio-temporal pricing for an SEM battery-swapping platform utilising spatial crowdsourcing. A real-world size environment simulates ODs' task reservations with a flexible working schedule, dynamic platform

participation, and heterogeneous task acceptance conditions. A continuous pricing policy based on the PPO algorithm is developed. Two feed-forward networks, actor and critic, are trained to generate pricing decisions through interactions within the simulation environment.

- A domain-specific masking technique is developed to enhance training efficiency, excluding information from inactive grids during the loss calculation. Inactive grids represent areas without tasks, hence where price updates are unnecessary. Experiments reveal the performance enhancement achieved by the masking technique in accelerating the algorithm.
- The performance of the PPO algorithm is demonstrated through computational experiments in both the solution quality and the computational time. Through the comparison with a MIP model with perfect information, the superiority of PPO among benchmark algorithms was evidenced in providing optimal solutions under the uncertainty of ODs' task acceptance. The algorithm shows consistent performance in real-world size episodes by resolving the spatio-temporal imbalance.

The remainder of this paper is structured as follows. Section 2 lists the previous studies relevant to our study with differentiation. Section 3 describes the problem statement, including the simulation environment. Section 4 introduces the framework and learning process of the proposed pricing algorithm based on PPO. Section 5 addresses the research questions of this study through computational experiments with managerial insights. Finally, Section 6 summarises the study.

2. Literature review

The success and sustainability of spatial crowdsourcing heavily depend on the compensation scheme (Guo et al. 2018). The most widely studied arena is the ride-sourcing platform. The main focus in the field has been the order dispatching problem, which involves assigning drivers to trip requests. Qin et al. (2020) discussed the evolution of this problem from a myopic combinatorial optimisation approach to deep reinforcement learning for managing large-scale, long-term optimisation. Meanwhile, pricing has become another critical issue, directly controlling trip request distribution and inducing driver repositioning, thereby leading to a more efficient matching of supply and demand. For spatial pricing, an equilibrium model optimises dual pricing strategies, balancing customer prices and driver wages (Bimpikis, Candogan, and Saban 2019; He and Max Shen 2015; Zha, Yin, and Xu 2018). Hu and Zhou (2020) introduce a fixed commission contract for on-demand matching platforms, where the wage is a fraction of the price. For dynamic pricing, various solution approaches, such as model predictive control (Nourinejad and Ramezani 2020) and a two-stage decision framework (Cachon, Daniels, and Lobel 2017) are used to increase the surplus of participants. Yang et al. (2020) designed a combined pricing scheme based on approximate dynamic programming using surge pricing during peak hours and a subsidy scheme during off-peak hours. Combining the features of both strategies, spatio-temporal pricing mechanisms were achieved. Ma, Fang, and Parkes (2022) developed a mechanism of subgame-perfect equilibrium that ensured drivers always accept their trip dispatches. Some studies tackle the high dimensionality by learning the pricing policy directly from the data, such as deep capsule network (He and Shin 2019) and RL (Chen et al. 2021). In particular, Chen et al. (2021) developed a two-sided optimal pricing policy, which increased the platform profits by 1.85 times, while dynamic pricing led to a 1.25 times increase in profits.

However, the SEM battery-swapping platform has unique characteristics. Different from leaving order demand in the ride-sourcing platform, in crowdsourced delivery platforms, tasks remain in the platform until they are served. Thus, the strategy evolves to assign tasks to specific ODs by minimising the cost. This becomes challenging when ODs' working schedules are flexible, and task preferences are uncertain. Existing research in this field typically considers the OD's willingness to work by requiring ODs to specify their conditions for task acceptance beforehand. These conditions include the distance or time flexibility (Archetti, Savelsbergh, and Grazia Speranza 2016, Macrina et al. 2020), which indicates how far a worker would detour from the original trip, and the stop willingness (Arslan et al. 2019), which restricts the number of accepted visiting sites by an OD. Usually, in these cases, the task price is a given parameter determined by factors, such as customer location (Archetti, Savelsbergh, and Grazia Speranza 2016; Gdowska, Viana, and Pedro Pedroso 2018), trip deviation of the OD's pre-planned trip (Archetti, Savelsbergh, and Grazia Speranza 2016; Arslan et al. 2019; Boysen, Emde, and Schwerdfeger 2022), and parcel size (Gdowska, Viana, and Pedro Pedroso 2018).

Simplified pricing strategies are used to stabilise workers' costs and streamline the optimisation of assignment decisions (Alnaggar, Gzara, and Bookbinder 2021). Nevertheless, much literature points out the necessity of exploring the dynamic pricing of OD wages. The initial steps were conducted by Gdowska, Viana, and Pedro Pedroso (2018) by correlating the wage with the willingness of ODs. Qi et al. (2018) included task pricing as a variable in their wage response model. Later,

Boysen, Emde, and Schwerdfeger (2022) considered the OD's minimum expected earnings per time unit as a compensation threshold, which is known to the platform. However, a more fine-grained pricing strategy is absent for platforms where task prices are predetermined and where ODs reserve tasks autonomously rather than being assigned to them.

Recent studies have concentrated on task pricing within spatial crowdsourcing, yet such studies remain constrained when applied to SEM battery-swapping platforms. Bai et al. (2022) introduced a dynamic pricing framework for a food-delivery platform based on a cooperative game-theoretic approach. A region-partition-based pricing scheme reduces the delivery fees in capacity overload regions and addresses entity recognition-based fine-grained pricing schemes that compensate couriers in capacity-insufficient regions. Although the objectives are similar to our study, adjusting initial pricing based on worker feedback is impossible in the SEM battery-swapping platform. Meanwhile, Liu et al. (2017) addresses the imbalanced task acceptance of workers in mobile crowdsensing platforms. The incentive mechanisms include a dynamic budget allocation algorithm followed by a price labelling algorithm. Rather than explicitly defining a task's completion potential, our approach directly determines the price required to motivate an OD based on spatio-temporal states. Tong et al. (2018) also accounts for task acceptance uncertainty by initially setting a base price from acceptance estimates, followed by dynamic adjustments. However, our study eliminates the need for such estimates while directly learning task acceptance by interacting with the environment. Fatehi and Wagner (2022) introduced a labour planning scheme for an on-demand crowdsourced delivery service that utilises a hybrid operation of 3PL firms and crowd drivers. What is different from our study is that Fatehi and Wagner (2022) derive an hourly wage to potentially recruit the minimum number of workers needed to complete all tasks. Our study focuses on attracting available workers each time with spatio-temporal per-task pricing.

To organise our contributions, our study fills the research gap of task pricing in spatial crowdsourcing in three ways: (i) it aims to maximise task completion by adjusting spatio-temporal imbalance due to dynamic OD participation; (ii) it directly learns the task acceptance conditions without worker feedback or estimation models; (iii) it provides real-time pricing solutions for problems with high dimensionality.

3. Problem statement

This research introduces a spatio-temporal pricing strategy for an SEM battery-swapping platform designed to resolve the imbalance between low-battery SEMs and occasional drivers (OD) responsible for swapping batteries. The strategy aims to minimise daily costs, which include the total expenditure for OD wages and penalties for unaccomplished tasks.

Table 1. Notations of RL-SCBS system.

$\mathcal{K}$	set of ODs $\{k = 1, 2, \ldots, K\}$
$\mathcal{N}$	set of SEM battery-swapping tasks $\{n = 1, 2, \ldots, N\}$
$\mathcal{H}$	set of grids $\{j = 1, 2, \ldots, H\}$
$\mathcal{T}$	set of time intervals $\{t = 1, 2, \ldots, T\}$
$\boldsymbol{a_t}$	action (grid price) at time step t, where $\boldsymbol{a_t} = [a_t^1, a_t^2, \ldots, a_t^H]$
$\boldsymbol{s_t}$	state at time step t
p_t^j	grid price of grid j at time step t, USD
$\boldsymbol{p_t}$	grid prices at time step t, where $\boldsymbol{p_t} = [p_t^1, p_t^2, \ldots, p_t^H]$
p^b	base task price, USD
p^u	the price upper bound set by the platform, USD
p^l	the price lower bound set by the platform, USD
c	penalty cost per task; USD
l^{ij}	required number of time steps to travel from grid i to j
l_{swap}	required number of time steps to finish a swapping task
z_t^{ij}	attractiveness of a task in grid j to an OD in grid i at time step t
q_k	capacity of OD k
α_k	willingness-to-accept (WTA) threshold value of OD k
d_t^j	number of tasks in grid j at time step t
o_t^j	number of idle ODs in grid j at time step t
$o_t'^j$	number of arriving ODs in grid j at time step t
w_t^j	number of reservations made in grid j at time step t

To model the real-world operational dynamics of the platform, an agent-based simulation environment, reinforcement learning for spatial crowdsourcing of battery swapping (RL-SCBS), is developed. Under pricing updates, it tracks the real-time interactions between ODs and the platform within the target city network.

The city is divided into H hexagonal grids with identical sizes in the environment. The total service time is split into T time intervals with equal duration (e.g. 5 minutes). In RL-SCBS, we assume that the time step required to travel to a neighbouring grid is 1. At each time step, the platform evaluates the current state $\boldsymbol{s_t}$ of the environment and determines regional prices as action $\boldsymbol{a_t}$. Tasks within the same grid are assigned the same price. ODs in the platform then respond to this action $\boldsymbol{a_t}$ by making decisions about reserving the tasks. In RL-SCBS, an episode indicates a daily process. The episode continues until a termination condition is met, either at the final time interval or earlier when all tasks are accomplished. Table 1 provides definitions of the variables involved in this simulation.

At the start of each simulation episode, a certain percentage of registered ODs willing to participate comes online. Each OD k has a capacity of q_k, indicating the number of swappable batteries received from city-wide battery distribution hubs. On a first-come, first-served basis, ODs reserve an SEM swapping task in a specific grid. Once an SEM battery is swapped, the OD earns the wage assigned to the SEM at time step t. Even if the

battery has not yet been swapped, tasks are considered completed if reserved before the service period ends. All the used batteries must be returned to hubs before the end of the service period.

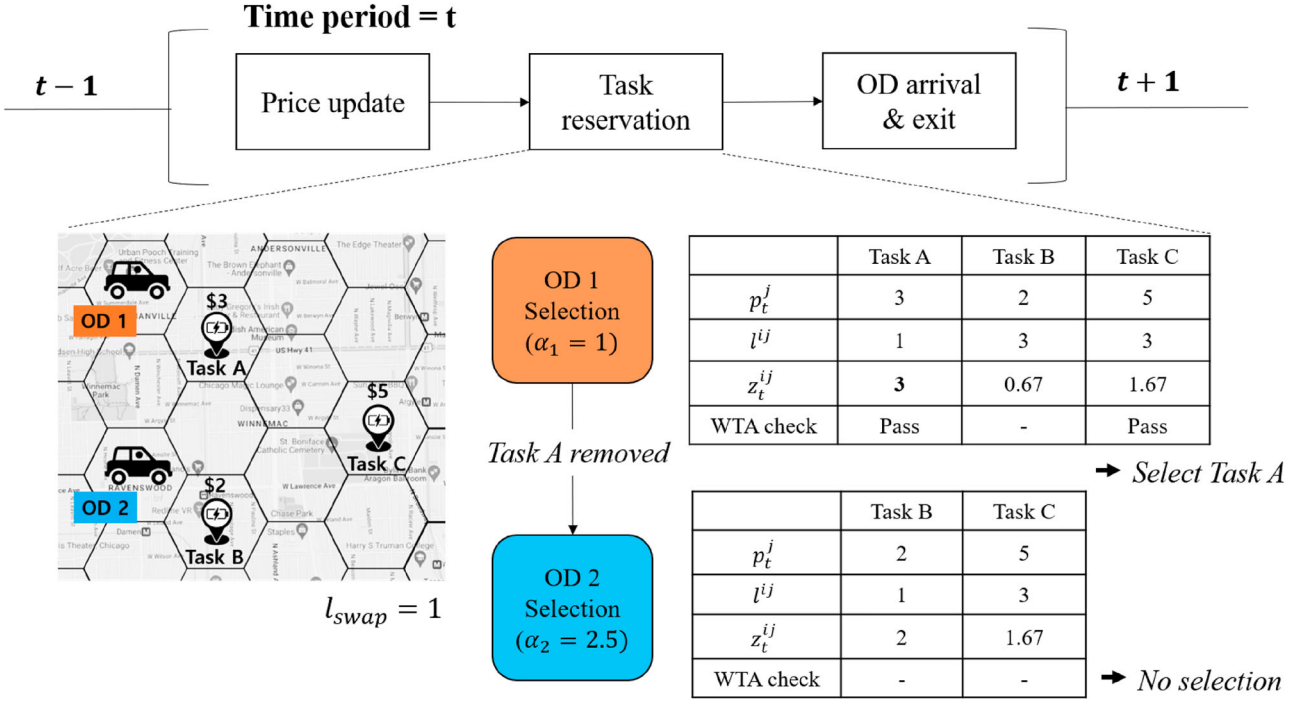
At every time step, an idle OD, either new to the platform or waiting for the next reservation, chooses one of the available options: make a new reservation, wait, or terminate. In case of making a reservation, an OD on grid i observes the price updates and selects the task in a target grid j based on price p_t^j and travel time, l^{ij}. In this study, ODs are assumed to prefer grids with higher prices for higher expected income. At the same time, the distance to the target grid is also influential, as longer distances require extra time and fuel. To balance these factors, the 'task (grid) attractiveness' concept is employed for modelling ODs' grid choices. Following the approach of Chen et al. (2021), it is expressed by the equation:

$$z_t^{ij} = \frac{p_t^j}{l^{ij}} \tag{1}$$

All tasks within the same grid have the same price; therefore, picking a task is the same as choosing a grid for an OD. Based on z_t^{ij}, each OD follows a two-step process for task reservation. The initial step involves filtering out grids that are not profitable. Many studies describe an independent contractor as setting a personal earnings threshold for accepting a task (Boysen, Emde, and Schwerdfeger 2022). This value represents the OD's willingness-to-accept (WTA), reflecting the minimum expected income per travelled time or distance. Tasks with attractiveness falling below the personal WTA are excluded from one's options. Next, ODs select the task with the highest attractiveness. This greedy decision-making stems from the competitive nature of the RL-SCBS system: the reservation process is on a first-come, first-served basis, and the available tasks on the platform gradually diminish over time, with no new low-battery tasks popping out, as SEM usage is assumed to be limited during the swapping periods.

Meanwhile, when an OD is dissatisfied with the current pricing, he or she may wait until the next update. This occurs when every candidate grid of an OD is filtered out. The OD remains idle in the subsequent time steps until a satisfactory alternative emerges.

Figure 2 illustrates the task selection of idle ODs in one time step. In practice, ODs are free to reserve at any time, but to prevent task overlap, it is assumed the platform prioritises ODs' reservation requests based on their order in the system queue. This results in OD1 being the first consideration. OD1 observes the remaining tasks and calculates the attractiveness of each, according to Equation (1). Assume $l_{swap} = 1$, which denotes the required time step for the swapping task. As the attractiveness of task 1 stands as the most appealing and surpasses the WTA threshold, α_1, OD1 opts to choose task A. The selected task A is then removed from the platform. Subsequently, OD2 evaluates the remaining tasks. As the highest attractiveness value of task C is less than the WTA



	Task A	Task B	Task C
p_t^j	3	2	5
l^{ij}	1	3	3
z_t^{ij}	**3**	0.67	1.67
WTA check	Pass	-	Pass

	Task B	Task C
p_t^j	2	5
l^{ij}	1	3
z_t^{ij}	2	1.67
WTA check	-	-

Figure 2. Illustration of task selection of ODs.

threshold, α_2, OD2 decides not to choose any task and to wait for the next update. Later, OD1 will be idle at time step $t+2$, because $l^{ij}=1$ and $l_{swap}=1$, while OD2 is still idle at time step $t+1$.

Finally, RL-SCBS considers dynamic OD participation. An OD exits the platform either upon depleting its capacity or after the scheduled duration has passed. New ODs keep entering the system throughout the entire service period, but no low-battery SEMs are created.

4. Proximal policy optimisation using a masking technique

The objective of the RL agent is to learn a policy π that maximises the total discounted rate.

$$R=\sum_{t=0}^{T}\mathbb{E}_{(\boldsymbol{s_t},\boldsymbol{a_t})\sim\pi}[\gamma^{t}r(\boldsymbol{s_t},\boldsymbol{a_t})] \tag{2}$$

The policy π is a mapping from states to actions, guiding the agent to take the most optimal pricing action $\boldsymbol{a_t}$ for each state $\boldsymbol{s_t}$ encountered. Optimizing π encompasses the following outlines.

4.1. Definitions of the Markov decision process (MDP) framework

The agent interacts with the environment by MDP components to infer the dynamics of the RL-SCBS system. The tuples of $(\boldsymbol{s_t},\boldsymbol{a_t},r_t,\boldsymbol{s_{t+1}})$ are saved for the learning process.

State

At each time step t, state $\boldsymbol{s_t}$ consists of the spatio-temporal status of ODs and tasks, denoted as $(\boldsymbol{d_t},\boldsymbol{o_t},\boldsymbol{o'_t},\boldsymbol{w_{t-1}})$.

- $\boldsymbol{d_t}$: Number of low-battery tasks in each grid at time t, where $\boldsymbol{d_t}=[d_t^1,d_t^2,\ldots,d_t^H]$
- $\boldsymbol{o_t}$: Number of idle ODs in each grid at time t, where $\boldsymbol{o_t}=[o_t^1,o_t^2,\ldots,o_t^H]$
- $\boldsymbol{o'_t}$: Number of arriving ODs in each grid at time t, where $\boldsymbol{o'_t}=[o_t'^1,o_t'^2,\ldots,o_t'^H]$
- $\boldsymbol{w_{t-1}}$: Number of reservations made in each grid at time $t-1$, where $\boldsymbol{w_{t-1}}=[w_{t-1}^1,w_{t-1}^2,\ldots,w_{t-1}^H]$

The state also includes a one-hot encoded time vector, forming the final state vector $\boldsymbol{s_t}\in\mathbb{R}^{4H+T}$.

Action

The action is defined as the regional price vector $\boldsymbol{a_t}=[a_t^1,a_t^2,\ldots,a_t^H]\in\mathbb{R}^H$, representing the additional reward price per grid. Therefore, the final regional price is calculated as $\boldsymbol{p_t}=\boldsymbol{a_t}+p^b\cdot\mathbf{1}_H$, where $\mathbf{1}_H$ is an H-dimensional vector with all elements equal to one. Additionally, elements of $\boldsymbol{p_t}$ lies within the range $[p^l,p^u]$.

Reward

The reward is the cost savings of battery-swapping tasks. It is designed to motivate completing as many tasks as possible at the lowest possible cost. To incentivize task completion, a penalty cost is imposed for any tasks remaining at episode termination. The reward function coefficient, η, balances the effect of the cost-savings and penalty terms during training. Based on the observation of the learning performance, we set η to 5.

$$r_t=\begin{cases}\sum_{j\in\mathcal{H}}\eta(p^u-p_t^j)w_t^j-c\sum_{j\in\mathcal{H}}(d_t^j-w_t^j), & \text{if } t=T\\ \sum_{j\in\mathcal{H}}\eta(p^u-p_t^j)w_t^j, & \text{otherwise}\end{cases}$$

4.2. Actor-critic framework

Seeking the optimal action in the RL-SCBS system becomes more complex as dimensionality increases with the growing number of grids (Pan et al. 2019). To tackle this complexity, this study adopts the actor-critic framework to design the policy architecture, known for its sample efficiency within high-dimensional environments (Yun et al. 2022). It consists of two deep neural networks; the actor generates the proper regional price based on environment states, and the critic evaluates the state. Figure 3 depicts the process of agents collecting data to train the policy through interactions with the environment. At each time step t, the process inputs the regional task price $\boldsymbol{p_t}$ into the environment for simulation. The simulation updates the system state $\boldsymbol{s_t}=(\boldsymbol{d_t},\boldsymbol{o_t},\boldsymbol{o'_t},\boldsymbol{w_{t-1}})$ and calculates the reward r_t, equal to the simulated cost-savings. Then, both the actor and critic networks utilise the state vector $\boldsymbol{s_t}$ as their input. The critic's output, $V(\boldsymbol{s_t})$, measures the potentials for the state yielding high expected returns, while the actor's output contains parameters for generating the action for the next time step. This pricing action then serves as the new input for the environment in the next cycle.

In Figure 3, the actor network generates a stochastic policy for continuous action space, expressed as $\pi(\boldsymbol{a_t}\,|\,\boldsymbol{s_t})$, where the policy is a probability distribution over actions $\boldsymbol{a_t}$ for a given state $\boldsymbol{s_t}$. Introducing randomness in action selection strikes a balance between exploring new actions and exploiting the best ones currently known, thereby preventing overfitting. The H output nodes of the actor become the action mean for each grid, while the H learnable parameter values are used as variance. These mean-variance pairs together establish a Gaussian distribution for each grid, from which the regional pricing action is sampled. The parameters are trained in a manner that increases the likelihood of sampling a specific price, deemed most likely to optimise the

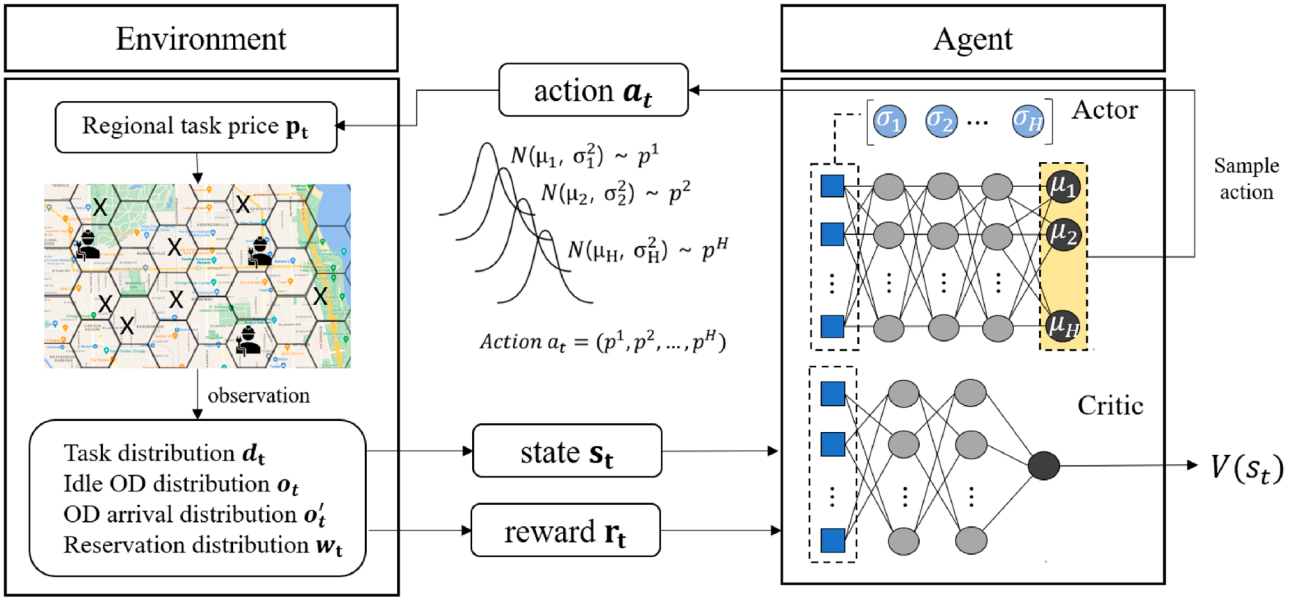


Figure 3. Actor-critic framework.

episode return. We use a hidden size of 256 for both the actor and critic network.

4.3. Optimization process with proximal policy optimisation (PPO)

To optimise a policy $\pi_\theta(\boldsymbol{a_t} \mid \boldsymbol{s_t})$, the most common approach involves fine-tuning the policy parameter θ, namely policy gradient (PG). This method aims to update θ in a direction that increases the probability of actions that lead to higher rewards and decreases the ones that lead to lower rewards. The process continues until the average episode return converges, signalling that the action sampled from the distribution is consistent and optimal. The general objective function of the PG method in which the agent takes a gradient ascent step is Equation (3).

$$L^{PG}(\theta) = \mathbb{E}_t \left[\log \pi_\theta(\boldsymbol{a_t} \mid \boldsymbol{s_t}) * A_t \right] \tag{3}$$

Here, A_t is the advantage function, determining how much better (or worse) taking a specific action is compared to the average action at that state.

However, the critical weakness of the PG method is that the policy faces extreme changes during updates, leading to unstable learning patterns. To address this drawback, the PPO algorithm constrains the policy update boundary for conservative updates of θ, with the clipped surrogate objective function as Equation (4).

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon)\hat{A}_t)] \tag{4}$$

$$r_t(\theta) = \frac{\pi_\theta(\boldsymbol{a_t}|\boldsymbol{s_t})}{\pi_{\theta_{\text{old}}}(\boldsymbol{a_t} \mid \boldsymbol{s_t})} \tag{5}$$

Equation (5) represents the probability ratio between the new and old policies. The condition $r_t(\theta) > 1$ indicates that for a given $\boldsymbol{s_t}$, the action $\boldsymbol{a_t}$ is more likely to be generated from the current policy than from the old policy, and the opposite holds for $r_t(\theta) < 1$. The key manipulation of PPO is to use the clip function to limit r_t between $[1 - \epsilon, 1 + \epsilon]$, with the hyperparameter $\epsilon = 0.2$ in our study. The policy avoids harmful large updates by taking the minimum of the clipped ratio and the unclipped ratio. The optimisation process utilises generalised advantage estimation (GAE), proposed by Schulman et al. (2015) to estimate $\hat{A}_t$, which is the discounted sum of temporal difference errors.

Entropy regularisation is implemented to encourage policy exploration. The agent is rewarded for generating actions that lead to higher rewards and for maintaining the randomness in its action selection.

The critic network evaluates the action quality of the actor network by outputting a state value function, $V_\phi(\boldsymbol{s_t})$, indicative of the expected return from the state $\boldsymbol{s_t}$. The critic updates its parameter ϕ to align $V_\phi(\boldsymbol{s_t})$ closely with the actual return estimate $\hat{R}$, achieved through minimising Equation (6),

$$L^{VF}(\phi) = \mathbb{E}_t \left[\left(V_\phi(\boldsymbol{s_t}) - \hat{R} \right)^2 \right] \tag{6}$$

where $\hat{R}_t = V_\phi(\boldsymbol{s_t}) + \hat{A}_t$.

4.4. Masking technique for valid action space

In the RL-SCBS system, pricing for the particular grid becomes unnecessary in two cases. The first is when no tasks are present at the beginning of the episode, and the second is when all tasks are completed, leaving no

remaining tasks. For such inactive grids, a PPO agent setting the price to zero can reduce unnecessary computational efforts.

The agent employs the following masking technique to handle the dynamic action space. At each time step, the agent observes the state to pinpoint inactive grids. Then, the agent skips sampling actions from the actor output for these inactive grids. Instead, the agent directly sets the regional price to zero. Subsequently, the values related to inactive grids, including the probability sum for sampled actions and the entropy sum of the action distributions, should be omitted from the loss calculation. Doing so prevents discrepancies between the actor's output distribution and the masked values, enhancing clarity and effectiveness in the learning process.

Algorithm 1 is the pseudocode for the total optimisation process with the PPO algorithm,

Algorithm 1 PPO with the masking technique.

Initialize actor policy π_θ, critic V_ϕ, the number of episodes M, batch $\mathcal{B}$, batch size B, the number of epochs K, discount factor γ , GAE parameter λ, Entropy coefficient β, actor learning rate α_A, critic learning rate α_C
for *episode* $= 1, 2, \ldots M$ **do**
  Initialize environment state
  while episode not terminated **do**
    Update time step t = t + 1
    Choose a_t and log probability $log\pi_{\theta_{old}}(a_t|s_t)$ by running policy $\pi_{\theta_{old}}(a_t \mid s_t)$
    for inactive grid j of time step t in $\{j \in H \mid d_t^j = 0\}$ **do**
      Set $a_t^j = 0$
      Subtract the log probability of grid j from $log\pi_{\theta_{old}}(a_t \mid s_t)$
    end for
    Record the trajectory $\tau = (s_t, a_t, r_t, s_{t+1})$ and $log\pi_{\theta_{old}}(a_t \mid s_t)$ in $\mathcal{B}$
    if $|\mathcal{B}| = B$ **then**
      Compute $\hat{A}$ for each episode in $\mathcal{B}$ by using GAE
        $\hat{A}_t = \delta_t + (\gamma\,\lambda)\delta_{t+1} + \cdots + (\gamma\,\lambda)^{T-t}\delta_{T-1}$,
          where $\delta_t = r_t + \gamma\, V_\phi(s_{t+1}) - V_\phi(s_t)$ and T= episode duration
      Compute $\hat{R}_t = V_\phi(s_t) + \hat{A}_t$
      for *epoch* $= 1, \ldots, K$ **do**
        for $t \in \mathcal{B}$ **do**
          Calculate $\log \pi_\theta(a_t \mid s_t)$, state value $V_\phi(s_t)$, and entropy H_t
          for inactive grid j in $\{j \in H \mid d_t^j = 0\}$ **do**
            Subtract the log probability of grid j from $\log \pi_\theta(a_t \mid s_t)$
            Subtract the entropy of grid j from H_t
          end for
          Calculate $r_t(\theta) = \exp(\log \pi_\theta(a_t \mid s_t) - \log \pi_{\theta_{old}}(a_t \mid s_t))$
        end for
        Calculate policy loss:
          $L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$
        Calculate value loss:
          $L^{VF}(\phi) = \mathbb{E}_t[(V_\phi(s_t) - \hat{R}_t)^2]$
        Update the agent policy parameters with the total loss function:
          $L = L^{CLIP} - 0.5L^{VF} + \beta\mathbb{E}_t[H_t]$
          $\theta \leftarrow \theta + \alpha_A \nabla_\theta L$
          $\phi \leftarrow \phi + \alpha_C \nabla_\phi L$
      end for
      Update θ_{old} to θ
      Reset batch $\mathcal{B}$
      Reset time step $t = 0$
    end if
  end while
end for

4.5. Mathematical formulation with perfect information

This section introduces a comparative model based on MIP to evaluate the PPO algorithm's ability to learn under uncertain OD task preferences. It is named 'spatio-temporal pricing under perfect information' (STP-PI). In contrast to PPO, the STP-PI model identifies the lowest payment required to encourage an OD by using exact threshold values for WTA.

The optimal solution of STP-PI serves as a lower bound for the PPO algorithm when two experimental conditions are met for the simulation environment: (i) workers do not enter or exit during the process, and (ii) workers have no capacity limits. We conduct experiments under these conditions, using the cost difference between STP-PI and PPO as the primary performance indicator.

The mathematical formulation of STP-PI is as follows:

Decision variables

$$p_t^j = \text{grid price at grid } j \text{ at time step } t \qquad \forall j \in \mathcal{H}, \forall t \in \mathcal{T}$$

$$x_{kt}^{ij} = \begin{cases} 1, & \text{if driver } k \text{ on grid } i \text{ reserves task on grid } j \text{ at time step } t. \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i,j \in \mathcal{H},\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T}$$

$$s_{kt}^{j} = \begin{cases} 1, & \text{if driver } k \text{ is idle or arriving at grid } j \text{ at time step } t. \\ 0, & \text{otherwise.} \end{cases} \qquad \forall j \in \mathcal{H}, \forall k \in \mathcal{K},\ \forall t \in \mathcal{T}$$

$$\min \sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{H}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} p_t^j \cdot x_{kt}^{ij} + c \cdot \left(N - \sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{H}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} x_{kt}^{ij} \right) \tag{7}$$

$$\text{s.t.} \quad p^l \le p_t^j \le p^u, \quad \forall j \in \mathcal{H},\ \forall t \in \mathcal{T} \tag{8}$$

$$\sum_{j\in\mathcal{H}} x_{kt}^{ij} \le 1, \quad \forall i \in \mathcal{H},\ \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{9}$$

$$\sum_{i\in\mathcal{H}} x_{kt}^{ij} \le 1, \quad \forall j \in \mathcal{H},\ \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{10}$$

$$\sum_{i\in\mathcal{H}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} x_{kt}^{ij} \le d_1^j, \quad \forall j \in \mathcal{H} \tag{11}$$

$$M_k^1(x_{kt}^{ij} - 1) \le \frac{p_t^j}{l^{ij}} - \alpha_k, \quad \forall i,j \in \mathcal{H},\ \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{12}$$

$$M_k^2(x_{kt}^{ij} - 1) \le \frac{p_t^j}{l^{ij}} - \frac{p_t^q}{l^{iq}}, \quad \forall i,j,q \in \mathcal{H},\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \tag{13}$$

$$\sum_{j\in\mathcal{H}} s_{kt}^j = 1, \quad \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \tag{14}$$

$$\sum_{j\in\mathcal{H}} x_{kt}^{ij} \le s_{kt}^i, \quad \forall i \in \mathcal{H},\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \tag{15}$$

$$s_{kt}^i \le 1 - \sum_{p\in\mathcal{H}, p\neq i}\sum_{j\in\mathcal{H}} x_{kt}^{pj}, \quad \forall i \in \mathcal{H},\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \tag{16}$$

$$\sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{H}} x_{kt'}^{ij} \le 1 - x_{kt}^{ij}, \quad \forall i,j \in \mathcal{H}, \forall k \in \mathcal{K},\ \forall t \in \mathcal{T},\ \forall t' \in \mathcal{T}'_{ijt} \setminus \{t + l^{ij} + l_{swap}\} \tag{17}$$

$$x_{kt}^{ij} \le s_{kt'}^j, \quad \forall i,j \in \mathcal{H},\ \forall k \in \mathcal{K},\ \forall t \in \mathcal{T},\ \forall t' \in \mathcal{T}'_{ijt} \tag{18}$$

$$s_{kt}^i - \sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{H}} x_{kt}^{ij} \le s_{k,t+1}^i \le s_{kt}^i + \sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{H}} x_{kt}^{ij}, \quad \forall i \in \mathcal{H}, \forall k \in \mathcal{K},\ \forall t \in \mathcal{T} \setminus \{T\} \tag{19}$$

The objective function (7) minimises the overall daily cost, the same as PPOwM. Constraint (8) bounds the price values. Constraints (9) and (10) prohibit multiple movements of each OD by time. According to Constraint (11), the total number of reservations cannot exceed the initial quantity of tasks available. Constraints (12) and (13) represent the conditions for an OD's task selection based on their WTA and greedy behaviour, where M indicates a sufficiently large number. Constraint (14) limits the given OD's location to a single, specified grid. Constraints (15) and (16) specify that an OD must stay in a particular grid to make a specific reservation.

Let $\mathcal{T}'_{ijt}$ be the set of time steps as follows.

$$\mathcal{T}'_{ijt} = \{t+1, \ldots, \min(t + l^{ij} + l_{swap}, T)\}, \quad \forall i,j \in \mathcal{H},\ \forall t \in \mathcal{T}$$

Once an OD makes a reservation, it is not permitted to make further reservations until completion, as specified in Constraint (17). Additionally, the arriving grid changes to the corresponding grid until it becomes idle, as described in Constraint (18). At last, Constraint (19) interconnects the two decision variables. The bilinear term $p_t^j \cdot x_{kt}^{ij}$ in Equation (7) is linearised by using the technique in MirHassani and Hooshmand (2019).

5. Computational experiments

This section presents the results of three distinct computational experiments. Each experiment addresses the following research questions:

(i) How does the solution quality differ between the PPO algorithm-based pricing strategy and the optimisation-based models in minimising the cost?
(ii) Does the PPO algorithm effectively resolve spatio-temporal imbalance in real-world scenarios with dynamic and uneven OD engagements?
(iii) How do key factors in RL-SCBS impact the performance of the PPO algorithm and platform operation costs?

Research questions (i), (ii), and (iii), are answered by Experiments 1–3, respectively. All experiments are conducted on a workstation with an Intel Pentium CPU G3250 @ 3.20GHz. Optimization models are solved by Python API for ILOG CPLEX solver 12.10.

5.1. Experiment 1: algorithmic performance

This experiment evaluates the performance of the spatio-temporal pricing algorithm in minimising the daily costs, even without the perfect information on ODs' WTA thresholds. Benchmark algorithms include advantage actor-critic (A2C), PPO, PPO with the masking technique (PPOwM), and STP-PI. Tests are conducted on small-scale episodes with $H = 5 \times 5$ and $T = 12$. We outline two scenarios based on the N/K ratio, calculated as the number of tasks (N) divided by the number of ODs (K) in an episode.

In Scenario *S1*, the N/K ratio is 6.66 ($K = 3$ and $N = 20$), where task completion of 100% within the T time steps is feasible. Conversely, completing all tasks within the allotted time is impossible in *S2*, with an N/K ratio of 10 ($K = 3$ and $N = 30$). For simplicity, we assumed that ODs neither enter nor exit the platform during the simulation and that the capacity is sufficient. The WTA threshold is set at 5 for all ODs, which requires a cost of \$5 per grid distance to attract an OD to reserve the task. As the price range is set to [\$0, \$20], an OD can be directed to a maximum of four grids per reservation.

Figure 4 compares the learning process of three RL algorithms, A2C, PPO, and PPOwM by scenarios. Batch sizes for PPO(wM) and A2C were 4,800 and 600, with learning rates of 0.0001 and 0.00025, respectively. The x-axis represents the time steps used in the training, and the y-axis represents the metric variation across episodes. Figure 4(a) shows that it took 3.3×10^7 time steps for PPOwM to converge in Scenario *S1*, while PPO and A2C were still improving then. Table 2 summarises the converged metrics values. According to Table 2, PPO required 1.8 times as long, and A2C needed more than 2.6 times as long to converge compared to PPOwM. In Scenario *S2*, it took larger time steps for convergence, specifically 3.9×10^7 for PPOwM. For the converged episodic cost, PPOwM showed the best performance, with values of 116.11 and 237.55 for Scenarios *S1* and *S2*, respectively, followed by PPO and A2C.

For further investigation into how close this value was to the optimal, we compared it to the results of STP-PI. As a performance measurement, an efficiency gap is defined as follows:

$$\text{Efficiency gap} = \left(1 - \frac{W_{\text{STP-PI}} - \text{Solution of algorithm}}{W_{\text{STP-PI}} - O_{\text{STP-PI}}}\right) \times 100\%$$

where $W_{\text{STP-PI}}$ and $O_{\text{STP-PI}}$ are the worst and optimal objective values of STP-PI, respectively. The efficiency gap measures the algorithm's ability to enhance the solution toward STP-PI's optimal outcome relative to the worst case. $O_{\text{STP-PI}}$ is gained by the CPLEX solver, and $W_{\text{STP-PI}}$ is calculated as the total penalty cost in cases where no tasks are completed ($p^u = c$). We examine three algorithms: PPOwM, A2C, and m-STP-PI. Here, m-STP-PI refers to an adaptation of STP-PI, where the STP model is solved using a rolling horizon approach for each immediate time interval.

Table 3 presents the efficiency gap of each algorithm across episodes with different initial distributions of ODs and tasks. The results of RL algorithms show the mean and standard deviation of the gap from 1,000 runs of the optimal policy on the target episode. Although a positive efficiency gap is unavoidable due to the sampling process, the results demonstrate that PPOwM performs reasonably well.

In both scenarios, PPOwM was closest to the optimal value of STP-PI on average, with an efficiency gap of 4.94 and 11.03, respectively. Interestingly, this is smaller than the efficiency gap of m-STP-PI. From this, we infer the superiority of PPOwM in sequential decision-making

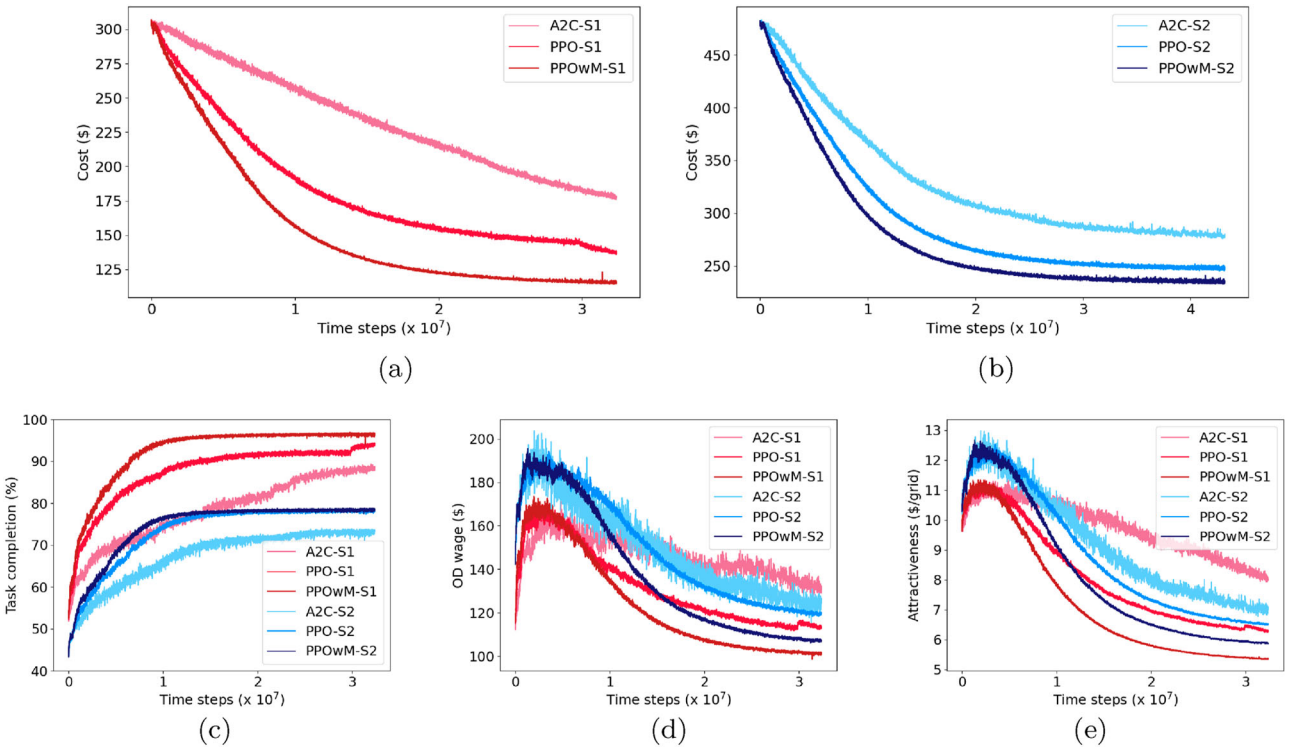


Figure 4. Convergence process over updates in Scenarios *S1* and *S2*. (a) Cost of Scenario *S1*. (b) Cost of Scenario *S2*. (c) Task completion. (d) Total OD wage. (e) Task attractiveness.

Table 2. Convergence values by simulation metric.

	Time steps (#)		Cost ($)		Task completion (%)		OD wage ($)		Attractiveness ($/grid)	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
A2C	8.9×10^7	5.8×10^7	132.72	263.03	95.8	76.0	115.13	119.56	6.15	6.46
PPO	5.3×10^7	5.6×10^7	124.56	246.45	94.3	78.1	102.83	116.05	5.57	6.27
PPOwM	**3.3×10^7**	**3.9×10^7**	**116.11**	**237.55**	**96.1**	**78.1**	**101.96**	**105.21**	**5.37**	**5.75**

Table 3. Comparison on efficiency gap(%).

Episode	*S1*: $K = 3, N = 20$			*S2*: $K = 3, N = 30$		
	m-STP-PI	A2C*	PPOwM*	m-STP-PI	A2C*	PPOwM*
1	13.13	12.25; 1.96	6.25; 3.29	10.39	17.72; 1.54	12.95; 3.19
2	8.87	4.95; 0.50	4.89; 1.19	17.28	14.49; 2.37	7.67; 3.66
3	6.90	7.77; 2.70	5.18; 1.90	13.66	10.00; 1.96	13.48; 1.78
4	9.14	7.60; 1.21	3.11; 0.90	7.05	9.69; 2.01	7.13; 2.19
5	5.37	3.94; 0.65	3.92; 1.29	9.88	24.25; 2.06	10.83; 2.83
6	11.17	5.83; 2.57	3.18; 1.54	15.34	6.84; 2.71	7.63; 3.36
7	4.81	5.74; 1.14	2.37; 0.88	19.76	13.27; 1.14	10.19; 2.43
8	5.00	7.83; 1.46	5.88; 1.16	15.03	15.53; 1.97	9.73; 6.24
9	3.94	6.96; 1.43	4.10; 1.10	9.09	11.96; 2.44	7.86; 2.31
10	2.99	3.07; 0.61	4.86; 0.83	23.08	8.56; 1.61	9.13; 2.04
11	10.50	4.61; 0.43	3.93; 1.26	10.30	13.37; 1.51	21.89; 2.36
12	6.97	3.83; 0.82	2.83; 1.47	10.49	12.86; 3.63	16.14; 8.24
13	0.98	10.62; 3.25	3.98; 1.36	16.05	19.61; 1.77	12.15; 7.08
14	5.05	14.47; 1.14	6.36; 2.07	19.87	22.99; 0.89	10.14; 3.78
15	1.51	2.99; 0.42	5.35; 7.35	0.65	15.83; 7.46	13.18; 3.13
16	14.07	5.80; 0.80	4.25; 1.67	12.88	32.80; 9.39	8.97; 3.20
17	0.50	4.29; 0.73	3.33; 0.82	10.46	17.73; 1.62	9.22; 3.07
18	14.21	8.66; 1.63	9.41; 2.58	7.14	17.27; 1.57	12.48; 3.19
19	8.79	8.12; 1.52	7.33; 1.76	4.76	22.74; 1.62	11.04; 1.72
20	13.64	7.78; 1.75	8.26; 0.78	13.73	16.92; 1.98	8.88; 2.10
Average	7.38	6.86	**4.94**	12.34	16.22	**11.03**

Note: *Mean; standard deviation of 1,000 runs.

even in the absence of perfect information on α_k. It suggests that PPOwM is capable of learning to navigate ODs across the entire duration of an episode to minimise overall cost. However, m-STP-PI minimises the cost of the immediate time step, which lacks long-term planning despite perfect information. When comparing scenarios, the performance gap of RL algorithms is smaller in Scenario *S1* than in *S2*. This suggests that PPOwM performs better when 100% task completion is achievable. In the subsequent experiment, we explore the underlying causes of this outcome.

Figure 4 includes the observation of other simulation metrics along training, involving task completion rate, overall OD wage, and task attractiveness averaged across episodes. Each metric converges to values listed in Table 2. In Figure 4(c), the task completion near 100% is achieved for Scenario *S1*, whereas it is around 78.1% in Scenario *S2*. The stable convergence of task completion is observed. Meanwhile, Figure 4(d) shows an initial rise followed by a steady decline in the average OD wage. This phenomenon implies that the RL agents initially offer high rewards to find the optimal OD directions for the entire time step to increase task completion. Then, the agent lowers the wage to find the minimum possible price to entice ODs. This fact is also evident in Figure 4(e), where the averaged attractiveness of task reservations soared to about 11 in Scenario *S1* and to 12.5 in *S2*, and then gradually decreased. It explains why Scenario *S2* required more time steps until convergence than did Scenario *S1*. Consistently, PPOwM shows the lowest in task attractiveness. This observation suggests that PPOwM investigated a pricing policy that compensates ODs minimally, offering payments closest to an OD's WTA threshold among the algorithms evaluated. The convergence value of 5.372 in Scenario *S1*, lower than 5.752 in Scenario *S2*, indicates the facilitated exploration in OD-sufficient scenarios.

To summarise, PPOwM outperforms other algorithms in both the learning speed and the solution quality. A2C exhibits the weakest performance, highlighting the benefits of PPO's conservative learning approach. Moreover, the masking technique accelerated the performance of PPO by filtering out the information from inactive grids during the update process. Finally, optimal convergence conditions are evidenced by achieving the highest task completion rates, with task attractiveness nearly matching the WTA threshold.

5.2. Experiment 2: application of PPOwM to a real-world RL-SCBS system

This experiment examines the implementation of the PPOwM algorithm in the RL-SCBS system. We addressed practical instance scales of H, T, N, and K. In particular, ODs' dynamic engagement with heterogeneous WTA thresholds and working hours are considered.

Data used for designing the RL-SCBS system is summarised in Figure 5. The simulation environment is modelled after a shared electric scooter platform in Chicago (Chicago data portal 2020). The area with the highest scooter usage is chosen as the target area of RL-SCBS. It encompasses four community areas of Chicago, which we refer to as Zone in Figure 5(a). The target area spans 29.37 km^2, bounded by latitudes 41.889°N to 41.976°N and longitudes −87.675°W to −87.615°W. We divide this into 14 × 5 number of hexagonal grids with a side length of 0.49 km. The total battery-swapping time of 6 hours between 10:00 p.m. and 4:00 a.m. is divided into 72 time steps, where one time step indicates 5 minutes.

The generation of task and OD data is outlined. To utilise realistic task distribution, a daily trip simulation was carried out based on an initial scooter deployment to identify scooters with low battery levels. The simulation covered September 1, 2020, to October 31, 2020, incorporating 172,202 usage records. Since the dataset provided the origin-destination pairs by community area, exact scooter locations within each area were randomly determined. Battery consumption for each trip was calculated based on usage distance using a linear regression model developed by Pender, Tao, and Wikum (2020). Scooters with battery levels falling below 80% by 10:00 p.m., a time when trip activity typically decreases, were marked for battery-swapping. The statistics revealed a weekly pattern in the task volume, as depicted in Figure 5(c).

On the supply side, the size and spatio-temporal distribution of the OD fleet significantly impact the simulation results. We assume that the times when taxi drivers and ODs engage with the platform will exhibit similarity. We refer to the Chicago taxi trip dataset to ground this assumption in real-world data (Google cloud 2017). A distinct weekly trend emerges, as illustrated in Figure 5(d), with peaks on Fridays and Saturdays. Let the fleet size rate indicate the number of workers temporally engaged in the platform divided by the total registered number. The hourly fleet size distribution of the dataset reveals a regular pattern, showing that most OD engagements conclude before midnight, detailed in Figure 5(e). The temporal OD engagement size in the RL-SCBS is generated based on this hourly fleet size rate multiplied by the fixed registered OD size. The engagement locations for ODs are generated randomly based on the population distribution data of Chicago, shown in Figure 5(b). Detail parameter values of the RL-SCBS system are summarised in Table 4.

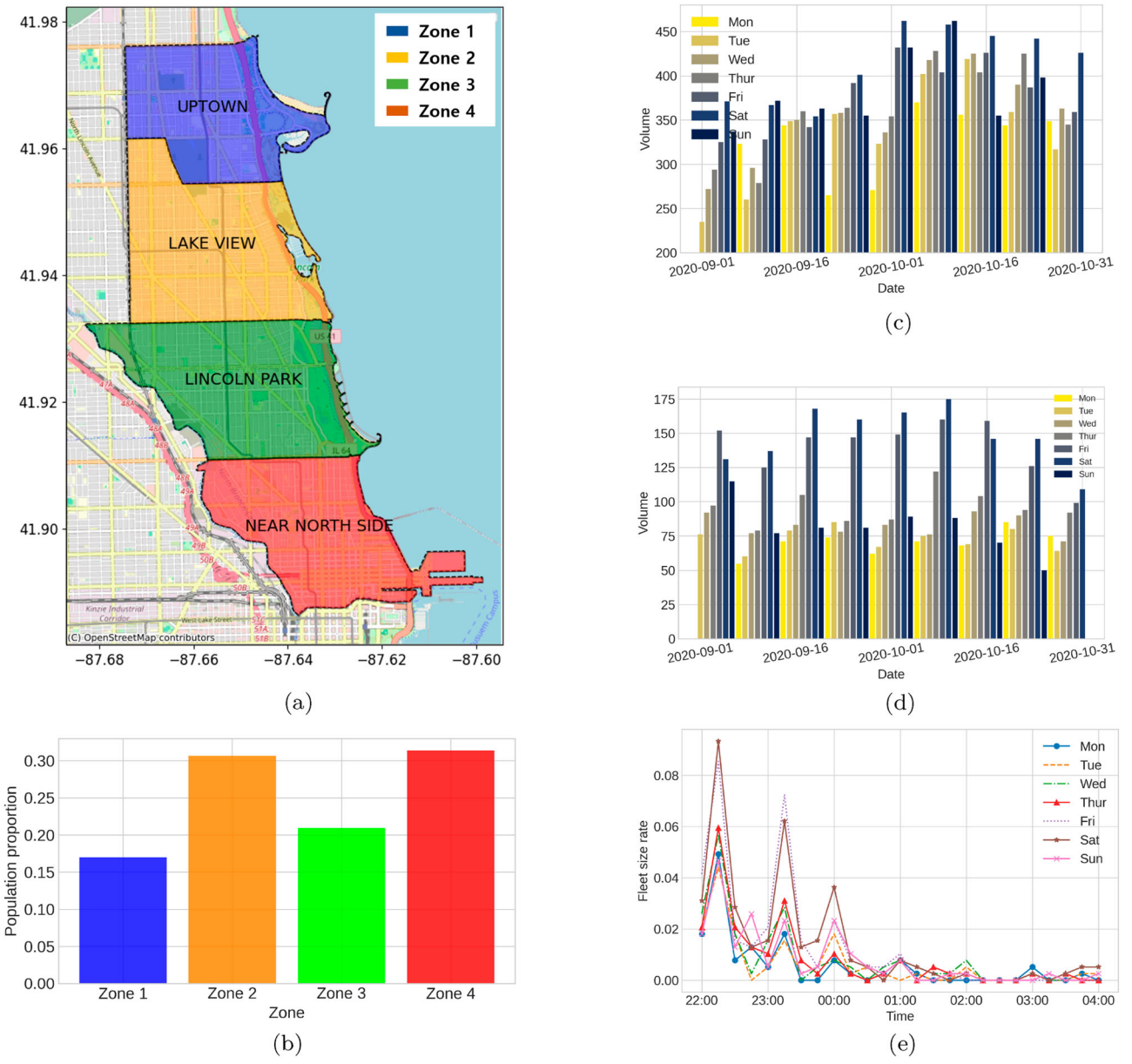


Figure 5. Data used in the RL-SCBS system. (a) Target area by Zone. (b) Population proportion. (c) Volume of low-battery e-scooter by date. (d) Driver fleet size by date. (e) Hourly fleet size rate by weekday.

Table 4. Parameters of RL-SCBS system.

H	14 × 5
T	72
N	[234, 456] (see Figure 5(c))
K	[20, 82]
p^b	2
p^l	2
p^u	6
c	6
q_k	20
α_k	$N(0.5, 0.1)$
OD working time steps	$N(36, 6)$
Registered OD size	200

Figure 6 shows the converge process. In Figure 6(a), the distribution of the N/K ratio of train episodes is sorted by weekdays. The N/K ratio is lower on Fridays and Saturdays as the OD fleet size is larger than on other days. For episodes with varying N/K ratios, the system achieved convergence after 2×10^7 time steps. A batch size of 7200 and a learning rate of 0.0001 is used. At the convergence point, the task completion rate maintains a maximum level, and the average attractiveness per reservation stabilises near the WTA threshold.

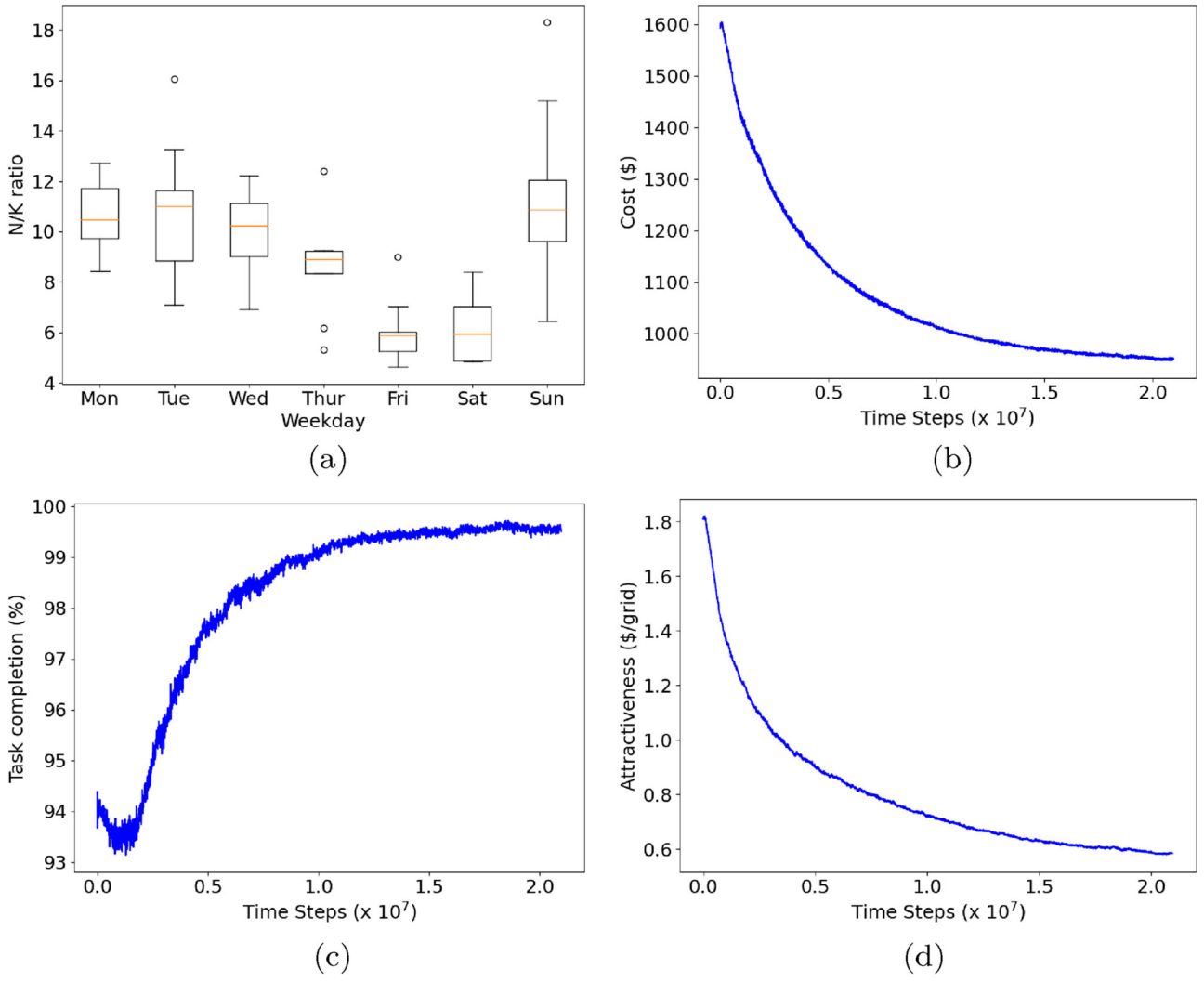


Figure 6. Convergence process in RL-SCBS system. (a) *N/K* ratio by weekdays. (b) Cost. (c) Task completion. (d) Task attractiveness.

We examine the performance of PPOwM on 30 test episodes not used in the training. Table 5 summarises the simulation result, with each episode averaged over 100 iterations. The episodes are divided into five groups according to the N/K ratio. Larger group numbers indicate higher average N/K ratios. The average number of tasks and ODs are presented. Task quantities showed no clear trend, whereas there was a noticeable decrease in OD size as the average group ratio increased. The OD size of Group 1 was 2.26 times bigger than that of Group 5. Due to the smaller OD size, task completion decreases as the average N/K ratio increases. At the same time, the average episode length is longer, which indicates the average time step taken for episode termination.

Table 5 reveals that the number of tasks influences the total cost (e.g. it usually costs more to complete a larger number of tasks). The reason is analysed from two perspectives: the proportion of penalty costs relative to the total cost and the average task price. First, a higher N/K ratio leads to a higher percentage of penalty cost, aligning with the low task completion rate. Second, the accept price, representing the average task price at which ODs accepted tasks, increased along the N/K ratio. We infer that the PPOwM strategically increased the task price to lower the penalty cost.

This observation is also evident in the average task attractiveness trend, increasing by the N/K ratio. In the case of group 1 with relatively many ODs, setting a modest acceptance price–allowing only ODs with α_k values close to the mean to accept tasks–resulted in high completion rates. Consequently, the average task attractiveness for ODs to reserve tasks is close to the mean value of α_k, which is 0.5. However, in group 5, with a scarcity of ODs, task prices tend to rise, as it is necessary to satisfy

Table 5. Simulation metrics of episodes grouped by *N*/*K* ratio.

Group	*N*/*K* ratio[a]	Task (#)	OD (#)	Task completion (%)	Episode length (#)	Total cost ($)[b]	Accept price ($)	Attractiveness ($/grid)
1	5.51; 0.51	366.50	66.80	99.9	44.26	928.24; 0.2	2.53	0.53
2	7.12; 0.60	376.50	53.49	99.9	47.08	985.78; 0.3	2.62	0.59
3	8.76; 0.70	326.67	37.37	99.4	52.03	864.70; 1.2	2.63	0.61
4	10.39; 0.61	357.67	34.48	98.1	60.86	990.02; 3.7	2.70	0.65
5	12.67; 0.76	373.67	29.54	97.2	65.27	1067.15; 5.8	2.75	0.67

Note: [a]Mean of *N*/*K* ratio values by groups; standard deviation of *N*/*K* values by groups.
[b]Total cost; percentage of penalty cost.

Table 6. Simulation metrics of three episodes A, B, and C.

	N/*K* ratio	Task (#)	OD (#)	Task completion(%)	Episode length (#)	Total cost ($)[a]	Accept price ($)	Attractiveness ($/grid)
A	9.62	356	37	100	51	909.40; 0	2.55	0.55
B	10.97	373	34	94	71	1189.53; 1.1	3.01	0.85
C	7.56	340	45	100	52	915.28; 0	2.69	0.60

Note: [a]Total cost; percentage of penalty cost.

ODs with higher α_k values to reduce penalty costs. As a validation of this inference, a higher task attractiveness is observed as the N/K ratio increases.

Next, we examine the spatio-temporal problem-solving performance of PPOwM in terms of its pricing actions and corresponding OD movements and task reservations. The experiment dealt with three representative episodes selected by two factors: uniformity in the regional engagement of ODs and task completion. Episode A represents the case where ODs' initial platform engagement is uniform across regions, with the eventual task completion close to 100%. Episode B depicts a case with uneven initial OD engagement across regions, resulting in a low task completion rate. Finally, Episode C outlines a case where, despite an uneven initial engagement of ODs across regions, task completion is 100%. Table 6 summarises the simulation conditions and resulting performance metrics for the experiments.

Figure 7 illustrates the regions where ODs initially participated and made reservations throughout the episode. The x-axis indicates the time step, while the y-axis indicates the number of OD fleets and the number of reservations made, respectively. Figure 7(a) shows a relatively uniform distribution of ODs in Episode A, and Figure 7(d) shows that reservations were also made evenly in Zones, accordingly. Due to the widespread and even presence of idle ODs across the Zones, PPOwM does not have to direct ODs to specific Zones with lower supply.

Conversely, Figure 7(b) exhibits a significant concentration of ODs' initial engagement in Zone 1 for Episode B. Higher prices are needed to motivate ODs to move longer distances to facilitate task completion in Zone 4. Initial observations from Figure 7(e) indicate an absence of reservations in Zone 4. Yet, PPOwM induced ODs in Zone 4 later on, even though it encountered a penalty cost in the end, owing to the initial shortage of ODs.

Meanwhile, Figure 7(c) displays a skewed initial engagement in Zone 2 for Episode C. Nonetheless, the task completion becomes 100%. This notable outcome compared to Episode B arises for two main reasons. First, Episode C features fewer tasks and more ODs, simplifying the completion of all tasks. Second, the proximity of Zone 2 to Zone 4 facilitates more efficient OD movement within the established price range. This inference is evidenced by Figure 7(f), where tasks in Zone 4 start to see reservations during the later stages of the episode. Observing the accepted price of Episode B in Table 6, we can see that the PPOwM needed to set higher prices to direct ODs toward Zone 4 rather than toward other Zones. This observation aligns with the observed increase in the average attractiveness for ODs in Table 6.

For a more intuitive understanding, Figure 8 visualises the pricing actions of PPOwM throughout Episodes A, B, and C for 14 × 2 selected neighbouring grids. The x-axis represents the given episode's time steps, where the black triangle on the x-axis marks the time step at which the grid becomes inactive. The dual y-axes show distinct information. The left y-axis, marked with blue line plots with error bars, outlines the action (mean as line and standard deviation as error bars) made by PPOwM. Each point represents the mean, with the error bars indicating the standard deviation of the action distribution sampled for pricing. A smaller error bar suggests that the policy has effectively learned to consistently sample similar prices. The right y-axis, marked with red bar plots, captures the OD density within a grid. The OD density measures how densely ODs are distributed within a grid and its surrounding grid, taking into account neighbouring grids with distances up to 4. Using the indicator function $\mathbf{1}[A]$ representing 1 when A is true and 0 otherwise, the OD density in the target grid is calculated as follows:

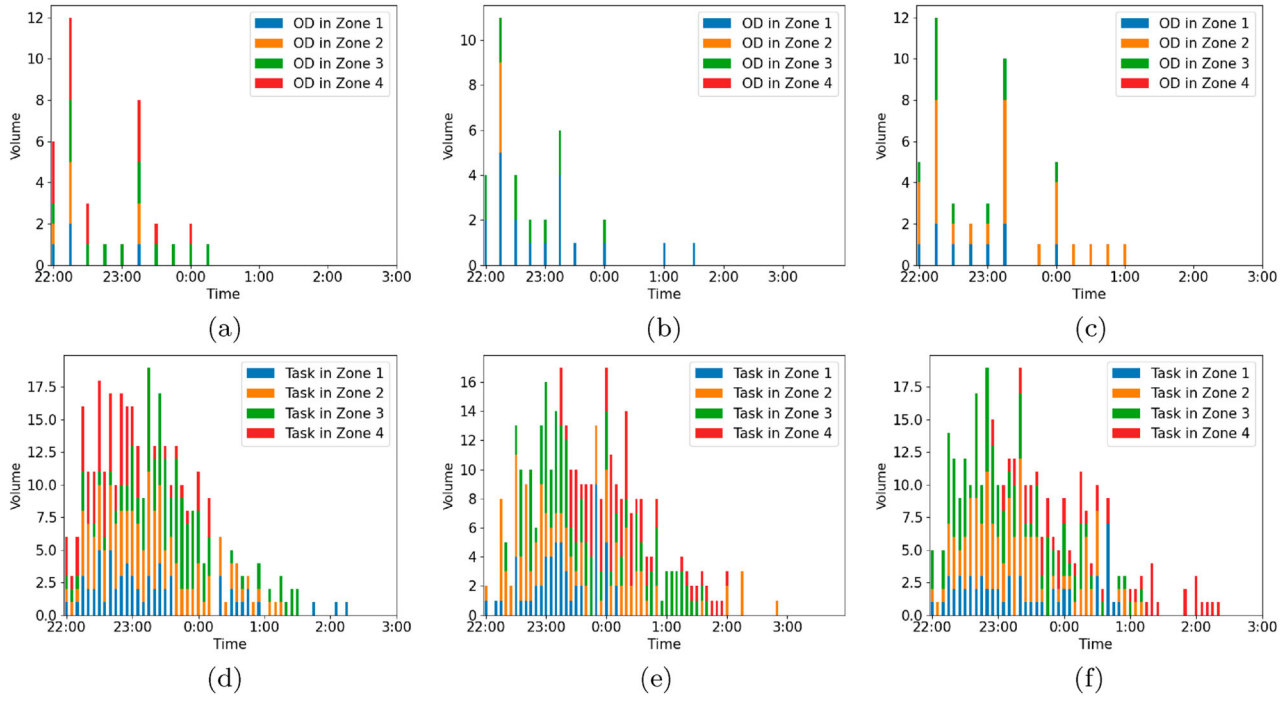


Figure 7. OD engagement and reservation by Zone. (a) OD engagement in Episode A. (b) OD engagement in Episode B. (c) OD engagement in Episode C. (d) Task reservation in Episode A. (e) Task reservation in Episode B. (f) Task reservation in Episode C.

$$\text{OD density of grid } i \text{ in time step } t = o_t^i + \sum_{j \in \mathcal{H}, i \neq j} o_t^j \times \left(\frac{1}{e}\right)^{l^{ij}} \times \mathbf{1}[l^{ij} \leq 4]$$

The main discovery highlights the connection between pricing decisions and OD density. Figure 8(a) shows a relatively consistent OD density across grids, with prices staying near the base level. A notable price increase occurs in Grid 10 at the end, which aims to attract nearby ODs. In episode B, such price surges are more frequently observed due to the uneven distribution of OD densities. Early on, Grids in Zone 1 become inactive with sufficient ODs, while price surges occur in OD-insufficient grids. Observe the price surges near the price maximum in Grids 57 and 58. For Grids 25, 48, and 52, tasks remain uncompleted, where no nearby ODs are available to be attracted within the current price range. In such cases, it is observed that prices are not raised to the maximum. Instead, a moderate pricing strategy is maintained, considering the potential for new ODs to join the platform. Interestingly, the intensity of price surges is influenced by the temporal OD densities. For example, Grid 48 in Figure 8(b) temporally lowers its price at 00:00 a.m., taking advantage of a higher OD density, and then gradually raises prices as the OD density diminishes to zero. In Figure 8(c), we observe the phenomenon of enticing ODs in Zone 2 with price surges in Zone 4. Compared to Episode B, the distance required for OD inducement is relatively shorter in Episode C, resulting in price surges at a lower level. These findings support the capability of PPOwM in solving spatio-temporal problems in the RL-SCBS system, by addressing various N/K ratios and spatially uneven OD engagement.

5.3. Experiment 3: sensitivity analysis

This experiment conducts a sensitivity analysis to examine the impact of the selected input parameters of the RL-SCBS system: the size of registered ODs, WTA variance, and the highest price, p^u. The pricing policy is trained for each different RL-SCBS setting and then tested on 30 test episodes used in Experiment 2. The baseline setting is as follows: registered OD size of 150, WTA variance of zero with a fixed mean of 0.5, and $p^u = 6$ with fixed $p^b = p^l = 2$. Each sensitivity analysis was performed by varying only one parameter while fixing the other two parameters on the baseline settings.

Figure 9 demonstrates the influence on performance metrics with box plot representations of 30 test episodes, each consisting of 100 iterations.

Figure 9(a) shows the impact of the size of the registered OD. As the size increased, the cost gradually decreased. This observation is because the reduction in penalties from increased task completion outweighs the

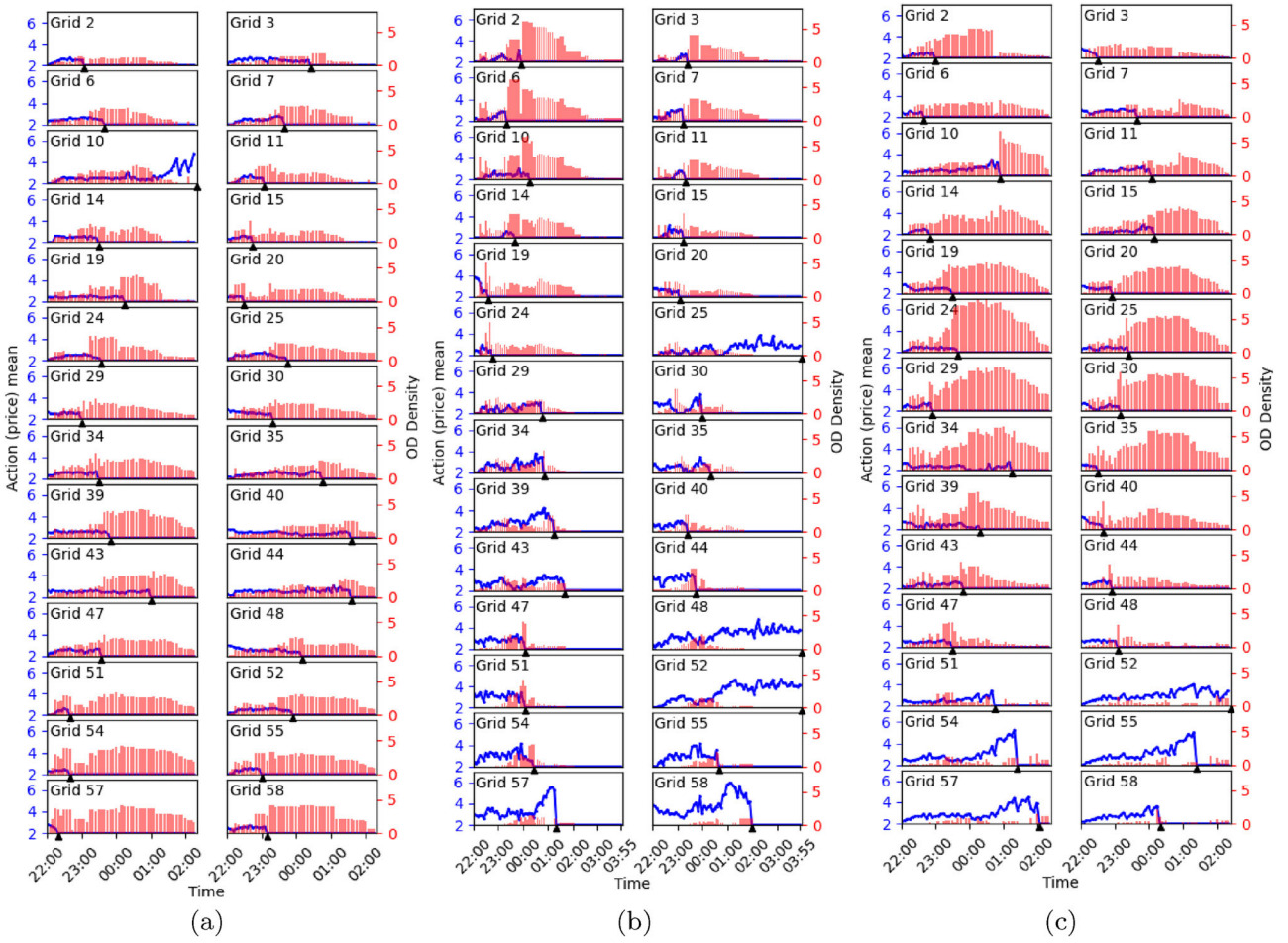


Figure 8. Temporal pricing actions and changes in spatial OD densities. (a) Episode A. (b) Episode B. (c) Episode C.

increase in OD wages. Comparing sizes 200 and 300, a similar task completion was observed, but OD wage was slightly reduced. This can be attributed to the presence of more ODs, allowing the agent to quickly find the maximum task completion rate and focus more on reducing minimal wages. The fact that a size of 300 provides a lower average attractiveness than a size of 200 further supports this inference.

Figure 9(b) shows the impact of the WTA variance. As the WTA variance increased, there was a slight increase in the overall cost, but no significant pattern in task completion or OD wages was observed. This suggests that the agent does not universally raise OD wages to match those with excessively high WTAs. Instead, it modulates prices to guarantee that only a subset of ODs accepts tasks, ensuring optimal task completion is feasibly attained. However, as WTA variance increased, the average WTA among those ODs who accepted tasks also rose, resulting in increased task attractiveness.

Figure 9(c) shows the impact of the price upper bound. As the p^u increased, the maximum distance capable of attracting ODs increased, leading to higher task completion rates. However, this also enlarged the agent's exploration space, which can result in a policy that assigned higher wages than the minimum necessary OD wage. When p^u was 4, due to the overly restrictive distance for attracting ODs, it could be observed that task completion rates were lowered, resulting in reduced costs.

Conversely, when p^u was set to 10, the task completion rate increased due to a higher inducement effect. However, this led to task attractiveness exceeding 1, resulting in a lower quality of solution due to the expanded solution space. Notably, the minimum cost was achieved with a moderate p^u of 6. Therefore, there is a trade-off in setting p^u between the inducement effect and the challenges of a larger action space.

5.4. Managerial insights

We recommend the following instructions for a practitioner in training and for applying the PPOwM algorithm based on the findings from three conducted experiments.

- The well-organised RL framework for the problem setting of RL-SCBS demonstrates performance on

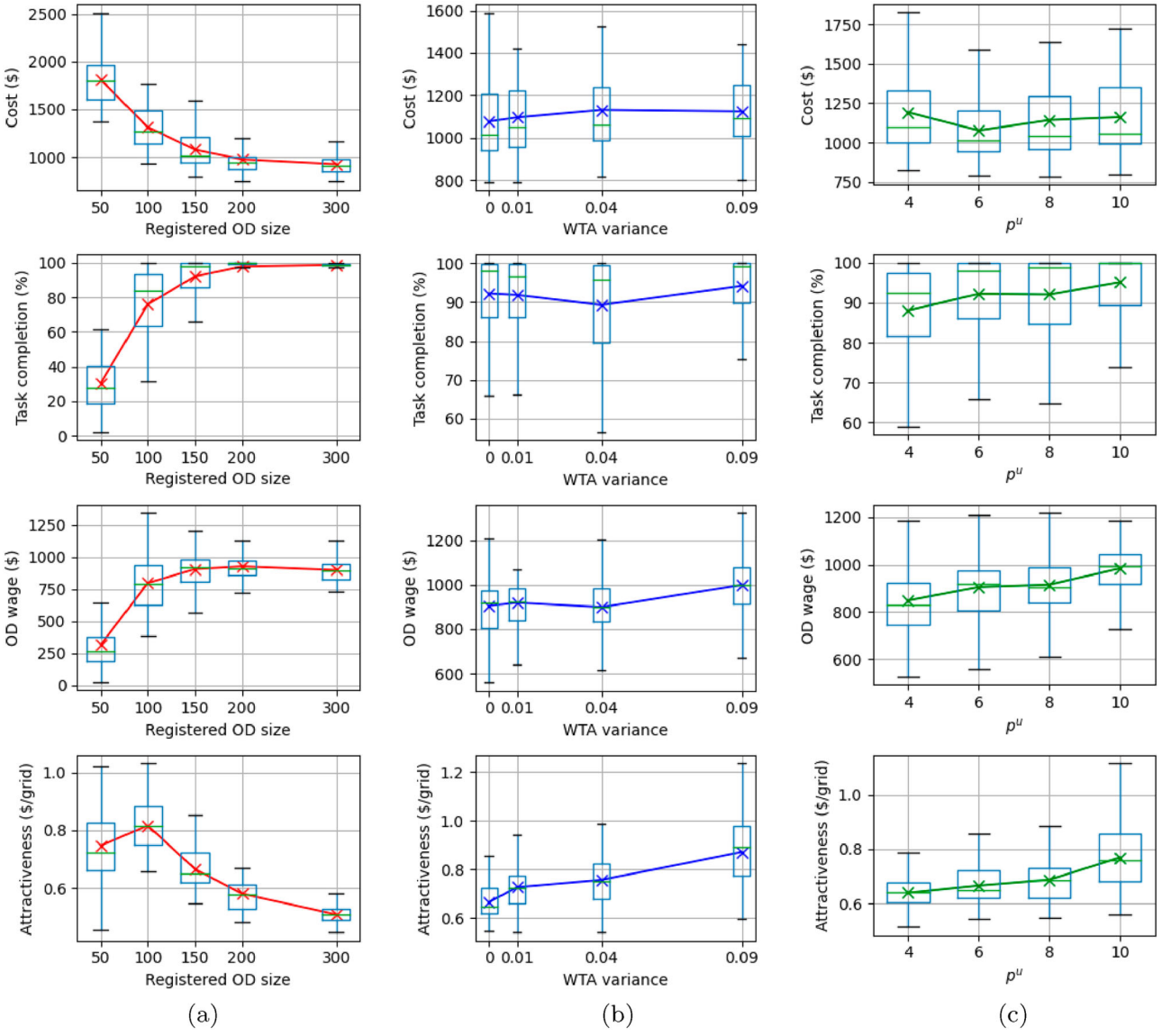


Figure 9. Performance metrics variations for 30 test episodes. (a) Registered OD size. (b) WTA variance. (c) p^u.

par with the optimisation result obtained by a solver. This is evidenced by the stable convergence of platform cost, task completion, and task attractiveness. During training, it is advisable to monitor all metrics to ensure that the learned policy approaches optimality. This may necessitate thorough hyperparameter tuning to identify the most effective settings where the metrics achieve their optimal values.

- Experiment 2 reveals that PPOwM can effectively attract OD movements even in cases when OD engagement location is uneven if they are within permittable range in the target city. In practice, OD distribution data may present characteristics such as dense engagement in specific areas and time intervals. This could accelerate the learning process with more effective pricing policies fit to the data.
- The sensitivity analysis highlights the importance of deciding the p^u value. We recommend collecting real data on the task price and distance at which an OD reserves a task. Once a sufficient amount of real data accumulates, this could be used to decide a moderate p^u that could attract the faraway ODs within a target area.

6. Conclusions

This study proposes a spatio-temporal pricing strategy that sets different compensation per region for ODs to swap shared electric micro-mobility (SEM) batteries. The goal is to resolve the spatial and temporal imbalance between ODs and tasks, thereby minimising the platform's operational cost, including OD compensation

and penalties for incomplete tasks. A proximal policy optimisation (PPO) algorithm is developed to generate continuous pricing actions, utilising a masking technique to enhance the learning process. This approach excludes information related to inactive grids from the loss function. The pricing strategy successfully provides real-time pricing solutions for a high-dimensional problem, with the complexity of dynamic OD participation and uncertainty in an OD's task acceptance condition.

The computation experiments demonstrate that the proposed algorithm minimises the daily cost near the optimal solution of a novel mixed integer programming model with perfect information of an OD's task acceptance condition. Simulation outcomes show that the RL agent assigns prices across diverse episodes with varying task and OD ratios, achieved by allocating higher prices in grids where OD density is low. Sensitivity analysis gives platform operators insights into controlling the parameters of the RL-SCBS system.

The proposed pricing strategy can be universally applied in spatial crowdsourcing with the following characteristics: (i) information on ODs' task acceptance conditions or preferences is concealed; (ii) ODs visit a specific location and dedicate a set time to complete a task; (iii) the task locations and prices are released for ODs to select autonomously in a first-come, first-served basis; (iv) uncompleted tasks at the end of the planning horizon are damaging to operational success and results in a per-unit penalty cost.

Future studies include expanding the decision-making to consider the relocation of SEMs, in which an OD's task becomes a pickup and delivery problem. Additionally, considering a backup workforce managed by the company to handle tasks left incomplete by ODs could broaden the application of the study.

Acknowledgments

The paper is an extended version of the work initially submitted to the 27th International Conference on Production Research (2023), further developed at the invitation for the special issue of ICPR 2023. The authors are grateful for the valuable comments from the editor-in-chief and anonymous reviewers.

Data availability statement

The data used in this study are available from the first author upon reasonable request.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT) [Grant Nos. RS-2023-00218913 and RS-2024-00337285].

Notes on contributors

Minjeong Kim is currently a Ph.D. student in the Department of Industrial Engineering at Seoul National University in Korea. She received the B.S. in Systems Management Engineering from Sungkyunkwan University, Korea, and M.S. in Department of Industrial Engineering from Seoul National University, Korea, in 2018 and 2020, respectively. Her research interests include SCM, platform operation, and reinforcement learning.

Ilkyeong Moon is a Professor of Industrial Engineering at Seoul National University in Korea. He received his B.S. and M.S. in Industrial Engineering from Seoul National University, and Ph.D. in Operations Research from Columbia University. His research interests include supply chain management, logistics, and inventory management. He published 180 papers in international journals. He was a former Editor-in-Chief of Journal of the Korean Institute of Industrial Engineers which is a flagship journal of Korean Institute of Industrial Engineers (KIIE). He was a president of KIIE in which he had served from 2019 to 2020. He currently serves as a co-editor-in-chief for European Journal of IE. He is a fellow of Asia Pacific Industrial Engineering and a board member of International Federation for Production Research.

ORCID

Ilkyeong Moon http://orcid.org/0000-0002-7072-1351

References

Alnaggar, Aliaa, Fatma Gzara, and James H. Bookbinder. 2021. "Crowdsourced Delivery: A Review of Platforms and Academic Literature." *Omega* 98:102139. https://doi.org/10.1016/j.omega.2019.102139.

Archetti, Claudia, Martin Savelsbergh, and M. Grazia Speranza. 2016. "The Vehicle Routing Problem with Occasional Drivers." *European Journal of Operational Research* 254 (2): 472–480. https://doi.org/10.1016/j.ejor.2016.03.049.

Arslan, Alp M., Niels Agatz, Leo Kroon, and Rob Zuidwijk. 2019. "Crowdsourced Delivery–a Dynamic Pickup and Delivery Problem with Ad Hoc Drivers." *Transportation Science* 53 (1): 222–235. https://doi.org/10.1287/trsc.2017.0803.

Bai, Sen, Shoufeng Tong, Xin Feng, Zhengang Jiang, Xin Bai, and Ranqi Xu. 2022. "Toward Dynamic Pricing for City-Wide Crowdsourced Instant Delivery Services." *IEEE Transactions on Mobile Computing* 23 (1): 909–924. https://doi.org/10.1109/TMC.2022.3228259.

Bimpikis, Kostas, Ozan Candogan, and Daniela Saban. 2019. "Spatial Pricing in Ride-Sharing Networks." *Operations*

Research 67 (3): 744–769. https://doi.org/10.1287/opre.2018.1800.

Boysen, Nils, Simon Emde, and Stefan Schwerdfeger. 2022. "Crowdshipping by Employees of Distribution Centers: Optimization Approaches for Matching Supply and Demand." *European Journal of Operational Research* 296 (2): 539–556. https://doi.org/10.1016/j.ejor.2021.04.002.

Cachon, Gerard P., Kaitlin M. Daniels, and Ruben Lobel. 2017. "The Role of Surge Pricing on a Service Platform with Self-Scheduling Capacity." *Manufacturing & Service Operations Management* 19 (3): 368–384. https://doi.org/10.1287/msom.2017.0618.

Chen, Chao, Shenle Pan, Zhu Wang, and Ray Y. Zhong. 2017. "Using Taxis to Collect Citywide E-commerce Reverse Flows: A Crowdsourcing Solution." *International Journal of Production Research*55 (7): 1833–1844. https://doi.org/10.1080/00207543.2016.1173258.

Chen, Chuqiao, Fugen Yao, Dong Mo, Jiangtao Zhu, and Xiqun Michael Chen. 2021. "Spatial-Temporal Pricing for Ride-Sourcing Platform with Reinforcement Learning." *Transportation Research Part C: Emerging Technologies* 130:103272. https://doi.org/10.1016/j.trc.2021.103272.

Chicago data portal. 2020. "E-Scooter Trips 2020." Accessed February 24, 2024. https://data.cityofchicago.org/Transportation/E-Scooter-Trips-2020/3rse-fbp6/data.

Fatehi, Soraya, and Michael R. Wagner. 2022. "Crowdsourcing Last-Mile Deliveries." *Manufacturing & Service Operations Management* 24 (2): 791–809. https://doi.org/10.1287/msom.2021.0973.

Gdowska, Katarzyna, Ana Viana, and João Pedro Pedroso. 2018. "Stochastic Last-Mile Delivery with Crowdshipping." *Transportation Research Procedia* 30:90–100. https://doi.org/10.1016/j.trpro.2018.09.011.

Google cloud. 2017. "Chicago Taxi Trips dataset." Accessed February 22, 2024. https://console.cloud.google.com/bigquery?project=alpine-sentry-271510&ws=!1m5!1m4!4m3!1sbigquery-public-data&2schicago_taxi_trips&3staxi_trips&pli=1.

Guo, Bin, Yan Liu, Leye Wang, Victor O. K. Li, Jacqueline C. K. Lam, and Zhiwen Yu. 2018. "Task Allocation in Spatial Crowdsourcing: Current State and Future Directions." *IEEE Internet of Things Journal* 5 (3): 1749–1764. https://doi.org/10.1109/JIOT.2018.2815982.

He, Fang, and Zuo-Jun Max Shen. 2015. "Modeling Taxi Services with Smartphone-Based E-hailing Applications." *Transportation Research Part C: Emerging Technologies* 58:93–106. https://doi.org/10.1016/j.trc.2015.06.023.

He, Suining, and Kang G. Shin. 2019. "Spatio-Temporal Adaptive Pricing for Balancing Mobility-On-demand Networks." *ACM Transactions on Intelligent Systems and Technology (TIST)* 10 (4): 1–28. https://doi.org/10.1145/3331450.

Hezewijk, Lotte, Nico Dellaert, Tom Van Woensel, and Noud Gademann. 2023. "Using the Proximal Policy Optimisation Algorithm for Solving the Stochastic Capacitated Lot Sizing Problem." *International Journal of Production Research* 61 (6): 1955–1978. https://doi.org/10.1080/00207543.2022.2056540.

Hu, Ming, and Yun Zhou. 2020. "Price, Wage, and Fixed Commission in On-Demand Matching." Available at SSRN 2949513.

Kadri, Ahmed A., Karim Labadi, and Imed Kacem. 2015. "An Integrated Petri Net and GA-based Approach for Performance Optimisation of Bicycle Sharing Systems." *European Journal of Industrial Engineering* 9 (5): 638–663. https://doi.org/10.1504/EJIE.2015.071777.

Kafle, Nabin, Bo Zou, and Jane Lin. 2017. "Design and Modeling of a Crowdsource-Enabled System for Urban Parcel Relay and Delivery." *Transportation Research Part B: Methodological* 99:62–82. https://doi.org/10.1016/j.trb.2016.12.022.

Liu, Jia-Xu, Yu-Dian Ji, Wei-Feng Lv, and Ke Xu. 2017. "Budget-Aware Dynamic Incentive Mechanism in Spatial Crowdsourcing." *Journal of Computer Science and Technology* 32 (5): 890–904. https://doi.org/10.1007/s11390-017-1771-6.

Ma, Hongyao, Fei Fang, and David C. Parkes. 2022. "Spatio-Temporal Pricing for Ridesharing Platforms." *Operations Research* 70 (2): 1025–1041. https://doi.org/10.1287/opre.2021.2178.

Macrina, Giusy, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. 2020. "Crowd-Shipping with Time Windows and Transshipment Nodes." *Computers & Operations Research*113:104806. https://doi.org/10.1016/j.cor.2019.104806.

Mancini, Simona, and Margaretha Gansterer. 2022. "Bundle Generation for Last-Mile Delivery with Occasional Drivers." *Omega* 108:102582. https://doi.org/10.1016/j.omega.2021.102582.

MirHassani, S. A., and F. Hooshmand. 2019. *Methods and Models in Mathematical Programming*. New York: Springer.

Nourinejad, Mehdi, and Mohsen Ramezani. 2020. "Ride-Sourcing Modeling and Pricing in Non-Equilibrium Two-Sided Markets." *Transportation Research Part B: Methodological* 132:340–357. https://doi.org/10.1016/j.trb.2019.05.019.

Osorio, Jesus, Chao Lei, and Yanfeng Ouyang. 2021. "Optimal Rebalancing and on-Board Charging of Shared Electric Scooters." *Transportation Research Part B: Methodological* 147:197–219. https://doi.org/10.1016/j.trb.2021.03.009.

Pan, Ling, Qingpeng Cai, Zhixuan Fang, Pingzhong Tang, and Longbo Huang. 2019. "A Deep Reinforcement Learning Framework for Rebalancing Dockless Bike Sharing Systems." In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 1393–1400. Honolulu, HI: AAAI Press.

Pender, Jamol, Shuang Tao, and Anders Wikum. 2020. "A Stochastic Model for Electric Scooter Systems." Preprint arXiv:2004.10727.

Qi, Wei, Lefei Li, Sheng Liu, and Zuo-Jun Max Shen. 2018. "Shared Mobility for Last-Mile Delivery: Design, Operational Prescriptions, and Environmental Impact." *Manufacturing & Service Operations Management* 20 (4): 737–751. https://doi.org/10.1287/msom.2017.0683.

Qin, Zhiwei, Xiaocheng Tang, Yan Jiao, Fan Zhang, Zhe Xu, Hongtu Zhu, and Jieping Ye. 2020. "Ride-Hailing Order Dispatching At Didi Via Reinforcement Learning." *INFORMS Journal on Applied Analytics* 50 (5): 272–286. https://doi.org/10.1287/inte.2020.1047.

Schulman, John, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. "High-Dimensional Continuous Control Using Generalized Advantage Estimation." Preprint arXiv:1506.02438.

Singgih, Ivan Kristianto, and Byung-In Kim. 2020. "Multi-Type Electric Vehicle Relocation Problem Considering Required

Battery-Charging Time." *European Journal of Industrial Engineering* 14 (3): 335–368. https://doi.org/10.1504/EJIE.2020.107697.

Tong, Yongxin, Libin Wang, Zimu Zhou, Lei Chen, Bowen Du, and Jieping Ye. 2018. "Dynamic Pricing In Spatial Crowdsourcing: A Matching-Based Approach." In *Proceedings of the 2018 International Conference on Management of Data*, 773–788. Houston, TX: Association for Computing Machinery.

Yang, Hai, Chaoyi Shao, Hai Wang, and Jieping Ye. 2020. "Integrated Reward Scheme and Surge Pricing in a Ridesourcing Market." *Transportation Research Part B: Methodological* 134:126–142. https://doi.org/10.1016/j.trb.2020.01.008.

Yun, Hyunsoo, Eui-Jin Kim, Seung Woo Ham, and Dong-Kyu Kim. 2022. "Price Incentive Strategy for the E-scooter Sharing Service Using Deep Reinforcement Learning." *Journal of Intelligent Transportation Systems* 28:409–423. https://doi.org/10.1080/15472450.2022.2135437.

Zha, Liteng, Yafeng Yin, and Zhengtian Xu. 2018. "Geometric Matching and Spatial Pricing in Ride-Sourcing Markets." *Transportation Research Part C: Emerging Technologies* 92:58–75. https://doi.org/10.1016/j.trc.2018.04.015.