Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

SPID: Solving the low-altitude takeoff and landing facility location problem via deep reinforcement learning

Xiaocheng LIU, Meilong LE, Yupu LIU‡, Minghua HU
College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106 , China,
E-mail: lxc2307084@nuaa.edu.cn; lemeilong@126.com; liuyupu@nuaa.edu.cn; minghuahu@nuaa.edu.cn
Received July 28, 2025; Revision accepted Nov. 26, 2025; Crosschecked

Abstract: The siting of low-altitude takeoff and landing platforms is a fundamental challenge for the future development of urban air mobility (UAM). We formulate the problem as a variant of capacitated facility location problem (CFLP) with flight-range and service-capacity constraints, and propose SPID, a Deep Reinforcement Learning (DRL)–based solution framework. The problem is modeled as a Markov Decision Process (MDP). To address dynamic coverage, we design a DRL framework with a multi-head attention mechanism to capture spatio-temporal patterns and integrate dynamic and static information into a unified input state vector. A Gated Recurrent Unit (GRU) is used to generate the query vector, enhancing sequential decision-making. The action network is guided by a loss function combining unmet demand penalties with service- distance costs, enabling end-to-end optimization. Experimental results demonstrate that SPID significantly improves solution efficiency and robustness over traditional methods under flight and capacity constraints. It also outperforms heuristic algorithms in accuracy, with a performance gap of around 6% compared to exact solutions. This work provides an efficient, scalable approach to low-altitude site selection, supporting rapid decision-making in large-scale UAM scenarios.

Key words: Low-altitude planning; Facility site selection; Deep reinforcement learning; Algorithm exploration
https://doi.org/10.1631/FITEE.2500534 **CLC number:**

unedited

1 Introduction

urban air mobility (UAM) refers to the low-altitude transportation of passengers and cargo within urban areas, typically involving electric vertical takeoff and landing aircraft, drones, and other low-altitude flying vehicles. It is considered a key direction for the future development of urban transportation. NASA predicts that by 2030, the number of drone flights in urban airspace will reach 500 million. To support this development, governments and industry stakeholders are actively promoting UAM, with low-altitude logistics identified as one of its most promising application areas. For example, Amazon launched drone delivery pilot projects in California and Texas at the end of 2022. Vertiports, which support various UAM operational scenarios, are critical infrastructure for UAM. Therefore, the site selection of these vertiports plays a crucial role in ensuring the efficient development of UAM, making the scientific and rational planning of their layout a core issue.

Selecting sites for UAM vertiports is not a one-time static planning task but rather a multi-stage decision-making process. Initially, it is necessary to evaluate the site selection costs and service levels under different project configurations, such as the number of facilities, capacity, and budget, to determine the optimal project configuration. Additionally, since UAM vertiports serve as essential social infrastructure, the site selection process should consider a broader management perspective, taking into account multiple social indicators. Moreover, given that the site selection of UAM vertiports is a strategic planning issue, the results must be forward-looking, meaning that current planning decisions must assess future service levels. Thus, the UAM vertiport site selection problem must integrate both operational

‡ Corresponding author
ORCID: Xiaocheng LIU, https://orcid.org/0000-0002-1748-0819; Meilong LE, https://orcid.org/0000-0002-1748-0819; Yupu LIU, https://orcid.org/0009-0003-1355-6104
© Zhejiang University Press 2025

constraints of UAM and site planning requirements, balancing construction costs and service levels with which requires multiple rounds of scheme comparison, the chosen method must be both efficient and capable of generalizing well to different situations.

The site selection problem for UAM vertiports falls within the category of urban facility location problems. These classical combinatorial optimization problems are typically solved using mathematical programming solvers or heuristic algorithms. While mathematical solvers can provide exact solutions, they require significant time and memory as the problem size increases, which can make computation impractical. Heuristic algorithms, on the other hand, improve computational efficiency but often sacrifice solution quality and generalization ability. Moreover, accelerating the solving process with heuristic algorithms requires developing targeted rules based on the current data and problem characteristics, which can be time-consuming and difficult. As a result, existing solution methods face a dilemma in practical engineering applications, where it is challenging to balance computational efficiency with solution accuracy.

Deep Reinforcement Learning (DRL) methods, which automatically learn solving strategies through data-driven approaches, can quickly generate feasible solutions for new problem instances. DRL's solving strategy captures the complex spatial relationships between demand and capacity, operates without relying on manually designed rules, and naturally possesses strong generalization capabilities. By using DRL, the site selection decision-making process can be significantly accelerated. During the early stages of decision-making, DRL can reduce the time required to compare schemes under various parameter settings. In the final decision stage, DRL can provide high-quality initial feasible solutions for the Mixed-Integer Linear Programming (MILP) model, thereby reducing computation time for exact solutions. The combination of DRL and MILP can achieve an optimal balance between efficiency and accuracy, better meeting the practical engineering needs.

In summary, this paper explores the use of DRL methods to solve the UAM vertiport site selection problem. The main contributions include:

(1) Formulating the UAM site selection problem as a variant of the facility location problem with capacity constraints and allowed unmet demand, reflecting the supplementary role of UAM within the existing transportation system. This model is more practical and has significant engineering relevance.
(2) Developing a DRL-based solution framework and exploring its use as an initial feasible solution for MILP exact solving. This approach strikes a balance between solving efficiency, robustness, and scalability.
(3) Introducing three key indicators—geographic coverage, effective coverage, and facility utilization—to quantify the social benefits and foresight of site selection plans.

The structure of the paper is as follows: Section 2 reviews related research; Section 3 presents the model construction; Section 4 details the algorithm design; Section 5 discusses the experimental study; Section 6 concludes the research and outlines future directions.

2 Literature review

This paper addresses the problem of optimizing the location of urban low-altitude vehicle take-off and landing sites, considering multiple candidate bases and the demand distribution of customer nodes. As low-altitude logistics represents one of the most rapidly advancing application scenarios in the UAM industry, most existing literature focuses on logistics and distribution-oriented site selection problem. The landing bases studied in these works are commonly referred to as UAV centers, distribution hubs, hospitals, UAV stations, warehouses, drone hives, or logistics platforms. In addition, UAV deployment for communication, particularly as temporary signal-enhancing base stations, has become a research hotspot. Despite different application contexts, these various platforms share the common function of service provision, conceptually aligning with the take-off and landing sites studied in this paper. Therefore, the following literature review is organized around three key UAM application scenarios: logistics and distribution, UAV-based communication, and urban air mobility infrastructure planning.

In the field of low-altitude logistics and distribution, many studies address landing point location. For example, Shavarani proposed a hierarchical model to minimize total system costs by optimizing

unedited

UAV launch and charging station locations (Shavarani et al. 2018). Shavarani extended this with a fuzzy multilevel model incorporating uncertain demand and costs (Shavarani et al. 2019). Kim explored stochastic siting under uncertain flight distances (Kim et al. 2019), while Meskar investigated battery consumption rates driven by payload effects (Meskar et al. 2024). Ibrahim introduced a multi-objective trajectory optimization algorithm based on k-means clustering and greedy sequencing (Ibrahim et al. 2024). Meng combined clustering, center-of-mass location, and genetic algorithms to coordinate UAVs and ground vehicles in blood delivery tasks (Meng 2024). Mahmoodi proposed a multi-objective site-path optimization model integrating time windows, dual service demands, and recharge constraints, solved via NSGA-II with Bayesian risk modeling (Mahmoodi et al. 2025). Zhang et al. addressed the integration of safety considerations with economic objectives in urban air logistics (UAL) hub location, proposing a multi-objective mixed-integer programming model that simultaneously minimizes costs and third-party safety risks, solved via an improved non-dominated sorting genetic algorithm with adaptive operator selection (Zhang et al. 2025).

In the communication domain, UAVs are studied as mobile base stations. Gopi applied reinforcement learning to optimize UAV base station locations, using data rate sum as the reward function (Gopi et al. 2021). Ryu modeled a partially observable multi-UAV deployment for supporting mobile terrestrial users, solved via deep reinforcement learning (Ryu et al. 2024). To address scalability. Premkumar et al. cluster customers first and then site drones, striking a balance between optimality and runtime (Premkumar et al. 2025).

With the emergence of the UAM concept, the siting and capacity planning of eVTOL infrastructure has become a central issue. Lynskey introduced a capacitated facility location model for "drone ports," integrating Traveling Salesman Problem (TSP) routing and k-means clustering (Lynskey et al. 2019). Khalife framed the siting problem as a p-Median variant to maximize travel time savings over ground transportation (Khalife et al. 2021). KrishnaKumar applied graph reinforcement learning for eVTOL decision support, using dual graph representations passed through Graph Convolutional Networks (GCNs) (KrishnaKumar et al. 2022). Zhu and Tengkuo et al. addressed robust siting under post-disaster demand uncertainty using column generation and Benders decomposition (Zhu et al. 2022; Tengkuo et al 2022). Other notable contributions include hybrid clustering and p-Median optimization (Rahman et al., 2023), genetic algorithms for infrastructure-accessible eVTOL locations (Volakaki et al., 2024), and data-driven Markov decision process formulations for dynamic drone delivery planning (Paul et al., 2024).

Further studies explored travel time valuation for vertiport site selection using iterative weighted k-means (Zhao and Feng, 2024), and multi-factor logistics hub siting using enhanced Particle Swarm Optimization (PSO) algorithms (Zheng et al., 2024). Song evaluated the influence of vertiport placement on transportation network topology (Song et al. 2025), while Lee proposed a three-stage geospatial analysis for site selection, including suitability, regulation, and location allocation (Lee et al. 2025). Sun developed a framework for siting coastal UAV monitoring bases, balancing spatial coverage and cost (Sun et al. 2025).

Existing studies on UAV landing site optimization commonly adopt three types of objective functions: (1) Cost-based objectives, aiming to minimize fixed facility costs, UAV acquisition and operating expenses, and transportation costs. Some adopt a two-stage approach: first optimizing facility costs, then operational costs; (2) Time-based objectives, focusing on minimizing total delivery time or completion time. Some models incorporate customer utility functions to reflect delay impacts; (3) Coverage-based objectives, often used in communication and medical scenarios, which seek to maximize demand coverage or service quality. In terms of constraints, most models consider UAV endurance (by distance, time, or energy), payload limits, facility counts, and budget constraints. Some also include practical factors such as time windows, demand uncertainty, network accessibility, charging station queuing, and obstacle-aware routing.

Many studies are based on classical facility location models, such as the p-Median, p-Center (Hakimi, 1965), and Maximal Covering Location Problem MCLP (Church, 1974), providing a solid theoretical foundation. For solution methods, exact algo-

unedited

rithms—mostly via solvers like CPLEX and Gurobi—are most common, while a few propose customized approaches. Heuristic methods (e.g., genetic algorithms, simulated annealing, large neighborhood search) are often used to handle large-scale or complex problems efficiently.

Beyond traditional operations research methods, recent advances in deep learning and reinforcement learning have opened new avenues for solving complex optimization problems. In particular, DRL has demonstrated remarkable capability in learning decision-making policies for combinatorial optimization tasks without requiring extensive domain-specific heuristics. The majority of existing DRL research in logistics optimization has focused primarily on routing problems, where the objective is to determine optimal visit sequences and travel paths given fixed facility locations. Kool et al. (2019) developed an attention-based model for the TSP and VRP , learning construction heuristics through encoder-decoder architectures. Nazari et al. (2018) applied reinforcement learning to vehicle routing with stochastic demands, demonstrating policy generalization across problem instances. Hottung and Tierney (2020) proposed neural large neighborhood search for Capacitated Vehicle Routing Problem CVRP , achieving competitive performance with traditional methods while offering computational efficiency on large-scale instances. These studies focusing on route planning have established a methodological foundation for applying machine learning to logistics optimization. However, this line of research addresses a problem that is fundamentally different from facility location decision-making. Route planning operates within an existing topological network, whereas facility location involves determining optimal sites within a dispersed demand distribution.

The integration of Graph Neural Networks (GNNs) with reinforcement learning has further emerged as a powerful paradigm for handling graph-structured optimization problems. Han et al. (2024) proposed a novel framework combining GNNs and DRL for resource allocation in computing power networks, showing that graph-based representations can effectively capture the structural dependencies inherent in network optimization problems. Their approach leverages GNNs' ability to aggregate neighborhood information, enabling the model to learn generalizable policies across different network topologies and scales. More directly relevant to spatial optimization, Liang et al. (2024) developed SPONet, a DRL framework specifically designed for urban spatial decision problems including billboard location optimization. Their work demonstrated that DRL-based approaches can achieve over 20× speedup compared to traditional solvers while maintaining solution quality, establishing the feasibility of learning-based methods for spatial facility location problems. Li et al. developed a reinforcement learning-based iterative algorithm for capacitated hub location problems in UAV delivery networks, employing specialized policy networks based on Graph Neural Networks and Gated Recurrent Units to embed demand and spatial patterns, demonstrating superior performance and robust generalization across networks of various scales (Li et al. 2025).

Within the UAM domain specifically, several recent studies have begun exploring DRL applications, though primarily focused on operational and tactical decision-making rather than strategic infrastructure planning. Park et al. (2023) applied multi-agent reinforcement learning to coordinate cooperative air transportation services, addressing real-time routing and scheduling decisions for active UAM operations. Garcia et al. (2023) developed a DRL framework for intelligent safety management in UAM, focusing on collision avoidance and secure flight operations. In the domain of UAM communication infrastructure, Han et al. (2022) proposed DRL-assisted spectrum management for cellular-based UAM networks, optimizing wireless resource allocation to support UAM operations. This line of research was extended by Apaza et al. (2023), who combined spectrum management with air traffic management using multi-agent DRL under resource constraints. While these studies demonstrate the viability of DRL in UAM contexts, they address operational-level problems (service coordination, safety control, communication resource allocation) rather than strategic facility location decisions.

In summary, existing DRL-based studies in logistics and transportation primarily focus on solving route planning problems, such as the TSP, VRP, and CVRP, under the assumption of fixed depot locations and predetermined visit sequences. In contrast, this study addresses the strategic facility location deci-

unedited

sion, which determines where to establish vertiports and how to allocate their capacities before operations begin. This represents a shift in research focus from tactical-level route optimization to strategic-level infrastructure planning, requiring distinct state representations, action spaces, and reward structures within the DRL framework. Based on Liang et al.'s approach, this paper explores the UAV facility location problem from a macro-management perspective (Liang,2024). It formulates the problem as a CFLP under aerial operational constraints and solves it using a DRL-based framework. This enables an end-to-end solution for the facility location problem under partial demand satisfaction, providing a scalable alternative to traditional exact or heuristic methods.

3 Model

3.1 CFLP Model

In classical facility location models, the p-median model aims to minimize total travel time and is therefore suited to time-sensitive services such as UAM passenger transport. By contrast, the maximal covering location model seeks to satisfy as much demand as possible using a limited number of facilities, and is commonly applied in emergency logistics. However, vertiports in a UAM network function as critical capacity nodes, so their placement cannot be determined by service distance alone. Site selection must also consider strict capacity constraints and the fact that part of the demand can be deferred. This is particularly relevant because UAM is still in an early stage of deployment, and many of its potential trips currently have mature alternative modes. Low-altitude takeoff and landing sites can be viewed as low-altitude analogues of civil aviation terminal areas, characterized by strict capacity limits on the number of daily operations.

The classic Capacitated Facility Location Problem (CFLP) requires all demand to be fully satisfied. However, the problem addressed in this paper allows for partially unmet demand, making it a variant of the traditional CFLP.

In the standard CFLP, the objective is to select p facility locations within a given network or space to minimize the total service cost (typically represented by the sum of distances between facilities and demand points). In contrast, our model relaxes the full demand satisfaction constraint and incorporates an unmet demand penalty directly into the objective function. This variant model better captures the phenomenon in UAM operations , where due to limited facility capacity, part of the demand cannot be immediately served. It also reflects early-stage urban air mobility systems, which are characterized by a small number of facilities and unevenly distributed demand. In this way, the model enforces capacity constraints while also preserving operational flexibility and cost efficiency.

To formalize the problem, we define the set of urban air mobility demand points as , $N = \{1,2,3,\dots,n\}$ and the set of candidate takeoff and landing sites for urban aerial vehicles as $M = \{1,\ 2,\ \dots,\ m\}$. The decision variables used in the proposed model are defined as follows:

$$x_{ij} = \begin{cases} 1, Demand\ point\ i\ is\ served\ by\ facility\ j \\ 0, \quad else \end{cases}$$

$$y_j = \begin{cases} 1, facility\ is\ selected \\ 0, \quad else \end{cases}$$

Secondly, let d_{ij} denote the distance between demand point i and takeoff and landing site j, n_i the total demand at point iii, and C_j the capacity limit of site j. Let N_i represent the portion of demand at point i that can be served by site j, and define the neighborhood set for each demand point as $N_j = \{j \in M | d_{ij} \leq D_j\}$, where D_j is the service radius of site j. Finally, let p denote the maximum number of takeoff and landing sites that can be constructed.

Based on these definitions, the variant of the Capacitated Facility Location Problem (CFLP) proposed in this paper allows for partial unmet demand, which can be served in a delayed or deferred manner. The formulation of this model is presented as follows.

$$Minimize\ z = \sum_{i=1}^{n}\sum_{j=1}^{m}[d_{ij}x_{ij} + M(\sum_{j=1}^{m} C_j - \sum_{i}^{n} n_i(1 - x_{ij})] \quad (1)$$

$$Subject\ to\ \sum_{j=1}^{m} x_{ij} \leq 1\ , i = 1,\dots,n \quad (2)$$

$$\sum_{j=1}^{m} y_j = p\ , i = 1,\dots,n \quad (3)$$

$$x_{ij} \leq y_j, i = 1,\dots,n, j = 1,\dots,m \quad (4)$$

unedited

$$\sum_{i=1}^{n} n_i x_{ij} \le C_j y_j, i = 1, \dots, n, j = 1, \dots, m \quad (5)$$

$$x_{ij}, y_j \in \{0,1\} \ i = 1, \dots, n, j = 1, \dots, m \quad (6)$$

Constraint (2) ensures that each demand point is assigned to at most one facility. Constraint (3) limits the total number of facilities selected for construction to the predefined number p. Constraint (4) ensures that demand can only be satisfied by the facilities that have been established. Constraint (5) guarantees that the total demand allocated to each facility does not exceed its maximum service capacity.

3.2 Reformulate as MDP Model

To solve the problem using a deep reinforcement learning (DRL) approach, the CFLP must first be formulated as a Markov Decision Process (MDP), enabling the use of DRL to iteratively update the policy or value function based on temporal dynamics. In this study, the MDP formulation is addressed using the REINFORCE algorithm (Williams, 1992). Fig. 1 illustrates the schematic of the reinforcement learning process, where an intelligent agent (SPID) interacts with the environment to determine optimal siting decisions for takeoff and landing facilities, guided by the MDP-based model. The components of the MDP framework—defined as a quintuple (S, A, P, R, γ)—are specified for the CFLP as follows, Table 1 is a comprehensive summary of all notations and variables used in the MDP formulation.

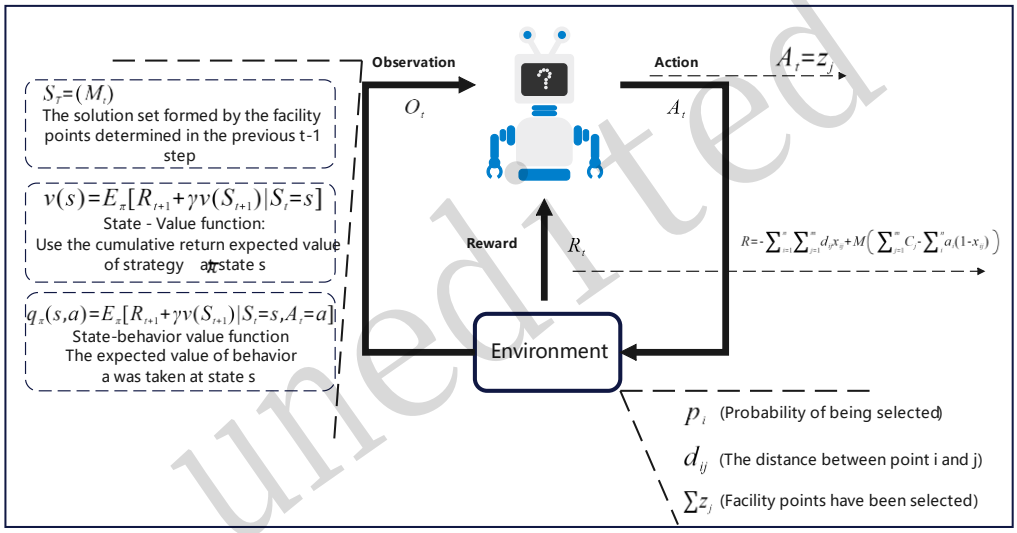


Fig. 1 SPID "RL expression"

For demand allocation, each demand point is assigned to a feasible facility (i.e., one that is not overloaded), based on a proximity-based allocation strategy denoted as $\pi(j)_t$. Here, $C_j\pi(j)_t$ represents the remaining capacity of facility j at the time of decision step t. A large penalty coefficient M is introduced, set to 10^6 in this study, to penalize infeasible assignments. The assignment of demand points to facilities is governed by the following rules:

$$\mathrm{j} \in \pi\ , \quad \text{if } C_j\pi(j)_t > n_i \quad (11)$$

$$C_i\pi_t = \min\{d_{ij} \times n_i\}\ , \forall \mathrm{j} \in \pi, \mathrm{d}_{ij} \le \mathrm{r} \quad (12)$$

$$C_i\pi_t = M \times n_i, if\ \pi(j)_t = \emptyset \quad (13)$$

The objective of the model is to minimize the total cost incurred in serving all demand points. With the discount factor set to $\gamma = 1$, the action-value function $q_\pi(s, a)$ is defined as the expected sum of rewards:

$$q_\pi(s_k, a) = p_\theta(\pi|s) * R_{S_1} + \cdots + p_\theta(\pi|s) * R_{S_k} \quad (14)$$

To solve the model, this paper employs a policy gradient method based on the Monte Carlo approach. The algorithm updates the parameters θ using stochastic gradient ascent, with the update rule given by:

$$\theta_{t+1} = \theta_t + \alpha \nabla L(\pi) \quad (15)$$

unedited

An increase in $L(\pi)$ indicates that the current policy yields higher rewards, and parameter updates are made in the direction that maximizes the expected reward. The loss function for training is defined as $L(\theta|s) = E_{p_\theta(\pi|s)}[L(\pi)]$, where:

Table 1. Notation and Variable Definitions

Notation	Description
State and Action Variables	
s_t	State at time step t
M_t	Solution set of selected facilities up to step t
a_t	Action taken at time step t
z_j	Node selected as facility at decision step
Decision Variables	
x_{ij}	Binary variable: 1 if demand i is assigned to facility j, 0 otherwise
π	Complete solution sequence $\{\pi_1, ..., \pi_p\}$
π_t	Facility selected at time step t
$\pi(j)_t$	Set of demand points assigned to facility j at step t
Policy and Learning	
θ	Policy parameters
$p_\theta(\pi\|s)$	Probability of selecting solution π given state s under policy θ
$p_\theta(\pi_t\|s, \pi_{1\sim t-1})$	Probability of selecting facility π_t given current state and previous selections
$q_\pi(s, a)$	Action-value function
α	Learning rate (step size) for gradient ascent
Capacity and Cost	
$C_j\pi(j)_t$	Remaining capacity of facility j at time step t
$C_i\pi_t$	Cost of assigning demand point i under policy π at step t
$L(\pi)$	Loss function (total cost)
$L(\theta\|s)$	Expected loss under policy parameters θ
$\nabla L(\pi)$	Gradient of loss function

State: The state at time step t, denoted as $s_t = (M_t)$, represents the solution set consisting of facility locations selected in the previous t−1 steps. The initial state is M_0, which corresponds to an empty solution set.

Action: At each decision step, an action involves selecting a node to serve as a facility. The action a_t corresponds to choosing node z_j, such that $a_t = z_j$ if node z_j is selected as a facility at time step t.

Transition: Given the current state s_t, the agent selects an action a_t that chooses node z_j as a new facility. Once z_j is added to the current solution setM_t, the environment transitions to the next state s_{t+1}, defined as $s_{t+1} = M_{t+1} = (M_t; z_j)$.

Discount Factor: The discount factor is set as $\gamma = 1$, indicating that future rewards are not discounted. Since the final objective value is treated as the only reward, cumulative reward values $v(s)$ are not considered.

Reward: The optimization direction of the model is driven by the reward function. In this paper, the reward is defined as the negative of the objective function:

$$R = -(\sum_{i=1}^{n}\sum_{j=1}^{m}[\, d_{ij}x_{ij} + M\left(\sum_{j=1}^{m} C_j - \sum_{i}^{n} n_i(1 - x_{ij})\right)] \tag{7}$$

This formulation encourages the agent to minimize the total service cost and penalize capacity violations.

In the MDP framework, p_θ denotes the policy that governs the probability of the agent selecting an action a_t given state s_t. At each time step, the agent selects one node as a facility location, and this process continues until a total of p facilities have been selected, forming the final solution $s_T = (M_t)$. This selection process can be represented as follows:

$$\pi = \{\pi_1, \dots, \pi_p, p \le n\} \tag{8}$$

unedited

$$p_\theta(\pi|s) = \prod_{t=1}^{p} p_\theta(\pi_t|s, \pi_{1\sim t-1}), p \le n \tag{9}$$

Each facility j has a fixed capacity C_j, and its remaining capacity is dynamically updated at each step based on the demand it serves. The update rule is defined as:

$$C_j\pi(j)_t = C_j - \sum_{i\in j} n_i \tag{10}$$

4 Methodology

The UAM takeoff and landing site location problem shares common characteristics with traditional ground facility location problems, as both can be considered facility location problems with capacity constraints. However, as UAM takeoff and landing sites are integrated facilities involving both ground and air operations, the location selection process must account for various constraints closely related to low-altitude operations.

To systematically explain the characteristics of this problem and the solution approach, the chapter is organized as follows: Section 4.1 analyzes the unique attributes of the UAM takeoff and landing site location problem. Sections 4.2 and 4.3 will explain how the proposed SPID method incorporates the key constraints that must be considered during the UAM location selection process.

4.1 Analysis of UAM Takeoff and Landing Site Location Problem Characteristics

The analysis of the UAM takeoff and landing site location problem can be approached from multiple perspectives, including those of air traffic control regulators, operators, and demand stakeholders. It can also be based on the full operational process of UAM. To make the research more general, this paper will construct the problem from the perspective of the entire UAM operational process, focusing on the strategic location selection phase for takeoff and landing sites.

UAM operations can be systematically divided into three sequential decision-making stages. The first stage involves site feasibility and candidate set generation, which is part of the upstream preparatory analysis. In this stage, spatial analysis is used to identify potential takeoff and landing site locations. Key factors considered include airspace constraints, structural feasibility, regulatory compliance, and land ownership and acquisition costs. The output of this stage is a discrete set of candidate locations, each associated with its development cost and operational constraints, providing the input for subsequent models.

The second stage is strategic location selection and capacity planning, which is the core focus of this study. Based on the candidate sites generated in the first stage, decisions are made regarding the specific location selection, demand distribution from the origin node to the selected sites, and evaluation of system-level performance metrics such as coverage, accessibility, and service level. The SPID model proposed in this paper addresses the problem at this stage, where the input parameters (candidate locations, demand nodes, coverage radius, and capacity) implicitly include upstream constraints, while the model output (selected facility locations) provides key input for downstream operational planning.

The third stage is operational planning and four-dimensional trajectory design, which is the downstream extension of the location decision. Once the facility locations are determined, this stage focuses on fine-tuned operational issues, including four-dimensional trajectory planning, flight scheduling, energy management, and detailed demand-capacity matching. While these operational decisions are crucial for system implementation, they are essentially downstream issues derived from the location selection results. The location model does not explicitly optimize trajectory details, but rather ensures the feasibility and efficiency of the selected locations in operation by maintaining proper facility spacing and adhering to capacity limits derived from operational analysis.

In summary, the strategic location selection of UAM takeoff and landing sites serves as a crucial link between upstream feasibility analysis and downstream operational planning. To explain the core constraints at each stage and their representation in this study's model, Table 2 summarizes the stage divisions, key constraints, and corresponding input

unedited

parameters throughout the UAM operational process. It also provides a systematic explanation of how physical, airspace, and operational constraints at each stage are connected to the strategic location selection phase. Specifically, the physical and regulatory constraints from the first stage are internalized as input parameters for the second stage, laying the foundation for the feasibility of the location selection solutions. Meanwhile, operational efficiency constraints from the third stage are proactively considered during the location selection phase through pre-set candidate facility locations and capacity limitations, thus ensuring the feasibility and effectiveness of subsequent operations at the strategic level.

Table 2 Correspondence between SPID Input Parameters and Key Constraints at Each UAM Operational Stage

Stage	Key Factor	SPID Model Input Parameters
Stage 1	Physical and Regulatory Constraints (e.g., Airspace, Structure, Regulations)	Pre-screened set of candidate sites Inherent capacity parameters of the sites
Stage 2	Strategic Objectives (e.g., Coverage, Capacity, Demand, Budget)	/
Stage 3	Operational Efficiency Constraints (e.g., Airspace Management, Scheduling, Energy Management)	Candidate site locations: Implicit airspace management feasibility Service radius and facility capacity limitations: Support operational scheduling and energy management

To further evaluate the comprehensive performance of the location selection solution from a system perspective, and in response to the decision-making needs of strategic public facility planning, this paper introduces the following set of performance metrics. These metrics are designed to quantify and reflect multiple objectives, such as facility coverage, resource utilization, and operational efficiency. Their definitions are as follows:

Geographic Coverage Rate ($\boldsymbol{S_g}$):

$$S_g = 1 - \left(\frac{Number\ of\ geographically\ uncovered\ users}{Total\ number\ of\ users} \right) \quad (17)$$

This indicator reflects the proportion of users that are geographically covered by at least one facility.

Effective Coverage Rate ($\boldsymbol{S_e}$):

$$S_e = \frac{Number\ of\ users\ assigned\ to\ a\ facility\ without\ exceeding\ its\ capacity}{Total\ number\ of\ users} \quad (18)$$

This indicator measures the proportion of users who are not only covered but also successfully assigned to a facility without violating capacity constraints.

Facility Utilization Rate ($\boldsymbol{S_f}$):

$$S_f = \frac{Total\ allocated\ demand}{Total\ facility\ capacity} \quad (19)$$

This indicator represents the overall utilization level of facility resources relative to their maximum service capacity.

Weighted Distance Cost $\boldsymbol{C_w}$:

$$C_w = d_{ij} x_{ij}, i \in j \quad (20)$$

The total demand-weighted Euclidean distance between each demand point and its assigned facility. Lower values indicate shorter average travel distance for users, which reflects higher operational efficiency.

Unweighted Distance Cost $\boldsymbol{C_u}$:

$$C_u = x_{ij}, i \in j \quad (21)$$

The average Euclidean distance between demand points and their assigned facilities, without demand weighting. This metric provides a distance-based evaluation that is independent of demand magnitude.

4.2 Problem restoration

The CFLP is inherently a combinatorial optimization problem, where the selection of takeoff and landing sites does not follow a specific sequence. However, when CFLP is formulated as a MDP for solution via deep reinforcement learning, the selection of facility sites becomes sequential in nature.

This artificially imposed decision order breaks the inherent symmetry of the original combinatorial problem by mapping essentially equivalent solutions

unedited

onto different decision trajectories. As a result, it can lead to a significant expansion of the state space and introduce "order bias"—where the model becomes more likely to find high-quality solutions under certain selection orders than others.

To address these issues, Liang et al. (2024), in their study on billboard placement optimization using DRL, proposed a dynamic coverage vector method, which effectively mitigates the adverse effects introduced by decision sequencing in combinatorial optimization problems.

A dynamic coverage vector D_i is introduced for each demand point iii to represent its coverage state. This vector is dynamically updated after each siting action a_t. Each D_i is a column vector with elements in the range [0,1]. The closer an element is to 0, the less likely the corresponding demand point is to be selected in subsequent steps—indicating that it has already been sufficiently covered or is very close to a selected facility. Initially, all coverage vectors are set to 1:

$$D_i = 1, i = 1, \dots, n \tag{22}$$

Let π_t denote the node selected at time step t, and $C_{r(\pi_t)}$ represent the set of demand points within a coverage radius r of node π_t. The points in $C_{r(\pi_t)}$ are sorted in ascending order of distance from π_t. Define $K_{r(\pi_t)}$ as the subset of $C_{r(\pi_t)}$ consisting of demand points that are actually served by facility π_t.. Let $N_{C_{r(\pi_t)}}$ be the cardinality of $K_{r(\pi_t)}$, and c_i the number of times point iii is covered. The iterative update of D_i during decoding is governed by the following equations:

$$c_i = \max\{i \mid \sum_{w \in K_j} n_i \le C_j, i \ge w\} \tag{23}$$

$$D_i = D_i \times \frac{c_i}{N_{C_{r(\pi_t)}}}, i = 1, \dots, n \tag{24}$$

$$D_i = \begin{cases} 0, if\ i \epsilon K_{r(\pi_t)} \\ 1, otherwise \end{cases} \tag{25}$$

According to the above update rules, the more frequently a node i is covered, the smaller its D_i becomes, thus reducing its likelihood of being selected in future steps. Both the degree of coverage and proximity to the selected node are incorporated into the decision-making process.

To more accurately calculate the probability of selecting a node as a facility at each time step, the model learns a refined representation that integrates both D_i and a learnable weight matrix W^I. The resulting dynamic embedding vector F_i is defined as:

$$F_i = D_i W^I \tag{26}$$

4.3 SPID Method implementation

Based on the introduction of the dynamic coverage vector D_i, this paper categorizes the information required for model inference into two types: static information and dynamic information. Static information includes the two-dimensional coordinates and demand values of each demand point, representing fixed attributes that remain constant throughout model training. In contrast, dynamic information reflects evolving system states, such as the coverage status of each demand point and the capacity utilization of each facility.

As shown in Fig. 2, within the Spatial Perceiver module of SPID, both static and dynamic information are first transformed from their original representations into high-dimensional embeddings that the model can process. The static information is then divided into two separate streams to enhance the model's ability to learn and represent different feature types.

One stream of static embeddings is used as input to the Inferential Decision module of SPID, where a GRU component captures the temporal dependencies and contextual information of the decision sequence, including historical decisions, current states, and constraints to be satisfied. The other stream is combined with dynamic information and passed into the Scaled Dot-Product Attention component, enabling the computation of attention weights through a multi-head attention mechanism.

These two branches complement each other: the GRU is responsible for sequence modeling, while the attention mechanism focuses on feature selection. Together, they guide the model in determining which static features should be emphasized at each decision point, thereby achieving adaptive feature selection and weight assignment.

unedited

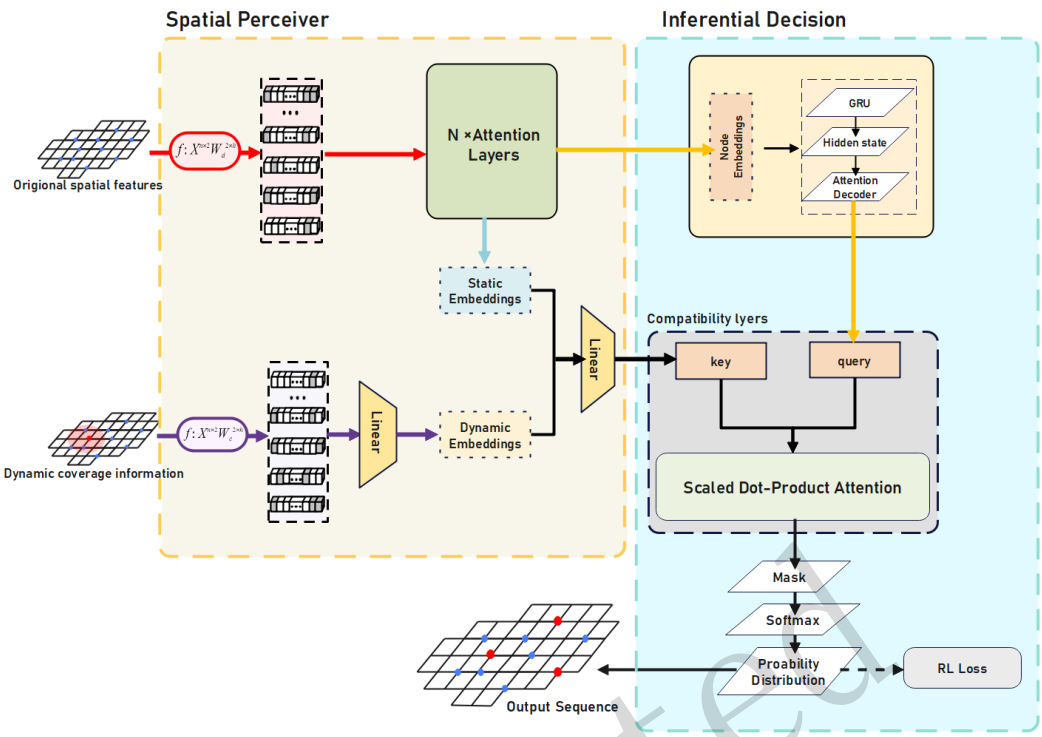


Fig. 2 SPID framework diagram

To ensure solution feasibility, SPID incorporates a masking mechanism. This approach enables the integration of problem constraints and decision-making rules directly into the neural network's inference process. Specifically, SPID employs three types of mask matrices: the access matrix, the service matrix, and the dynamic coverage matrix. These matrices are used to record historical selections, track cumulative demand fulfillment, and represent current decision features, respectively. The current decision features are encoded in the form of a matrix composed of dynamic coverage vectors D_i.

After each siting decision, SPID updates all three mask matrices to ensure that the constraints of the facility location problem are correctly handled. These constraints include capacity limits, service radius restrictions, and the avoidance of duplicate selections.

Finally, the model generates a probability distribution over all candidate points using a Softmax function and selects the point with the highest expected reward as the current decision. The inference and decision-making workflow of SPID is illustrated in Fig. 3.

5 Experiments and results

This chapter presents a thorough experimental validation of the SPID framework for addressing the UAM facility location problem. The evaluation is structured into three progressive phases: (1) Ablation experiments, which assess the contribution of each component to the overall solution quality; (2) Comparison experiments, where SPID is benchmarked against mainstream clustering-based and GNN-based methods; and (3) Simulation experiments, which examine the framework's adaptability across the three development stages of UAM, showcasing its ability to adjust to the evolving UAM operational scenarios. Details of the SPID training framework can be found in Appendix A.

5.1 Ablation experiment result

The SPID framework integrates a multi-layer Graph Attention Network (GAT) encoder with a decoder that combines Gated Recurrent Units (GRU) and an attention mechanism. To isolate and quantify the contributions of each architectural component to the quality and computational efficiency of the UAM facility location solution, ablation experiments were conducted with two model variants: one without the GRU and one without the attention mechanism. The GRU-free variant removes the recurrent context modeling and replaces the GRU update with a simple embedding averaging: $h_t = \text{mean}(v_i : i \in \text{selected})$.

unedited

The attention-free variant removes the adaptive node scoring mechanism and assigns a uniform probability to unselected locations: for all j, $u_j = 0$.

To ensure a fair comparison, all model variants share the same training protocol. The network uses a 128-dimensional embedding, with 3 layers of GAT encoders (each with 8 attention heads), and a 128-dimensional decoder. Training is carried out using the Adam optimizer, with a learning rate of 5×10^{-5}, weight decay of 1×10^{-4}, batch size of 4, and gradient clipping norm of 10. Each model is trained for up to 640 epochs on 1,000 problem instances, with early stopping (patience = 15). Detailed architecture and hyperparameter configurations can be found in Appendix B. Table 3 reports test performance across the 1,000 instances。Fig 4 shows the value function trajectories during training for all three models in the ablation study.

The ablation experiment results demonstrate that the full SPID model performs the best, validating the synergistic contributions of the GRU and attention mechanisms. Specifically, the GRU component improves performance by 0.94% by maintaining sequential context, while the attention mechanism enhances performance by 1.34% through adaptive node selection. Although the ablation variants significantly improve inference speed, the solution quality decreases. However, for UAM infrastructure planning, where facility location decisions have long-term strategic implications, the moderate computational overhead is justifiable.

Training dynamics further clarify the functional roles of each component. The SPID model shows the smoothest convergence trajectory, reaching the optimal validation loss at the 50th epoch and maintaining high stability thereafter. In contrast, the GRU-free variant converges the fastest but ultimately yields slightly lower solution quality, while the attention-free variant experiences a substantial increase in convergence time. Overall, the ablation experiments indicate that both the GRU and attention mechanisms contribute to improved model performance, with the SPID model achieving the best balance between accuracy, stability, and convergence efficiency. Furthermore, SPID demonstrates high consistency between training and testing phases, with a minimal training-test performance gap of just 0.0002, confirming its robustness and generalization capability in solving the facility location problem.

5.2 Comparative experiment result

To comprehensively evaluate the performance of the proposed SPID method, this study conducts a systematic comparison with two categories of baseline methods: traditional clustering-based methods and Graph Neural Network (GNN)-based methods. These baseline approaches are mainstream solutions for location optimization problems and provide a scientific benchmark for assessing the proposed deep reinforcement learning framework.

For the GNN-based methods, each model is trained for 50 epochs using the same training instances as SPID. All GNN models use 128-dimensional hidden representations to match SPID's embedding dimensions and are optimized using the Adam optimizer with the same loss function to ensure a fair comparison.

The problem scenarios for the comparison experiments are set as $p = 5$, $C_j = 60$, $r = 0.3$, and $\sum d_i = 550$, where candidate locations and demands follow a uniform random distribution within the unit square $[0,1]^2$. A total of 1,000 problem instances are generated using these scenario parameters for comprehensive testing. Table 4 presents the performance comparison of all nine methods (SPID, five clustering-based methods, and three GNN-based methods) evaluated on the 1,000 test instances. Figure 5 shows a radar chart of multi-dimensional performance metrics comparing SPID with GNN-based algorithms.

As can be seen, the SPID method performs the best on social indicators S_g, S_e, and S_f. In terms of distance cost control (the C_u indicator), the GCN method outperforms others, with SPID showing a 9.8% increase over GCN. The GRU and attention mechanisms in SPID increase the computational overhead, making its inference time approximately four times that of GCN and 3.2 times that of K-Means. For applications that require real-time dynamic adjustments, such as rapid deployment of temporary facilities in disaster emergency responses, this computational delay could become a practical limitation.

The multi-dimensional radar chart further highlights SPID's comprehensive advantages in facility location problems. From the chart, it is evident that SPID forms a significantly expanded polygon in the

unedited

dimensions of S_g, S_e, and S_f, with facility utilization approaching an ideal state. This indicates that SPID not only achieves broad coverage but also ensures the efficient allocation of resources, avoiding the dilemma of "high coverage but wasted capacity" or "high utilization but insufficient coverage." In contrast, while GraphSAGE shows competitiveness in distance indicators ($1/C_w$ and $1/C_u$), its noticeable contraction in coverage dimensions exposes the flaw of over-optimizing operational costs at the expense of service equity. This difference is particularly crucial in public facility planning: for essential services such as hospitals and schools, ensuring service accessibility should take priority over mere cost minimization. From a long-term investment return perspective, the adaptability and stability of SPID's location solutions provide stronger robustness in the face of uncertainties like population distribution changes and demand fluctuations, reducing the risk of future replanning while promoting greater social equity.

As a public infrastructure, UAM takeoff and landing sites are evaluated based on two core indicators: coverage and effective coverage, which reflect the scientific quality of their location selection. As shown in Figure 6, the SPID location results significantly outperform other models in terms of coverage performance. Under the same facility quantity and capacity constraints, SPID can serve almost all candidate locations, with its geographical coverage improving by nearly 20 percentage points compared to the second-best model, Spectral. The lower service blind spot ratio of the SPID solution makes it more aligned with the service attributes required for public facilities. In terms of effective coverage, SPID maintains its leading position, although the margin of advantage over other methods has narrowed.

The simultaneous achievement of high geographical coverage and high effective coverage validates that SPID's sequential decision-making mechanism can strike the optimal balance between spatial layout and capacity allocation. This capability is especially valuable when facing future dynamic demand changes. For example, when population growth in a certain area leads to a surge in demand, the initial location selection already covers that area (with a high S_g value), and only local capacity adjustments are needed, avoiding the need for a complete re-selection of locations. This makes SPID's location results more strategically forward-looking and better suited to urban infrastructure development requirements.

Additionally, the narrow box plot combined with the high median distribution pattern of SPID indicates that SPID not only performs excellently but also demonstrates high stability in solution quality when solving problems with different characteristics. In contrast, the second-best method, Spectral, shows high sensitivity to the characteristics of the problem instances, with its interquartile range (IQR) value being four times that of SPID and its range reaching approximately 40 percentage points. To further quantify the robustness of each model, the coefficient of variation (CV = σ/μ) is used as a standardized volatility indicator. SPID's geographical coverage CV is approximately 0.021, significantly lower than Spectral's CV of 0.054, GraphSAGE's CV of 0.071, and DBSCAN's CV of 0.137. This further proves that SPID has strong robustness to input data perturbations and the heterogeneity of problem instances. It indicates that the SPID model is better able to handle demand forecast errors, spatial data noise, and dynamic environmental changes.

In conclusion, SPID can achieve an optimal trade-off between unconstrained spatial coverage and constrained capacity distribution. The reliability and long-term value of its solutions are evident in both theoretical research and practical deployment.

5.3 Simulations Across Development Stages

5.3.1 Experimental configuration

Low-altitude aviation is currently in its early development stage, during which the operational capacity of takeoff and landing platforms is generally sufficient to meet demand. However, as the sector evolves, the supply-demand balance is expected to shift—from an initial state of oversupply to relative scarcity, and eventually to undersupply. To capture this progression, this paper designs three experimental scenarios that simulate the early, middle, and mature stages of low-altitude aviation. Each scenario focuses on the facility location problem for low-altitude flight services under varying demand conditions.

For theoretical validation, these scenarios are used to assess the model's performance under different constraint configurations: (1) flight radius

unedited

constraints only, (2) capacity constraints only, and (3) both constraints applied simultaneously. All experiments in this section are solved using both SPID and Gurobi. Gurobi is widely recognized in operations research and industry as one of the leading commercial mathematical programming solvers, particularly for mixed-integer optimization problems. It therefore provides a reliable reference solution for evaluating the accuracy and quality of SPID's outputs. A summary of the experimental design and corresponding analysis is provided in Table 5.

The experimental environment is as follows: operating system, Windows 11.0 (22631.2) on a 64-bit Windows platform; CPU model, 12th Gen Intel(R) Core (TM) i9-12900H; and Gurobi Optimizer version 11.0.3 build v11.0.3rc0. The coding language is the Python programming language. The training loss plots for the three stages are shown in Appendix C.

5.3.2 Initial stage simulation experiment

This subsection examines the facility location problem during the early stage of low-altitude aviation, where facility capacity is sufficient. At this stage, key constraints include the spatial distribution of demand points and UAV flight limitations. As this scenario aligns with current practical needs, it is used to assess SPID's effectiveness in facility location optimization.Comparative experiments are conducted on three problem sizes (20, 50, and 100 nodes), with SPID benchmarked and the exact solver Gurobi.

All algorithms yielded feasible solutions within acceptable time. Fig. 7 shows node configurations, and Table 6 presents the results. For the 100-node case, 50 additional simulations with varying capacities and demand distributions were run. Selected results and summary charts are shown in Fig. 8

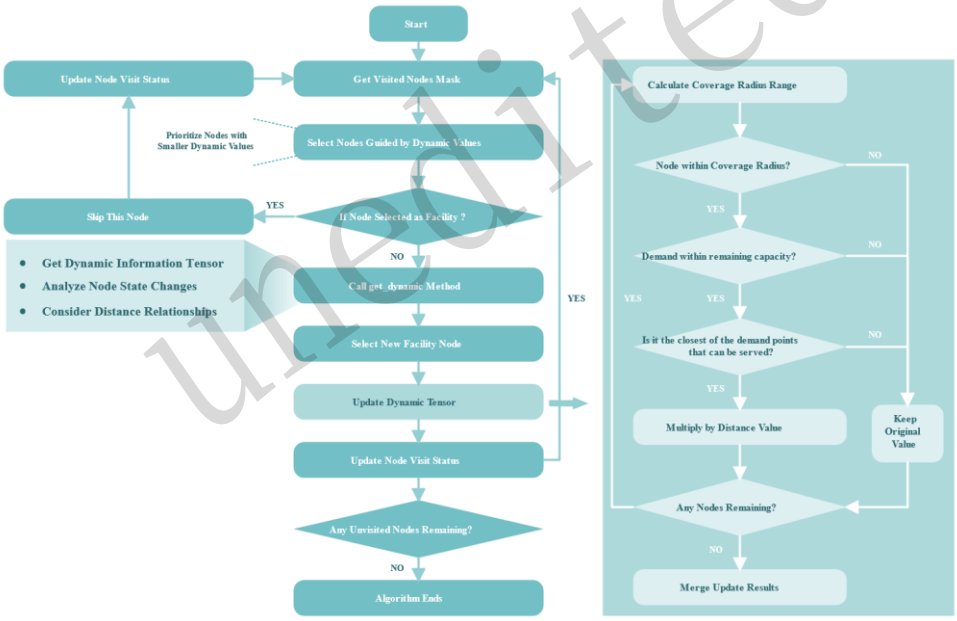
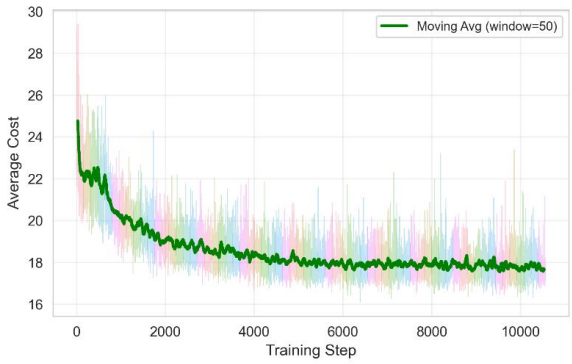


Fig. 3 SPID RL state workflow chart

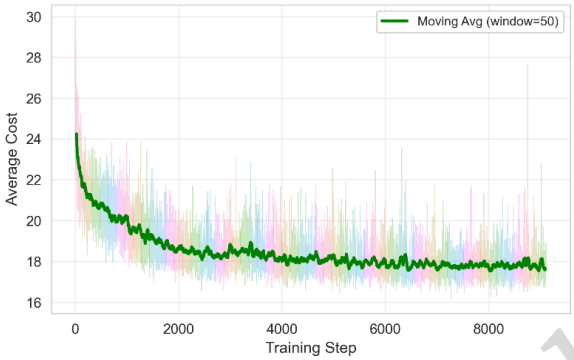
Table 3 Test Set Performance Table

Model	Average loss	Standard deviation	Minimum value	Maximum value	Time (milliseconds)
Complete model	**17.4123**	0.5234	15.8923	19.2341	23.4
No GRU	17.4287	0.5412	**15.8734**	19.3456	**19.8**
Lack of attention	17.4356	**0.5123**	15.9123	19.1234	**18.7**

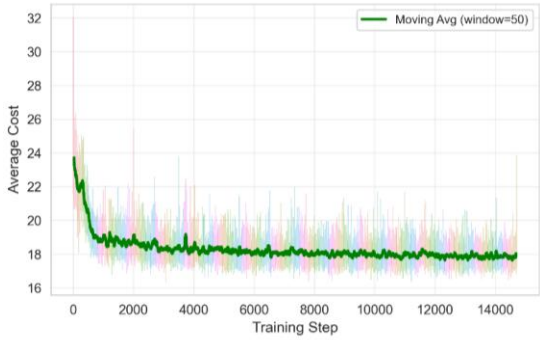
unedited



(a)



(b)



(c)

Fig. 4 Model Training Value Function Curve

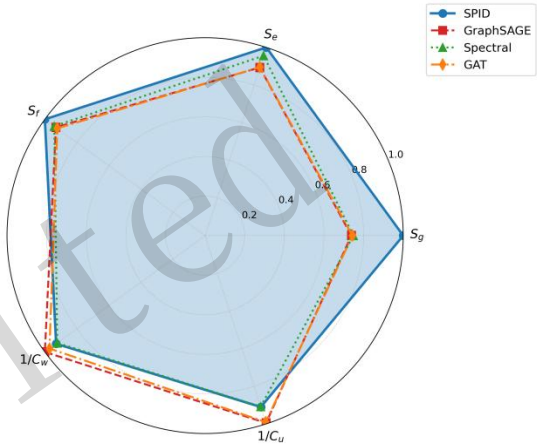


Fig. 5 Multi-dimensional performance radar chart

unedited

Table 4 Comprehensive Performance Comparison Across All Methods

Method	S_g (%)	S_e (%)	S_f (%)	C_w	C_u	Inference Time (ms/instance)	Training Time (hours)
SPID (Ours)	**95.02**	**57.10**	**98.25**	35.45	7.17	28.5	1.38
GraphSAGE	70.16	51.02	91.45	**32.96**	6.56	15.2	1.29
GAT	70.64	51.11	90.60	33.95	6.59	12.8	1.41
GCN	70.06	50.86	91.52	34.07	**6.53**	**7.1**	0.95
Spectral	73.89	54.58	92.45	36.21	7.32	42.3	-
Hierarchical	70.82	54.32	92.01	36.15	7.29	38.6	-
K-Means	69.34	53.86	91.23	36.08	7.28	8.9	-
K-Medians	68.97	53.67	90.98	36.12	7.31	9.2	-
DBSCAN	52.18	38.42	72.15	45.67	9.12	156.7	-

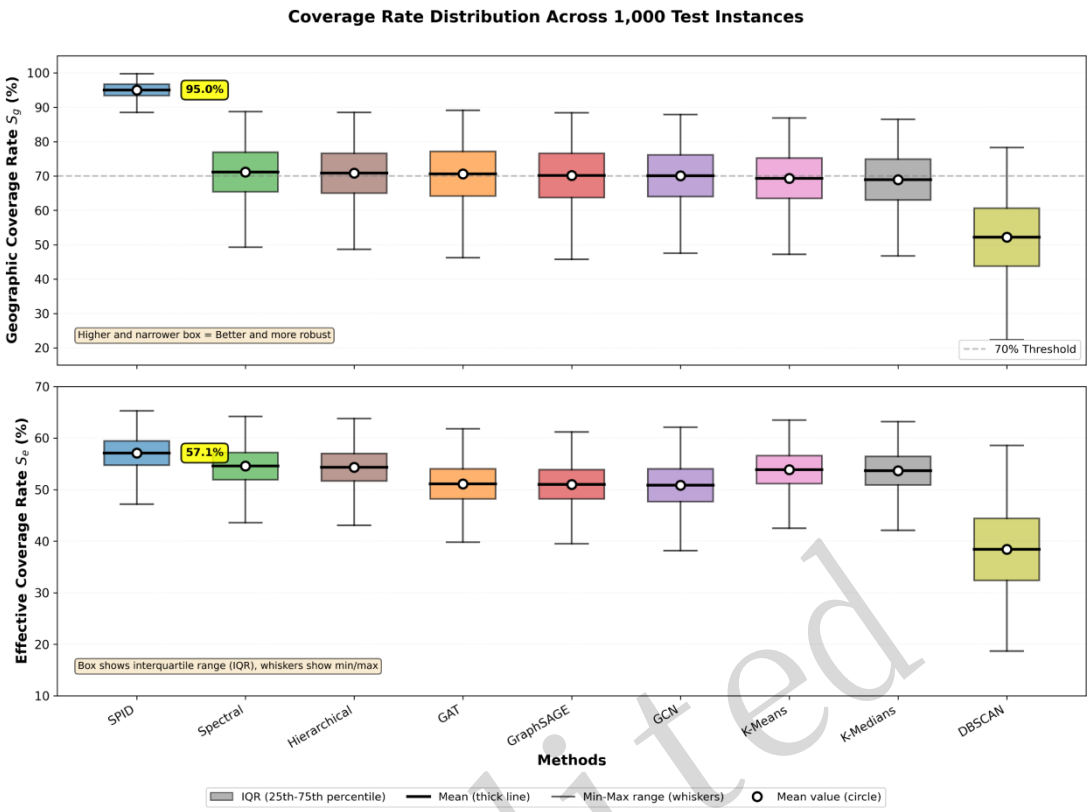


Fig. 6 Comparison chart of coverage rate and effective coverage rate

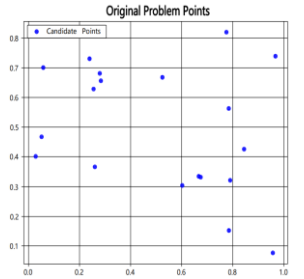
Table 5. Experimental chapter Settings Summary Table

Stage of development	Supply-Demand Ratio	Key constraints	Chapter
Early Stage	Sufficiency	Flight radius	5.3.2
Intermediate Stage	Tight	Both	5.3.3
Rapid growth phase	Shortage	Capacity	5.3.4

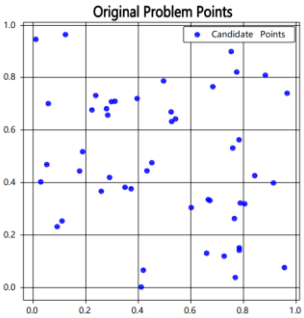
Table 6 Experimental Result Table

Problem	Algorithm	Objective function value	Solving Time	Gap
20 nodes 3 facility points Total demand 113.92 Individual facility capacity 60	SPID	3.2257	/	-0.002
	Gurobi	**3.2237**	0.019	/
50 nodes 4 facility points Total Demand 279.98 Individual facility capacity 100	SPID	**7.8333**	/	0
	Gurobi	**7.8333**	0.101s	/
100 nodes 5 facility points Total demand 554.97 Individual facility capacity 120	SPID	16.0871	/	-0.0861
	Gurobi	**16.0010**	2.64s	/

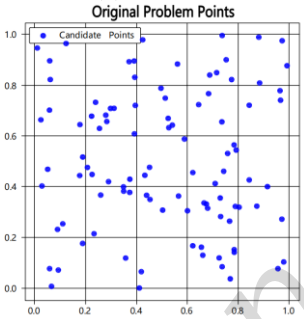
unedited



(a) 20 nodes



(b) 50 nodes



(c) 100 nodes

Fig. 4 Schematic diagram of experimental node Settings

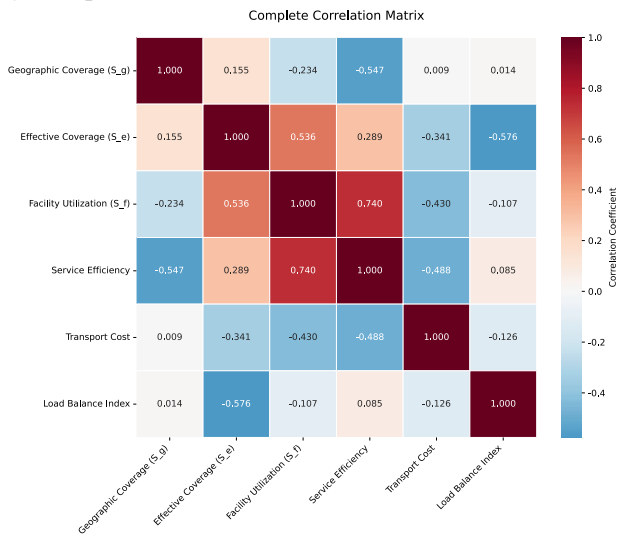
Experimental results highlight three key advantages of SPID in facility location optimization: (1) Stable near-optimal performance across different problem sizes; (2) Strong multi-objective balancing, particularly in coverage and resource utilization; (3) Superior computational efficiency compared to exact algorithms, making it more suitable for large-scale applications.

While slightly less effective than Gurobi in minimizing transportation cost, SPID offers better overall performance in service quality and runtime, enhancing its practical value. Ranging 20 to 100 nodes, SPID consistently maintained solution gaps below 0.1. Gurobi achieved lower transportation cost (31.03 vs. 49.83, –37.73%) and shorter total distance (6.64 vs. 11.49, –42.18%), but SPID excelled in demand coverage (50.62% vs. 48.36%) and facility utilization (93.63% vs. 89.46%), reflecting its strength in balancing efficiency and service. SPID also demonstrated strong computational performance, completing solutions within 6 seconds. Gurobi failed to solve without initialization in 1 hour, but when using SPID's output as a starting point, its time dropped to 0.71 seconds—proving the quality of SPID's solutions and the value of DRL-based acceleration.

To assess the stability of SPID's solutions, multiple experiments were conducted alongside a correlation analysis (Fig. 8). Results revealed strong correlations among key metrics. Geographic and effective coverage were highly correlated (r = 0.734), as were facility utilization and service efficiency (r = 0.740).

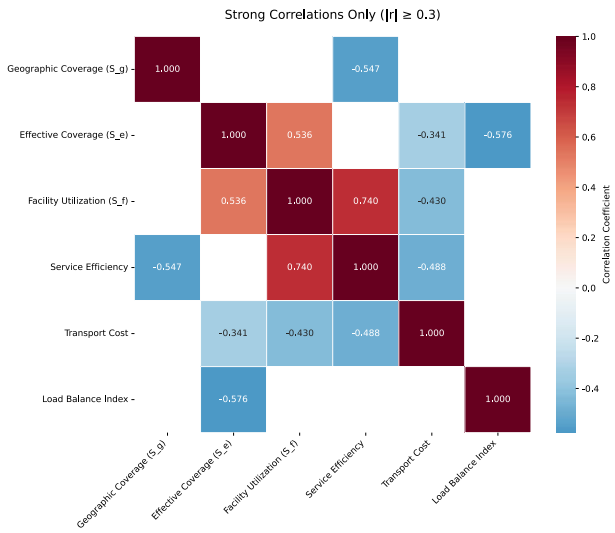
In contrast, transport cost was negatively correlated with most indicators, particularly geographic coverage (r = −0.341) and effective coverage (r = −0.311), suggesting that higher costs reduce service reach. The Load Balance Index showed weaker correlations overall, with its most significant being a negative relationship with service efficiency (r = −0.276), implying that uneven load distribution may hinder performance.

Further analysis(Fg.8b) of strong correlations ($|r| \geq 0.3$) confirmed two key patterns: (1) geographic and effective coverage are tightly linked, and (2) facility utilization is closely associated with service efficiency. These findings provide theoretical support for optimizing service configuration and improving system performance.



(a)

unedited

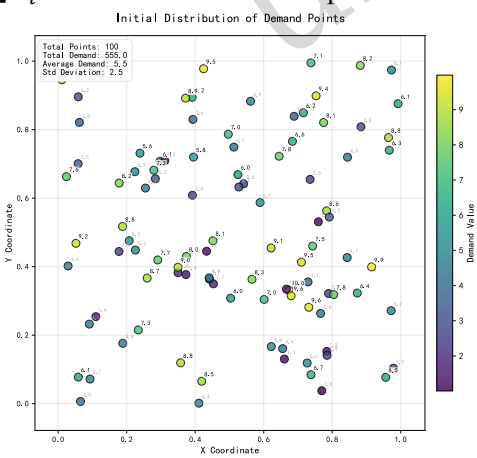


(b)

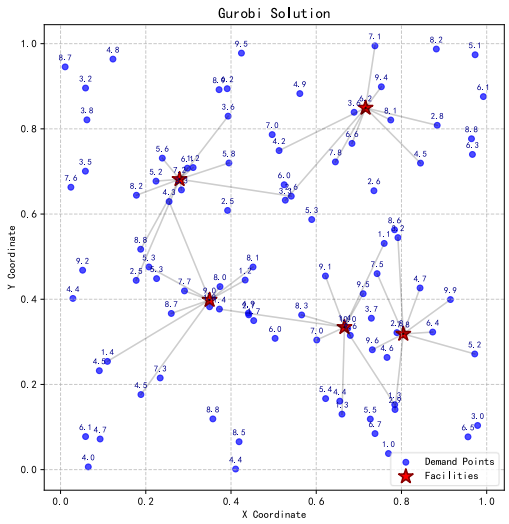
Fig. 5 Schematic diagram of experimental node Settings

5.3.3 Mid-term stage simulation experiment

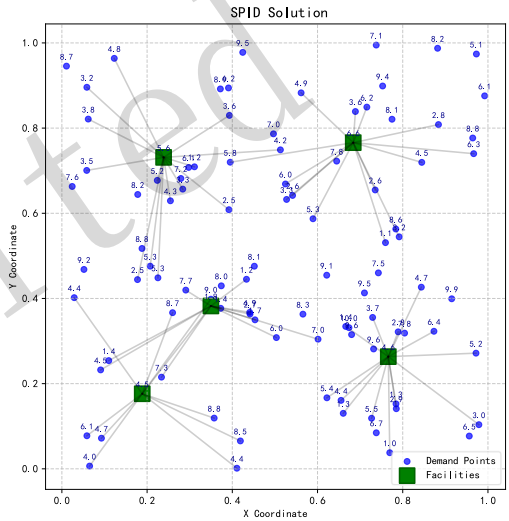
This subsection focuses on the facility location problem during the mid-stage of low-altitude aviation development. At this stage, facility capacity becomes limited, and capacity-related constraints begin to play a critical role in location decisions. To assess the effectiveness and solution performance of SPID under these conditions, a comparative experiment was conducted against the exact solver Gurobi using a 100-node problem instance. The experimental parameters were set as follows: $p = 5, C_j = 60, r = 0.3, \sum d_i = 555.0$. The results are presented in Fig 9.



(a)



(b)



(c)

Fig. 6 Mid-term stage simulation experiment result figure

Experimental results show that SPID achieved a transportation cost of 49.83, a total distance of 11.49, a demand coverage rate of 50.62%, and a facility utilization rate of 93.63%. In comparison, Gurobi achieved lower transportation cost (31.03) and shorter distance (6.64), with slightly lower demand coverage (48.36%) and facility utilization (89.46%).

SPID outperformed Gurobi in service-related metrics, with higher coverage (+2.26%) and utilization (+4.17%), while Gurobi performed better in cost (–37.73%) and distance (–42.18%). This reflects SPID's strength in balancing service quality and resource use, and Gurobi advantage in cost efficiency.

In terms of computational efficiency, SPID completed optimization in about six minutes. Gurobi, without an initial solution, failed to converge within

unedited

one hour. When initialized with SPID's output, it reached a global optimum in just 0.71 seconds (MIP gap = 0.0003), confirming the high quality of SPID's solutions and their value in accelerating exact solvers.

5.3.4 Fast growth stage simulation experiment

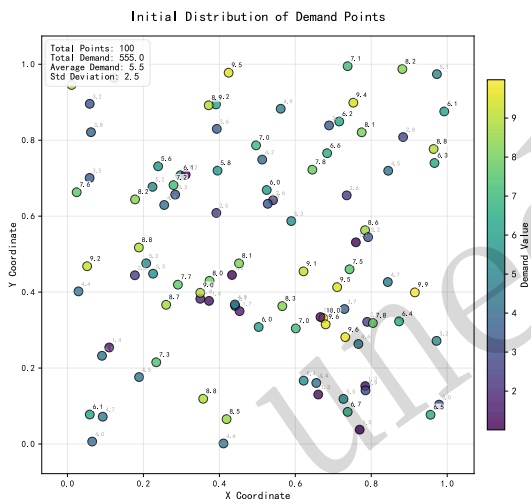
This subsection focuses on the facility location problem during the rapid growth phase of low-altitude aviation. At this stage, facility capacity becomes increasingly constrained, while the endurance of low-altitude aircraft is expected to improve significantly. As a result, capacity-related constraints become the primary factor influencing location decisions. To assess SPID's effectiveness under these conditions, comparative experiments were conducted using the exact solver Gurobi on a 100-node problem instance. A summary of all results is provided in Table 7. Two experimental scenarios were designed:

Scenario 1: $p = 5, C_j = 60, r = 1.2, \sum d_i = 550.0$; results are presented in Fig. 10c.

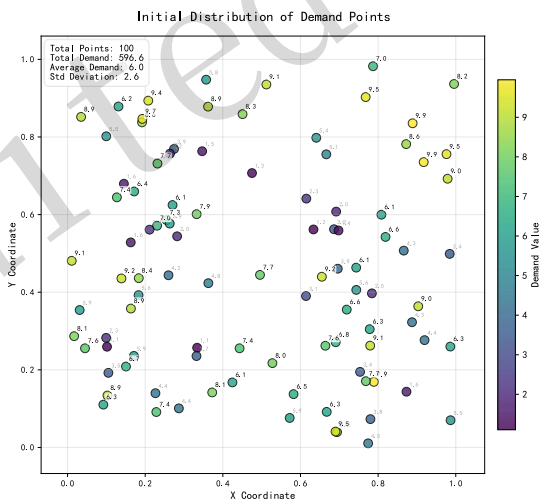
Scenario 2: $p = 5, C_j = 60, r = 1.2, \sum d_i = 596.6$; results are shown in Fig. 10d.

Experimental results reveal distinct performance differences between SPID and Gurobi in the rapid growth phase scenarios. SPID yielded higher objective function values and longer total distances but achieved better effective coverage and facility utilization. It also provided more balanced demand allocation; for instance, in Fig. 10c, SPID covered third-quadrant demand points that Gurobi did not. This supports earlier findings that higher demand satisfaction may increase overall cost.

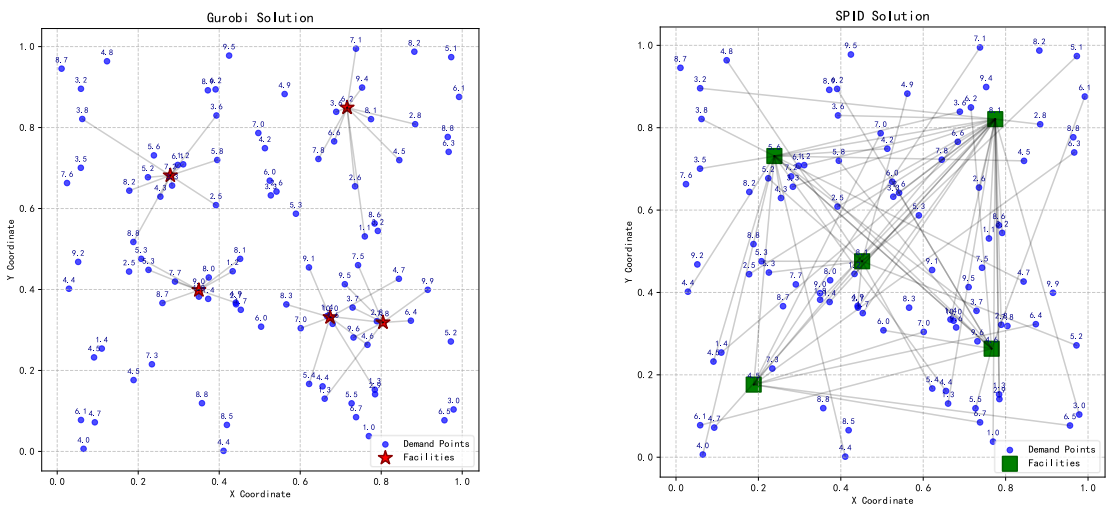
unedited



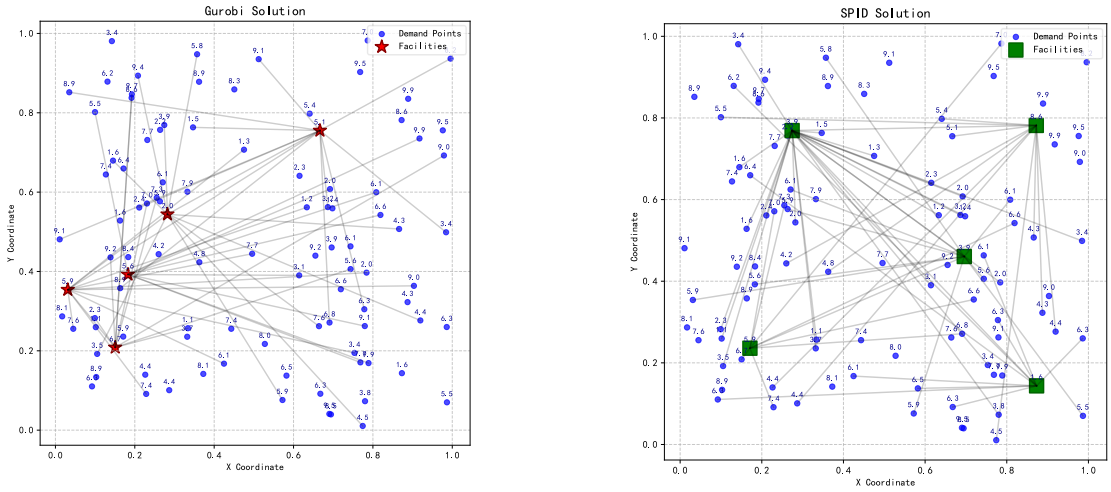
(a) Scenario 1



(b) Scenario 2



(c) Solution result of Scenario 1



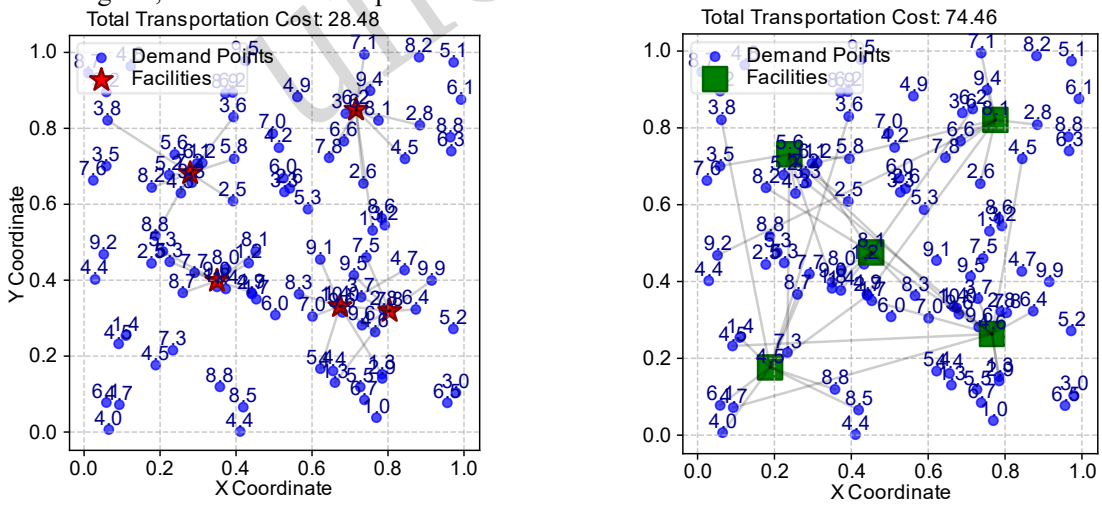
(d) Solution result of Scenario 2

Fig. 7 Fast growth stage simulation experiment result figure

Table 7 . Experimental Result Table

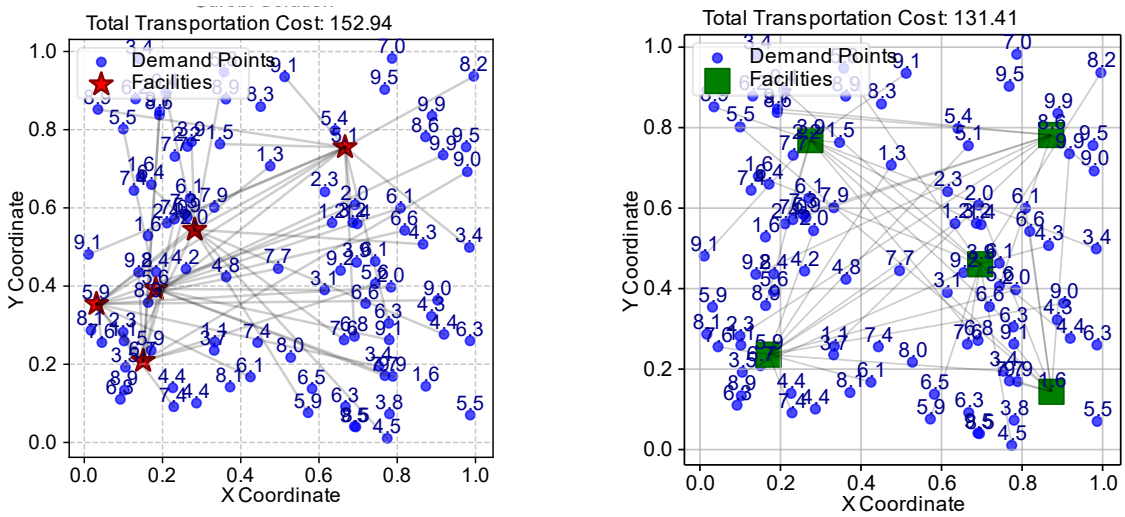
Experimental Result	Algorithm	Objective function value	Total distance	Effective Coverage	Facility utilization rate
Scenario 1	SPID	145.87	5.53	46.78%	86.54%
	Gurobi	28.28	36.68	49.29%	96.07%
Scenario 2	SPID	128.86	30.54	45.07%	94.76%
	Gurobi	152.94	25.71	47.57%	99.20%

As the facility coverage radius increases, overlapping service areas make allocation strategy more critical. Under capacity constraints, SPID showed limited ability to handle multi-coverage conflicts. As shown in Fig. 11, when Gurobi re-optimized allocations based on SPID's facility placements, objective values dropped significantly (very close to Gurobi original results). This suggests SPID performs well in site selection but could improve in allocation refinement.



(a) Solution results after redistribution in Scenario 1

Unedited



(b) Solution results after redistribution in Scenario 2

Fig. 8 Gurobi resolved experiment result figure

Under the experimental condition of r=1.2, all demand points in the plane are covered by at least two facilities. When Gurobi is used for exact demand allocation based on SPID's facility locations, the gap in objective function values between the two methods narrows significantly. In Scenario 2, where total demand nearly doubles the total facility capacity, the two approaches yield highly similar objective values. These reallocation results indirectly confirm the strength of SPID's site selection, indicating that its performance gap stems primarily from limitations in the allocation strategy rather than location accuracy.

6 Conclusions

This study presents a novel solution to the low-altitude facility location problem by reformulating the classical capacity-constrained combinatorial optimization as a Markov Decision Process, solved via deep reinforcement learning. The proposed SPID model demonstrates consistently strong performance in siting decisions. Notable performance differences with the exact solver Gurobi only arise when allocation quality becomes critical—typically under scenarios with extremely wide facility coverage. In such cases, applying Gurobi to re-optimize demand allocation based on SPID's siting decisions reduces the performance gap to approximately 3%, thereby validating the robustness of SPID's site selection strategy. SPID also offers notable advantages in computational efficiency. While training requires a modest time investment, inference with a pretrained model solves new problems in under six minutes—substantially outperforming Gurobi, which may fail to return a solution within an hour without a feasible starting point. Overall, SPID delivers reliable site selection performance under both standard and challenging conditions, even in the absence of a dedicated allocation sub-model. Its emphasis on maximizing service coverage and promoting equitable distribution makes it particularly suitable for public service applications and regulatory contexts that value accessibility over cost minimization. Future enhancements may include integrating an allocation module to further improve performance in complex allocation scenarios.

This study makes several simplifying assumptions in both the problem formulation and the solution setup. On the modeling side, the framework assumes a steady-state demand distribution appropriate for long-term infrastructure planning. It does not explicitly capture exogenous demand fluctuations driven by time-of-day patterns, seasonal variation, or special events. The current formulation also omits several practical deployment constraints, including regulatory requirements (airspace restrictions, zoning limits), environmental impacts (noise and energy consumption), and three-dimensional airspace coordination. These factors are critical for real-world deployment but can be treated as additional constraint layers. Future work may incorporate them by extending the state space or by formulating the problem as a multi-objective optimization task.

On the solution side, the main objective of this

unedited

study is to assess the potential of DRL as an alternative to traditional operations research techniques for solving mixed-integer linear programs (MILPs). The goal is not to claim superiority over all possible DRL approaches. For this reason, SPID is compared against baselines that reflect established practice in facility location problems: exact solvers, heuristic algorithms, and graph neural network–based methods. We do not benchmark against more general-purpose DRL algorithms such as Proximal Policy Optimization or Soft Actor-Critic, which are designed for broader classes of control and decision-making tasks. A systematic comparison with such algorithms is an important direction for future work in evaluating DRL-based MILP solvers.

In summary, the SPID framework can be enhanced along several dimensions. Potential extensions include modeling temporal dynamics, integrating an explicit demand allocation module, and incorporating more realistic deployment constraints. These additions would increase its practical applicability. Future work should also include systematic comparison with state-of-the-art DRL methods developed for combinatorial optimization, to better characterize performance limits and identify opportunities for architectural improvement. Thus, this study should be viewed as an initial step toward a comprehensive planning system for urban air mobility infrastructure, rather than as a complete deployment-ready solution.

unedited

Contributors

Xiaocheng LIU conducted the investigations, developed the methodologies, designed the software, and drafted the original paper. Meilong LE provided supervision, contributed to the conceptualization, and reviewed and edited the manuscript. Yupu LIU contributed to methodology development, supervised the work, participated in conceptualization, and reviewed and edited the paper. Minghua HU provided resources and additional supervision.

Conflict of interest

All the authors declare that they have on conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abdel-Basset, M., Mohamed, R., Hezam, I.M., et al., 2024. Multiobjective trajectory optimization algorithms for solving multi-UAV-assisted mobile edge computing problem. Journal of Cloud Computing, 13:35. https://doi.org/10.1186/s13677-024-00594-z

Al-Rabiaah, S., Hosny, M., AlMuhaideb, S., 2022. An efficient greedy randomized heuristic for the maximum coverage facility location problem with drones in healthcare. Applied Sciences, 12(3):1403. https://doi.org/10.3390/app12031403

Apaza, R.D., Li, H., Han, R., Knoblock, E., 2023. Multi-agent deep reinforcement learning for spectrum and air traffic management in UAM with resource constraints. In: 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), pp. 1–7. IEEE.

Aurambout, J.-P., Gkoumas, K., Ciuffo, B., 2019. Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities. European Transport Research Review, 11(1): (article no. unspecified).

Garcia, C.P., Li, W., Hirata, N.S.T., Neumann, C., 2023. IS-UAM: Intelligent and safe UAM with deep reinforcement learning. In: 2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS), pp. 378–383. IEEE.

Gopi, S.P., Magarini, M., 2021. Reinforcement learning aided UAV base station location optimization for rate maximization. Electronics, 10:2953. https://doi.org/10.3390/electronics10232953

Goyal, R., Cohen, A., Serrao, J., Kimmel, S., Fernando, C., Shaheen, S., 2018. Urban Air Mobility (UAM) market study. National Aeronautics and Space Administration (NASA), Washington, DC. Available from https://ntrs.nasa.gov/api/citations/20190001472/downloads/20190001472.pdf

Han, R., Li, H., Apaza, R., Knoblock, E., Gasper, M., 2022. Deep reinforcement learning assisted spectrum management in cellular-based urban air mobility. IEEE Wireless Communications, 29(6):14–21.

Han, X., Xie, M., Yu, K., Huang, X., Du, Z., Yao, H., 2024. Combining graph neural network with deep reinforcement learning for resource allocation in computing force networks. Frontiers of Information Technology & Electronic Engineering, 25(5):701–712. https://doi.org/10.1631/FITEE.2300009

Hottung, A., Tierney, K., 2020. Neural large neighborhood search for the capacitated vehicle routing problem. European Journal of Operational Research, 294(3):853–868.

Khalife, C., Slim, J., Escribano Macias, J., Angeloudis, P., 2021. An integrated vertiport placement model considering vehicle sizing and queuing. arXiv preprint arXiv:2107.13332.

Kim, D., Lee, K., Moon, I., 2019. Stochastic facility location

model for drones considering uncertain flight distance. Annals of Operations Research, 283(1):1283–1302.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA. (arXiv:1412.6980)

Kool, W., Van Hoof, H., Welling, M., 2019. Attention, learn to solve routing problems! In: International Conference on Learning Representations (ICLR).

KrisshnaKumar, P., Witter, J., Paul, S., Chowdhury, S., 2023. Graph learning-based decision support for multi-aircraft take-off and landing at urban air mobility vertiports. In: AIAA SciTech 2023 Forum, National Harbor, MD, 23–27 Jan 2023.

Lee, S., Cho, N., 2025. Optimal location of urban air mobility (UAM) vertiport using a three-stage geospatial analysis framework. Future Transportation, 5:58. https://doi.org/10.3390/futuretransp5020058

Li M, Zhao P, Bao Z, et al. A Reinforcement Learning-Based Iterative Method for Capacitated Hub Location Problems in UAV Networks. 2025 Integrated Communications, Navigation and Surveillance Conference (ICNS), IEEE, 2025: 1-11. https://doi.org/10.1109/ICNS.2025.00000

Liang, H., Wang, S., Li, H., Zhou, L., Chen, H., Zhang, X., Chen, X., 2024. SPONet: Solve spatial optimization problem using deep reinforcement learning for urban spatial decision analysis. International Journal of Digital Earth, 17(1):2299211. https://doi.org/10.1080/17538947.2023.2299211

Lynskey, J., et al., 2019. Facility location problem approach for distributed drones. Symmetry, 11(1):118.

Mahmoodi, A., Sajadi, S.M., Sadeq, A.M., et al., 2025. Enhancing unmanned aerial vehicles logistics for dynamic delivery: A hybrid non-dominated sorting genetic algorithm II with Bayesian belief networks. Annals of Operations Research. https://doi.org/10.1007/s10479-025-06504-z

Meng, Z., Yu, K., Qiu, R., 2024. Location-routing optimization of UAV collaborative blood delivery vehicle distribution on complex roads. Complex & Intelligent Systems, 10:8127–8141. https://doi.org/10.1007/s40747-024-01591-0

Meskar, M., Ahmadi-Javid, A., 2024. Optimizing drone delivery paths from shared bases: A location-routing problem with realistic energy constraints. Journal of Intelligent & Robotic Systems, 110:142. https://doi.org/10.1007/s10846-024-02129-9

Nazari, M., Oroojlooy, A., Snyder, L., Takáč, M., 2018. Reinforcement learning for solving the vehicle routing problem. In: Advances in Neural Information Processing Systems (NeurIPS), 31:9839–9849.

Paul, A., Levin, M.W., Waller, S.T., Rey, D., 2024. Data-driven optimization for drone delivery service planning with online demand. arXiv preprint arXiv:2404.02442.

Park, C., Kim, G.S., Park, S., Jung, S., Kim, J., 2023. Multi-agent reinforcement learning for cooperative air transportation services in city-wide autonomous urban air mobility. IEEE Transactions on Intelligent Vehicles, 8(8):4016–4030.

Premkumar, V.G.R., Van Scoy, B., 2025. Optimal positioning of unmanned aerial vehicle (UAV) base stations using mixed-integer linear programming. Drones, 9:44. https://doi.org/10.3390/drones9010044

Rahman, M.A., Basheer, M.A., Khalid, Z., Tahir, M., Uppal, M., 2023. Logistics hub location optimization: A K-means and P-median model hybrid approach using road network distances. arXiv preprint arXiv:2308.11038. https://doi.org/10.48550/arXiv.2308.11038

Ryu, K., Kim, W., 2024. Energy-efficient deployment of aerial base stations for mobile users in multi-hop UAV networks. Ad Hoc Networks, 157:103463. https://doi.org/10.1016/j.adhoc.2024.103463

Shavarani, S.M., Nejad, M.G., Rismanchian, F., et al., 2018. Application of hierarchical facility location problem for optimization of a drone delivery system: A case study of Amazon Prime Air in the city of San Francisco. International Journal of Advanced Manufacturing Technology, 95:3141–3153. https://doi.org/10.1007/s00170-017-1363-1

Shavarani, S.M., Mosallaeipour, S., Golabi, M., İzbirak, G., 2019. A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system. Computers & Operations Research, 108:57–68. https://doi.org/10.1016/j.cor.2019.04.001

Song, K.-H., Lee, H., 2025. Network topology-driven vertiport placement strategy: Integrating urban air mobility with the Seoul Metropolitan Railway System. Applied Sciences, 15:3965. https://doi.org/10.3390/app15073965

Sun, J., et al., 2025. Location optimization of unmanned aerial vehicle (UAV) drone port for coastal zone management: The case of Guangdong coastal zone in China. Ocean & Coastal Management, 262:107576.

U.S. Department of Transportation, Federal Aviation Administration (Fontaine, P., et al.), 2023. Urban Air Mobility (UAM) Version 2.0 Concept of Operations. Washington, DC, 26 Apr 2023.

Volakakis, V., Mahmassani, H.S., 2024. Vertiport infrastructure location optimization for equitable access to urban air mobility. Infrastructures, 9:239. https://doi.org/10.3390/infrastructures9120239

Zhang C, Du W, Guo T, et al. Multi-objective hub location for urban air mobility via self-adaptive evolutionary algorithm. Advanced Engineering Informatics, 2025, 64: 102974. https://doi.org/10.1016/j.aei.2025.102974

Zhao, Y., Feng, T., 2024. Strategic integration of vertiport planning in multimodal transportation for urban air mobility: A case study in Beijing, China. Journal of Cleaner Production, 467:142988. https://doi.org/10.1016/j.jclepro.2024.142988

Zheng, L., Xu, G., Chen, W., 2024. Using improved particle swarm optimization algorithm for location problem of

unedited

drone logistics hub. Computers, Materials & Continua, 78(1):935–957. https://doi.org/10.32604/cmc.2023.046006

Zhu, T., Boyles, S.D., Unnikrishnan, A., 2022. Two-stage robust facility location problem with drones. Transportation Research Part C: Emerging Technologies, 137:103563. https://doi.org/10.1016/j.trc.2022.103563

unedited