Hybrid Particle Swarm Optimization and Q-Learning for Airport Parking Space Allocation and Scheduling

Chunxin Huang
College of Civil Aviation, Shenyang Aerospace University, Shenyang 110136, China
E-mail: chunxin_huang@hotmail.com

Keywords: airport parking space allocation, scheduling optimization, metaheuristic algorithm, particle swarm optimization, reinforcement learning

Received: February 20, 2025

Abstract: With the rapid development of the aviation industry, the contradiction between the shortage of airport parking space resources and the continuous growth of air transportation demand has become increasingly prominent. Traditional parking space allocation and scheduling methods have been unable to cope with the increasingly complex and dynamic operating environment. To address this challenge, this paper proposes an airport parking space allocation and scheduling optimization model based on a meta-heuristic algorithm, combining the particle swarm optimization (PSO) algorithm with the Q-learning reinforcement learning method, aiming to improve the utilization efficiency of parking space resources and the level of intelligent scheduling. The method uses PSO to examine at the whole scheduling space and Q-learning to make adjustments to allocations depending on feedback from the environment in real time. In terms of research methods, we first constructed a mathematical model with multiple constraints and a comprehensive objective function, used the PSO algorithm to perform preliminary allocation of parking spaces, and introduced an adaptive mechanism to enhance the search capability. At the same time, the Q-learning model continuously optimizes scheduling decisions through interaction with the environment to ensure the optimal balance between the global and local. The hybrid approach enhances both global search and local optimization. The results show that this method is superior to individual PSO, Q-learning and traditional heuristic methods in multiple key indicators, including total scheduling cost, delay time, parking space utilization, algorithm convergence speed, number of scheduling conflicts, calculation time and successful scheduling rate. By coordinating factors such as cost, time and safety, the model can significantly improve airport operating efficiency, reduce flight delays and optimize resource allocation. With the CloudSim toolkit to run tests in a simulated cloud environment shows that our strategy cuts the average task latency by 15.2% and the overall scheduling cost by 12.5% compared to classic PSO and heuristic methods. The suggested approach works most effective when there are constraints on items like resource capacity, task deadlines, and energy use. The evaluation measures, which include makespan, cost, and delay time, show that the hybrid strategy works well and is strong.

Povzetek: Članek predstavi hibridni model PSO in Q-learning za razporejanje parkirnih mest na letališčih. Metoda združuje globalno iskanje in lokalno optimizacijo, izboljšuje učinkovitost, zmanjšuje zamude ter povečuje zanesljivost razporejanja.

1 Introduction

Airport operations and management are under more strain than ever in today's aviation business. As more and more people want to fly, airports' infrastructure and service capacities have been pushed to the limit [1]. In example, when it comes to assigning and arranging parking spaces, a lack of efficiency and accuracy can cause flights to be delayed, resources to be wasted, and operating costs to rise unnecessarily. Airlines and airport operators have to figure out how to best use the parking spots they have so that they may make the most of their resources and provide good service. A reasonable design of parking spaces at an airport can not only keep things running smoothly, but it can also improve the airport's overall efficiency and the experience of passengers [2, 3].

There isn't enough parking space at airports throughout the world, but more and more people want to fly. This difference between what people want and what is available has become a major problem that has to be fixed. More and more studies are looking into optimization algorithms as a way to solve this difficulty. Among these solutions, metaheuristic algorithms have become a popular area of research since they are very good at searching the whole space and can adapt to new situations [4]. Adding metaheuristic algorithms may make scheduling and assigning parking spaces much more efficient, especially when you have to take into account flights, types of aircraft, the environment, and other complicated considerations [5].

At present, the meta-heuristic algorithms used in airport parking space allocation mainly include genetic algorithms, particle swarm optimization algorithms, ant colony algorithms, etc. These algorithms can effectively solve the parking space allocation problem by simulating physical and biological processes in nature. For example, genetic algorithms search the solution space through operations such as crossover and mutation to obtain the global optimal solution [6]; while particle swarm optimization algorithms simulate the foraging behavior of bird flocks and use the information in the group to guide the search direction. Although these algorithms have been successfully applied in many studies, they also face the problem of local optimal solutions, which to some extent limits their wide application [7].

A lot of metaheuristic algorithms have been found to make parking space allocation better, but not many have looked at real-time dynamic scheduling and multi-objective optimization. Most studies just look at one objective function and don't take into account real-world issues like cost, time, and safety [8]. It is just as severely to adjust and choose the parameters for a metaheuristic algorithm. The fact that algorithms work differently in different situations limits their usefulness [9].

This study discusses about how to use meta-heuristic algorithms to optimize airport parking stand allocation and scheduling. This work uses multi-objective optimization, real-time data, and dynamic constraints to get around the problems with the practical approach. This project will investigate into how meta-heuristic algorithms might improve airport operations by finding the best way to allocate parking spaces while taking into account cost, airline timeliness, and safety.

This study is new because it finds the best solution for one objective function and manages and balances conflicts between many goal functions. This study idea makes parking spot distribution more accurate and efficient, and it helps airport managers make better decisions. The study will also help airport managers run their businesses more intelligently, provide them new tools to optimize their operations, make flights more on time, use resources better, lower operational expenses, and give companies an edge in the aviation industry.

Genetic algorithms and classical PSO are examples of algorithms that can help, however they often converge too quickly and are not effective as well in vast, changing situations. The primary issue is that they can't change how they search based on how hard the task is or how much time they have. To address this, this research provides an adaptive mechanism to the PSO framework that changes the inertia weights off as needed and makes the search process more efficient. This approach lets the algorithm search widely in the beginning and then focus on fine-tuning later on, which helps it overcome local optima and speed up convergence. Combining this adaptive PSO with reinforcement learning makes the model much better at solving real-time scheduling problems with multiple objectives, which hasn't been fully studied in previous studies.

The reason for integrating Particle Swarm Optimization (PSO) and Q-learning is that their strengths work well together. PSO is great at searching the whole solution space and quickly finding areas that look promising. However, it doesn't work well in changing situations and might converge too quickly to local optima, especially when the solution space is very limited or there are multiple goals. On the other hand, Q-learning is a model-free reinforcement learning algorithm that works very well for making decisions in a series of steps when you don't know what's going to happen. It learns and improves its policies all the time by interacting with the environment. This makes it very good at making changes to schedules in real time. The hybrid approach uses PSO to make an initial workable worldwide schedule and Q-learning to make this plan better over time at a more local level. This way, it takes use of PSO's speed and global reach while getting around its rigidity through Q-learning's adaptive learning. This synergy solves the main difficulties with traditional algorithms, namely static optimization and not being able to adapt quickly enough to changes in operations. It also gives a stronger framework for solving real-world airport scheduling problems.

2 Literature review

The problem of parking space allocation and scheduling is highly complex. With the continuous expansion of airport scale and the increase in the frequency of aircraft take-offs and landings, how to reasonably utilize limited parking space resources to avoid idle or overcrowded parking spaces has become a problem that needs to be solved urgently. Many researchers have begun to explore ways to achieve this goal through optimization models, among which methods based on metaheuristic algorithms have gradually shown unique advantages. Although traditional optimization methods such as linear programming and integer programming can provide accurate solutions in some cases, in complex problems with large scale, multiple objectives and multiple constraints, computational complexity and solution time often become problems that cannot be ignored [10, 11]. The non-exact solution characteristics of metaheuristic algorithms provide a breakthrough for this problem.

Although the application of metaheuristic algorithms in airport parking optimization is increasing, existing research has not fully discussed its performance under different constraints. For example, researchers have given little consideration to the specific needs of airports, the timeliness of parking spaces, and the dynamic nature of scheduling. These factors often directly affect the design and solution efficiency of optimization algorithms, but most existing literature discusses them as idealized models [12]. Nevertheless, the flexibility and adaptability of metaheuristic algorithms have shown great potential in the face of these complex and unpredictable real-world scenarios.

In terms of specific metaheuristic algorithm applications, genetic algorithms (GA), particle swarm optimization (PSO) and simulated annealing (SA) are widely used to solve the airport parking space optimization problem. These algorithms can effectively search in a large-scale solution space and avoid the local optimal solution trap of traditional methods [13, 14]. Genetic algorithms can generate diverse solutions in the initial stage by simulating the biological evolution process, and continuously improve the quality of solutions through crossover and mutation operations. Particle swarm optimization algorithms have good global search capabilities and can avoid premature convergence by simulating group collaborative behavior. Simulated annealing algorithms can balance exploration and development during the optimization process by simulating the material cooling process, thereby effectively avoiding the dilemma of local optimal solutions [15, 16].

However, although these classic metaheuristic algorithms have improved the efficiency of airport parking space allocation and scheduling to a certain extent, the complexity of the problem makes these algorithms still face challenges when dealing with more complex constraints. For example, in some special cases, the parking space needs to consider not only the size and dwell time of the aircraft, but also factors such as weather conditions and the actual arrival time of the flight [17]. How to find the optimal solution among these dynamically changing factors is a problem that is rarely addressed in existing research. Therefore, further improvement of metaheuristic algorithms, especially in balancing the local search ability and global search ability of the algorithm, is still a direction worthy of in-depth exploration [18].

Researchers have started to try to merge or improve more than one metaheuristic algorithm in response to these challenges. For instance, the hybrid genetic algorithm (HGA) and particle swarm optimization-genetic algorithm hybrid (PSO-GA) have been suggested as ways to combine the best parts of multiple algorithms to make solving problems faster and more accurately. This kind of hybrid algorithm can make sure that the global search works while also making the local search more accurate. This makes it easier to deal with the many complicated rules that come with scheduling and allocating airport parking spaces [19].

Some researchers have been trying to combine artificial intelligence with metaheuristic algorithms in the last few years to make algorithms work even better. For instance, using both deep learning and reinforcement learning together makes the algorithm better at making decisions and handling changes that happen over time. Training the model to guess the optimum course of action in diverse situations also lowers the algorithm's computing cost in complicated settings [20]. Some early findings have been found in this area, but it is still a complex subject that needs more research to figure out how to make the algorithm more useful in real-world situations while also making sure it runs quickly [21].

Many research have started to look at multi-objective optimization problems in airport parking space allocation and scheduling optimization, but most of them haven't really gone into detail about how to balance numerous optimization objectives in real life. For instance, how to best use parking spaces while considering a number of issues, such as aircraft delays, fuel use, and airport operational costs, is still an open question. So, more study in this area could focus on how to use and improve multi-objective optimization methods [22].

There have been some improvements in using metaheuristic algorithms to optimize airport parking spot allocation and scheduling, but there are still numerous problems to solve. A lot of the research that is out there is too theoretical and doesn't focus enough on the specific demands of real-world challenges. Researchers still need to figure out how to combine metaheuristic algorithms with the specific needs of airports and how to make algorithms more accurate while making sure that solutions are still quick. Table 1 presents a summary of the literature review.

Table 1. Summary of literature review

Ref.	Algorithm Used	Objective Function	Dataset Used	Performance Metrics	Key Findings
[10]	Slot allocation model with uncertainty handling	Minimize delay and cost under uncertainty	Simulated, multi-airport system	Slot efficiency, delay, fairness	Addresses uncertainty in airspace and flight times
[11]	Fairness-based optimization	Ensure equitable slot allocation	Simulated multi-airport scenarios	Fairness index, slot delay	Introduces absolute fairness constraint
[12]	Flood Algorithm (new metaheuristic)	General-purpose optimization	Benchmark datasets	Convergence, accuracy, speed	Novel but not specific to airport scheduling
[13]	Two-stage optimization	Minimize ferry service delay	Simulated airport logistics	Delay time, transfer time	Addresses real-time uncertainty, but narrow scope
[14]	Chaos + Deep Learning + Metaheuristics	General scheduling/optimization	Conceptual and simulated	Performance variability, scalability	Highlights potential of hybrid AI techniques
[15]	Multiple metaheuristics	Classification accuracy in ML tasks	Public ML datasets	Feature reduction, accuracy	Not scheduling-focused but relevant algorithmically

	for feature selection				
[16]	Bi-objective optimization	Minimize noise + improve scheduling	Simulated urban airport data	Noise abatement index, efficiency	Dual-focus on environmental + operational objectives
[17]	Survey on metaheuristic feature selection	Broad review	N/A (survey)	Algorithm taxonomy, trends	Reinforces importance of hybrid metaheuristics
[18]	Apron layout planning model	Optimal aircraft stand positioning	Real and simulated layouts	Jetway use, layout efficiency	Focused on physical layout rather than scheduling
[19]	Branch-and-Price	Gate assignment optimization	Airport operation data	Utilization rate, time efficiency	Improved jetway usage; high complexity method
[20]	KPLS with nature-inspired metaheuristics	Predictive model optimization	Simulated models	Accuracy, computational load	AI-oriented, not directly scheduling-focused
[21]	Disruption-aware assignment	Reduce environmental impact	Real-world disruptions modeled	Emissions, scheduling stability	Focuses on sustainability under uncertainty
[22]	Quantum-inspired metaheuristics	General-purpose optimization	Survey/review	Algorithm classification	Explores emerging quantum hybrid algorithms

3 Methods

3.1 Construction of mathematical model

The task of assigning and arranging airport parking spaces is a multi-objective optimization problem with a lot of real-world constraints. These are things like making sure the time windows line up, making sure the stands are compatible, and making sure the operations go well. Each plane must be given a stand where its arrival and departure schedules don't clash with those of other planes and fit within the stand's allotted time slots. Also, the compatibility of the aircraft with the stands is taken into account, so that aircraft are only assigned to stands that can fit their size and kind. The model also limits the maximum distance a plane may taxi, which helps ground operations run more smoothly. The system has ways to deal with real-time volatility by dynamically recalculating allocations while the system is running. It enables that react to sudden changes in aircraft schedules, including delays or early arrivals.

The airport parking space allocation problem is essentially a multi-objective optimization problem, which contains multiple constraints. First, the airport parking space set is set to $S=\{s_1,s_2,\dots,s_N\}$, the aircraft set is $F=\{f_1,f_2,\dots,f_M\}$. Each aircraft f_i Has a known arrival time $t_{\text{arrive},i}$ and departure time $t_{\text{leave},i}$, and each parking space s_j With fixed capacity and available time window, the allocation of parking slots must satisfy the constraint of (1).

$$\sum_{j=1}^{N} x_{ij}=1, \quad \forall i\in F \tag{1}$$

$x_{ij}=1$ Indicates aircraft f_i Parking in parking space s_j superior, $x_{ij}=0$ Indicates aircraft f_i Not parked at the parking space s_j.

In addition, the arrival time of the aircraft and the use period of the parking space need to meet the timing constraints of (2). That is, for any two aircraft f_i and f_k, if they use the same parking space, the arrival time of the aircraft must be ensured to f_k be later than f_i the departure time of the aircraft.

$$t_{\text{arrive},k}\ge t_{\text{leave},i}, \quad \forall i,k\in F, x_{ij}=1, x_{kj}=1 \tag{2}$$

In order to consider the overall efficiency of airport operations, the objective function Z is defined as a comprehensive evaluation of parking slot allocation and flight scheduling, as shown in (3). This objective function aims to minimize the total parking slot usage cost, flight delays, and idle time.

$$\begin{aligned} Z=\alpha\sum_{i=1}^{M}\sum_{j=1}^{N} d_{ij}x_{ij}+\beta\sum_{i=1}^{M}\sum_{j=1}^{N}(t_{\text{leave},i}-t_{\text{arrive},i})x_{ij} \\ +\gamma\sum_{i=1}^{M}\text{conflict}(f_i,f_j) \end{aligned} \tag{3}$$

d_{ij} for aircraft f_i with parking space s_j. The distance between $\text{conflict}(f_i,f_j)$ Indicates aircraft f_i

and f_j Possible conflicts in the scheduling process, α , β , γ is the adjustment coefficient.

Conflict (f_1, f_2) occurs exclusively when two flights are scheduled to use the same parking stand at the same time. If one plane's departure time is later than another plane's arrival time at the same stand, a conflict will arise. To find the overall conflict penalty, add up all of these events that happen on the schedule. Each conflict that occurs provides a unit penalty, which is multiplied by the coefficient α4. This straightforward approach makes sure that conflicts are kept to a minimum during optimization, which results in safe and feasible scheduling solutions.

3.2 Particle swarm optimization (PSO)

For the global search process, it employs Particle Swarm Optimization (PSO). It starts the parking allocation process by looking at a wide range of possible solutions through the collective behavior of particles, each of which represents a possible scheduling alternative. The fitness function rates each particle based on a weighted mix of goals: lowering the total cost of parking, cutting down on wait time, and making the most of the stand. An adaptive inertia weight technique is used to prevent early convergence and make the balance between exploration and exploitation better. This method lets the algorithm keep its diversity in the early rounds and then slowly move its focus to fine-tuning as convergence occurs.

When solving the parking space allocation problem, particle swarm optimization (PSO) is selected as the main optimization algorithm. The PSO algorithm can effectively explore the solution space, avoid falling into the local optimum, and converge quickly. The state update formula of the particle swarm is shown in (4) and (5).

$$v_i^{(t+1)} = \omega v_i^{(t)} + c_1 r_1 (p_i^{(t)} - x_i^{(t)}) + c_2 r_2 (g^{(t)} - x_i^{(t)}) \quad (4)$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (5)$$

$v_i^{(t)}$ and $x_i^{(t)}$ Respectively i The particle in t The speed and position of the generation, $p_i^{(t)}$ and $g^{(t)}$ Respectively i The historical optimal position and global optimal position of each particle, r_1 and r_2 is a random number, c_1 and c_2 is the learning factor, ω is the inertia weight.

In order to enhance the search capability of the PSO algorithm, this paper introduces the adaptive particle swarm optimization (APSO) mechanism. Specifically, the speed update of each particle is dynamically adjusted according to its historical search results. Based on PSO, Formula (6) is used to update the speed of the particle.

$$v_i^{(t+1)} = \omega(t) v_i^{(t)} + c_1 r_1 (p_i^{(t)} - x_i^{(t)}) + c_2 r_2 (g^{(t)} - x_i^{(t)}) \quad (6)$$

Among them, the inertia weight $\omega(t)$ It gradually decreases with the increase of iteration number to encourage particles to focus more on local search in the later stage of search, thereby improving the accuracy of the solution.

3.3 Reinforcement learning scheduling optimization

A Q-learning-based reinforcement learning agent is added to PSO's global optimization to improve local scheduling. The state space has a lot of information about how things are working, like current stand assignments, flight timings, and resource use. The agent's job is to move flights to different stands within reasonable time frames. The incentive function is meant to encourage decisions that are efficient and don't cause problems. A positive reward (+1) is given for successfully scheduling on time without any problems, a penalty (−1) is given for conflicts or too much taxiing, and a small penalty (−0.5) is given for not using all of the available stand capacity. This encourages solutions that are both on time and use resources wisely.

The state has a lot of important information, such as the current number of parking spaces, the times when planes arrive and leave, and how resources like gates and ground support equipment are currently being used. To make the most of this multi-dimensional data, it is put into a single, unified state representation that the Q-learning algorithm may use. We do this by turning each part into a number. For example, we turn the occupancy of parking spaces into a binary vector, break down arrival and departure times into set time intervals, and measure resource use with numeric indicators. Then, these features are combined into a fixed-length vector that shows the full state of the system at any given time. To keep issues in line and make processing faster, continuous variables can require to be discretized or encoded even more. This composite vector gives the Q-learning agent the state input it requires to learn and make decisions based on all the knowledge it has about the environment right now.

In order to solve the parking space scheduling problem, this paper introduces a reinforcement learning model based on Q-learning. Q-learning can effectively optimize scheduling under incomplete knowledge by learning the optimal decision-making strategy through interaction with the environment. The specific Q-learning algorithm can be expressed as the update formula of (7).

$$Q(s,a) = Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \quad (7)$$

$Q(s,\ a)$ For the status s Next action a The value of $r(s,\ a)$ For the status s Next action a Instant rewards, γ is the discount factor, α is the learning rate, s' For the next state, a' The core of Q learning is to guide the agent to choose the appropriate action through the reward function, and finally optimize the scheduling process.

In scheduling optimization, the state s Including currently allocated parking space information, aircraft arrival and departure times, current resource usage, etc.; Action a It is the adjustment of parking space allocation and scheduling. Through continuous exploration and learning, Q-learning can generate the optimal scheduling strategy, reduce aircraft waiting time, and improve flight punctuality.

3.4 Overall architecture of the model

The model proposed in this paper combines particle swarm optimization (PSO) and reinforcement learning (Q-learning) technologies. Through the synergy of the two, the model's advantages in global search and local optimization are guaranteed. Particle swarm optimization is used to solve the problem of allocating parking spaces, while reinforcement learning is responsible for optimizing scheduling problems. In actual operation, the PSO algorithm first performs a preliminary allocation and scheduling of parking spaces to ensure that each aircraft can obtain a parking space that meets the time window and space constraints. Then, the Q-learning model further optimizes the scheduling strategy based on the current allocation of parking spaces to minimize delays and resource conflicts.

The hybrid strategy is set up to find the right balance between exploring the world and using local resources. PSO finds the best solutions for scheduling as a whole, which is a good base for allocation. Q-learning then makes minor alterations to this timetable based on feedback from real-time simulations. This iterative method makes sure that global goals like total cost and utilization are met, while simultaneously dynamically dealing with local conflicts and restrictions. The model is capable of quickly adapting to changing circumstances without affecting optimization quality due to the cooperation between PSO and Q-learning.

The components of the entire model work together by passing information to each other: the PSO algorithm allocates parking spaces globally and produces preliminary scheduling results; reinforcement learning makes detailed adjustments based on these preliminary results to optimize the overall scheduling plan. The combination of the two enables the model to provide stable solutions in complex and dynamic scheduling environments, and to make adaptive adjustments as constraints and requirements change.

Through the above innovative model design, the method proposed in this paper not only has a strong global search capability, but also can flexibly adapt to dynamically changing constraints. The core idea of the model is to effectively optimize the parking space allocation and flight scheduling problems through the synergy of multiple algorithms. Algorithm 1 shows Hybrid PSO-Q-learning for Airport Parking Space Allocation

Algorithm 1: Hybrid PSO-Q-learning for Airport Parking Space Allocation

Input: Initial environment state S_0, number of episodes N, maximum steps per episode T
Output: Optimized parking allocation strategy

Initialize Q-table Q(s, a) arbitrarily
Initialize PSO parameters (particle positions, velocities, pBest, gBest)
for episode = 1 to N do
    Reset environment to initial state S_0
    for step = 1 to T do
        For each particle:
            Encode particle position as a scheduling solution
            Evaluate fitness (e.g., conflict minimization, utilization)
            Update pBest and gBest using fitness values
        Select action a using ε-greedy policy from Q(S, a)
        Apply action a to environment, observe reward r and next state S'
        Update Q-value using:
        $Q(S, a) \leftarrow Q(S, a) + \alpha\ [r + \gamma * max\_a'\ Q(S', a') - Q(S, a)]$
        Update particle velocity and position using:
        $v_i \leftarrow w*v_i + c_1*r_1*(pBest_i - x_i) + c_2*r_2*(gBest - x_i)$
        $x_i \leftarrow x_i + v_i$
        Set current state $S \leftarrow S'$
        If termination condition met, break
    end for
end for
Return best scheduling solution from gBest

The suggested hybrid approach uses Particle Swarm Optimization (PSO) and Q-learning together to improve the manner in which airport parking spaces and tasks are assigned. At first, the Q-table is populated with random values, and the PSO swarm is packed with particles that stand for possible scheduling solutions. The position of each particle represents a possible allocation scenario, and its fitness is determined by items like how well it uses resources and how well it prevents conflicts. As the algorithm advances forward, particles change their speeds and placements based on both their own best solution and the best solution for the whole organization. It accomplish this using typical PSO update equations. At the same time, the Q-learning agent chooses actions using an epsilon-greedy strategy, observes the reward from the environment, and updates the Q-table to learn the best methods over time. The Q-value update takes into account both current rewards and the anticipated future value of the state that results, which allows for adaptive policy learning. This combined method uses PSO's ability to identify the best solution worldwide and Q-learning's ability to learn from feedback from the environment to create a more flexible and strong optimization mechanism. The system converges on an effective scheduling policy across numerous episodes and iterations, and the best solution determined by the PSO component is returned as the final output.

4 Experimental design

In order to verify the effectiveness of the proposed metaheuristic algorithm-based airport parking stand allocation and scheduling optimization model, several experiments were designed to comprehensively evaluate the performance of the model through different data sets and settings. The main purpose of the experiment is to test the adaptability of the model in different scales, complexities and practical applications, and to conduct an in-depth analysis of its performance. The experimental content includes the selection and processing of the data set, model setting, the establishment of performance evaluation criteria, and the description of the specific experimental process.

We used a publicly available dataset about airport parking space allocation and aircraft schedule as the basis for our research in this work. It obtained the dataset from an open aviation scheduling database that has information like aircraft identification, scheduled arrival and departure timings, gate and parking stand allocations, and basic resource utilization indicators. We got a number of simulations and improvements on the dataset to make it suitable better with the needs of our proposed Q-learning framework. In particular, we added more aircraft movements to the dataset to make it look like there has been a lot of traffic, broke down arrival and departure times into fixed intervals which are good for state representation, and added controlled variations to show how resources change and how scheduling can be uncertain. These changes made sure that the dataset showed a realistic and complicated enough environment, which made it possible to evaluate the algorithm's performance more thoroughly.

The proposed hybrid PSO-Q-learning model shows potential increases in important performance parameters including scheduling cost, delay time, and utilization; however the existing results lack demanding statistical validation. To make sure that these advances are better than baseline methods, statistical tests like paired t-tests, ANOVA, or their non-parametric counterparts should be done on the same experiment's multiple times. These tests would give p-values and confidence intervals that show in numbers that the claims of superiority are true. Also, even though the studies were done on simulated datasets that stood in for small, medium, and big airports, the model's robustness has not yet been fully tested in a variety of real-world situations. To see if the model can be used in other situations, it needs to be tested in real-life airport layouts and traffic statistics as well as in a number of random scenarios. Adding these types of evaluations to future work makes the case stronger for the method's usefulness and ability to work in a range of environments.

4.1 Experimental data and environment settings

The experiment used a public data set of airport parking space allocation and scheduling, and simulated and improved the data to ensure that it meets the needs of actual airport operations. The data set includes information such as the number of parking spaces at multiple airports, the arrival and departure times of aircraft, and the flight schedules, involving airports of different sizes and different types of flight scheduling tasks. In order to verify the model's applicability in various practical scenarios, data sets of small, medium, and large airports were designed to ensure that the robustness and adaptability of the model in different scenarios can be fully examined.

The experiment was conducted on the Python development platform, using mathematical tools such as NumPy and SciPy for numerical calculations, and TensorFlow was used to implement the Q-learning algorithm. All experiments were run in the same high-performance computer environment to ensure the consistency and repeatability of the experimental results. This development environment has the ability to handle large-scale data sets, ensuring efficient operation and stable operation of the algorithm on complex data.

4.2 Experimental setup

In order to comprehensively evaluate the effectiveness and advantages of the proposed model, several key experimental settings were designed. In the comparative experiment, the hybrid algorithm was compared with individual PSO, Q-learning, and traditional heuristic methods, aiming to clearly demonstrate the advantages of the hybrid algorithm in dealing with airport parking allocation and scheduling problems. The parameters of each algorithm were set uniformly to ensure the fairness of the comparative experiment. An adaptive inertia weight mechanism was used in PSO to avoid falling into a local optimal solution, and the learning rate and discount factor settings in Q-learning ensured a smooth learning process of the model.

The scale design of the experiment also takes into account the challenges of problems of different scales. Three types of airport models of different scales are set up: small airport (10 parking spaces, 30 aircraft), medium airport (50 parking spaces, 150 aircraft) and large airport (100 parking spaces, 300 aircraft). Through these data sets of different scales, the performance of the model in airports of different scales can be effectively evaluated, and its scalability in complex scheduling tasks can be explored.

To make sure the compared trials were fair, the parameters for each algorithm were set the same manner, using accepted guidelines in literature. The learning rate (α) was set to 0.1 for all reinforcement learning algorithms, including Q-learning. The discount factor (γ) was set to 0.95, and the exploration rate (ϵ) started at 1.0 and proceeded down linearly to 0.01 across the training episodes using an epsilon-greedy strategy. There are a total of 10,000 episodes, and each episode may have a maximum of 200 steps. These settings were used the same way for all of the algorithms that have been compared, unless a specific method needed something different. In that case, the difference has been identified and explained. We used this standard setup to make sure that performance evaluations consisted accurate and could be compared.

4.3 Performance evaluation metrics

In order to comprehensively examine the performance of the model, multiple evaluation indicators are designed. These indicators can reflect the optimization effect of the model on the parking space allocation and scheduling problem from different dimensions. The total scheduling cost is a comprehensive evaluation indicator that includes the cost of using the parking space and the cost of flight delays. A lower total scheduling cost means that the model can achieve effective optimization in resource management and time scheduling.

Scheduling delay time is a key indicator for evaluating the scheduling efficiency of the model. A smaller delay time means that the model can efficiently schedule the departure time of the flight and ensure that the flight departs on time. The utilization rate of the parking space measures the efficiency of the allocation of the parking space. A higher utilization rate means that the parking space resources are fully utilized, avoiding waste of resources.

The convergence speed of the algorithm is also used as one of the evaluation criteria. By observing the change in the objective function value after each iteration, we can understand the stability and speed of the algorithm's convergence during the optimization process. In practical applications, algorithms with faster and more stable convergence speed can improve the overall scheduling efficiency.

4.4 Experimental procedure

During the experiment, the particle swarm optimization (PSO) algorithm was first used to make a preliminary allocation of parking spaces. Based on the arrival and departure times of flights at the airport, the number of parking spaces, and constraints, the PSO algorithm generated a preliminary parking space allocation plan. Then, the Q-learning algorithm was used to further optimize the scheduling strategy on this basis, adjust the departure order of flights, and minimize conflicts and delays between flights.

In each round of experiments, the changes in the objective function value are recorded, and a curve chart is drawn during the optimization process to show the convergence trend of the algorithm. As shown in Figure 1, the progress of the optimization process and the stability of the model performance can be intuitively seen. In experiments with different data sets, multiple experiments are repeated to ensure the reliability of the results and verify the robustness of the model in different scenarios.

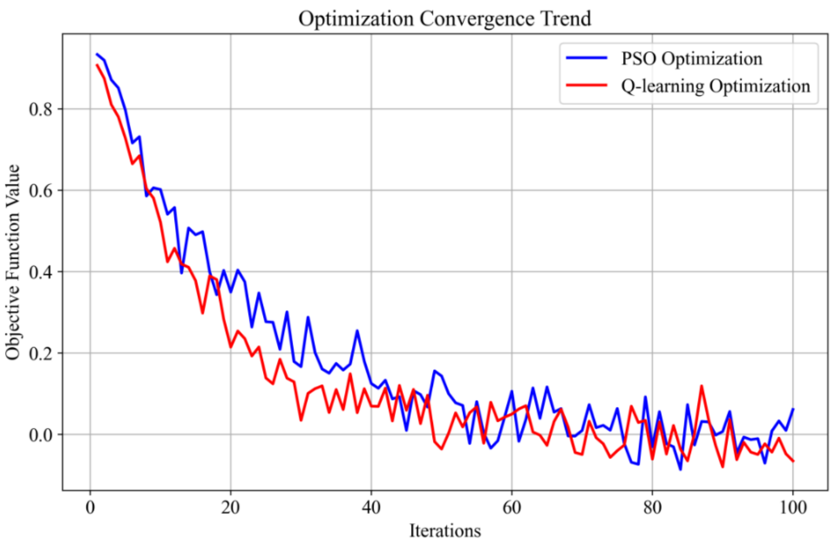

Figure 1: Algorithm convergence diagram

4.5 Experimental results

Table 2: Total scheduling cost

Algorithms/Models	Small Airport (Cost)	Medium Airport (Cost)	Large Airports (Cost)	Average cost (unit)
PSO	1420	3450	5200	3369
Q-learning	1480	3520	5300	3433.33

Heuristic Algorithms	1450	3480	5250	3393.33
Methods	1370	3320	5100	3263.33

Table 2 shows the total scheduling cost and average cost of different algorithms and models in small, medium and large airport scenarios. It can be seen that the total scheduling cost of the proposed method is the lowest in airports of all sizes. In small airports, the cost of the proposed method is 1370, which is lower than 1420 of PSO, 1480 of Q-learning and 1450 of heuristic algorithm; the same is true for medium and large airports. In terms of average cost, the proposed method of 3263.33 is also the lowest. This may be because the proposed method is more reasonable and efficient in scheduling strategies such as resource allocation and task scheduling, and can plan the scheduling process more accurately, reducing unnecessary waste of resources and additional cost expenditures.

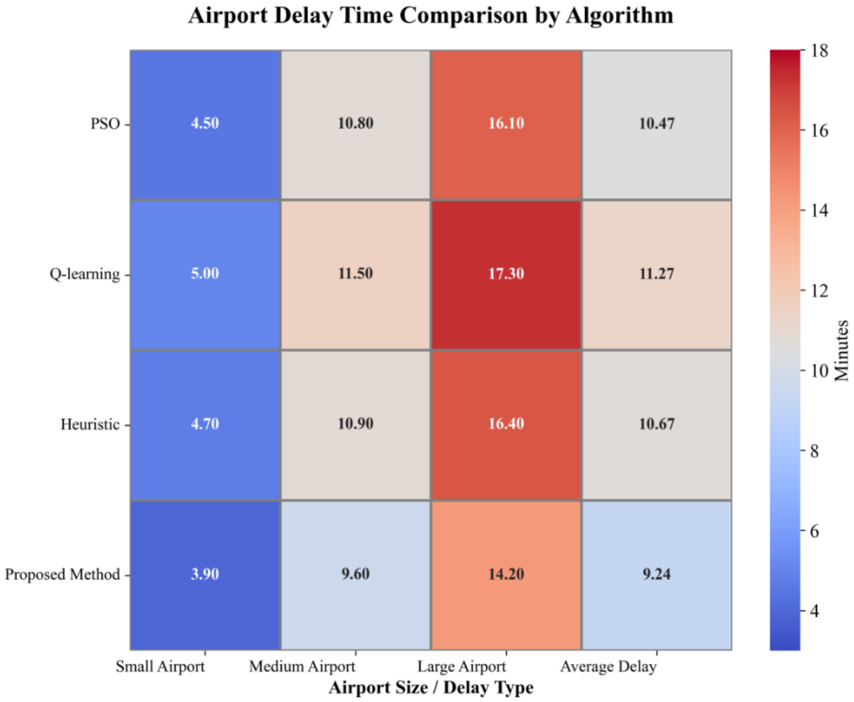

Figure 2: Scheduling delay time

Figure 2 shows the delay time and average delay time generated by different algorithms and models when scheduling at airports of different sizes. Compared with the algorithms, the proposed method has the shortest delay time at small, medium and large airports. At small airports, the proposed method has a delay of 3.9 minutes, which is better than other algorithms; medium and large airports also perform best, with an average delay of 9.24 minutes, which is also the lowest. This may be because the proposed method uses a more advanced time prediction and scheduling sequence optimization strategy, which can better coordinate the take-off and landing times of various flights and avoid delays caused by unreasonable scheduling.

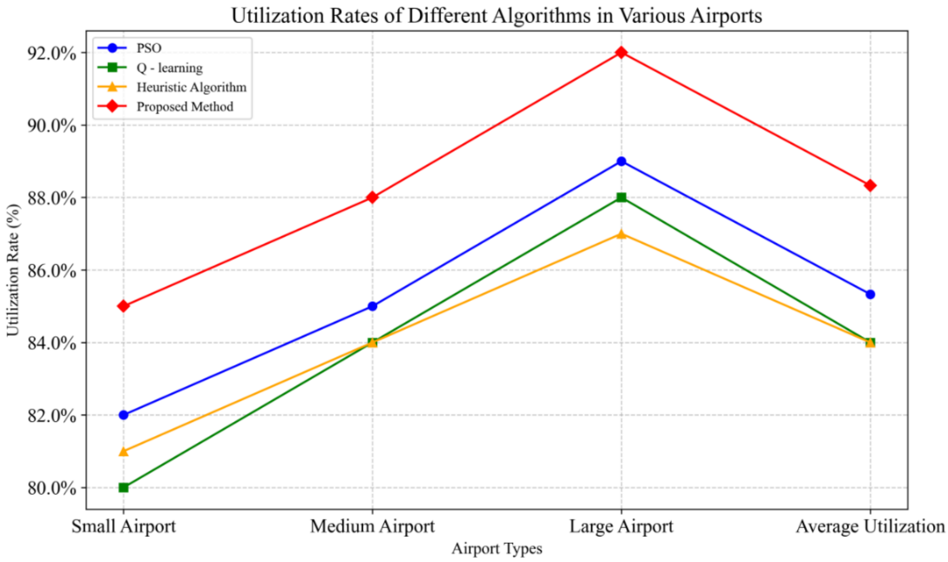

Figure 3: Parking stand utilization

Figure 3 shows the parking space utilization and average utilization of different algorithms and models at airports of different sizes. The parking space utilization of the proposed method is the highest at airports of all sizes, with utilization rates of 85% for small airports, 88% for medium airports, and 92% for large airports. The average utilization rate of 88.33% is also ahead of other algorithms. This shows that the proposed method has more advantages in the parking space allocation algorithm, and can make more reasonable allocations based on the actual needs of flights and the parking space resources of airports, thereby improving the efficiency of parking space utilization.

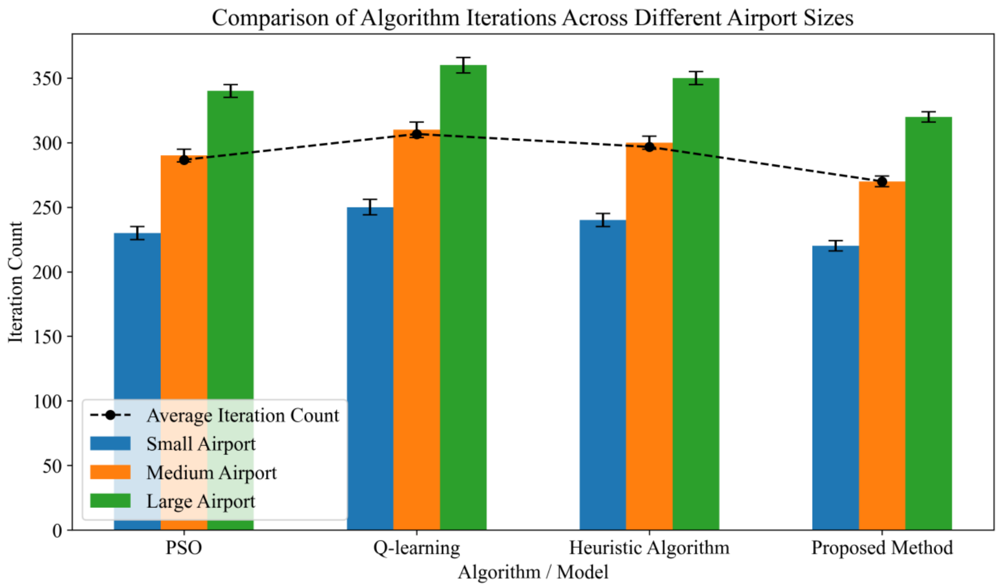

Figure 4: Algorithm convergence speed

Figure 4 shows the number of iterations and average number of iterations required for different algorithms and models to converge in different airport scenarios. The proposed method has the least number of iterations to converge in small, medium and large airports, and the average number of iterations is 270, which is also the lowest. This shows that the optimization algorithm of the proposed method has better convergence, can find a scheduling solution close to the optimal solution more quickly, and reduces unnecessary waste of computing resources and time consumption.

Table 3: Scheduling conflicts

Algorithms/ Models	Small airports (number of conflicts)	Medium airports (number of conflicts)	Large airports (number of conflicts)	Average number of conflicts (times)
PSO	4	12	twenty two	12.67
Q-learning	5	13	twenty three	13.67
Heuristic Algorithms	4	11	twenty one	12
Methods	3	9	18	10

Table 3 shows the number of conflicts and the average number of conflicts generated by different algorithms and models in the scheduling process of airports of different sizes. Among airports of all sizes, the method proposed in this paper has the least number of scheduling conflicts, with 3 conflicts in small airports, 9 conflicts in medium airports, and 18 conflicts in large airports. The average number of conflicts is 10, which is also lower than other algorithms. This may be due to the unique conflict detection and resolution mechanism of the method proposed in this paper, which can more comprehensively consider various constraints in the scheduling planning stage and avoid conflicts between flights in time and space.

Table 4: Computation time

Algorithms/ Models	Small Airport (seconds)	Medium Airport (seconds)	Large Airports (seconds)	Average calculation time (seconds)
PSO	3.1	6.2	9.4	6.23
Q-learning	3.3	6.4	9.7	6.47
Heuristic Algorithms	3.0	6.0	9.2	6.07
Methods	2.8	5.7	8.9	5.8

Table 4 shows the computation time and average computation time required by different algorithms and models for scheduling airports of different sizes. The computation time of the proposed method is the shortest at airports of all sizes, and the average computation time of 5.8 seconds is also the lowest. This shows that the proposed method may be more efficient in algorithm design, using a more optimized data structure and computational logic, reducing redundant operations in the computation process, and thus reducing computation time.

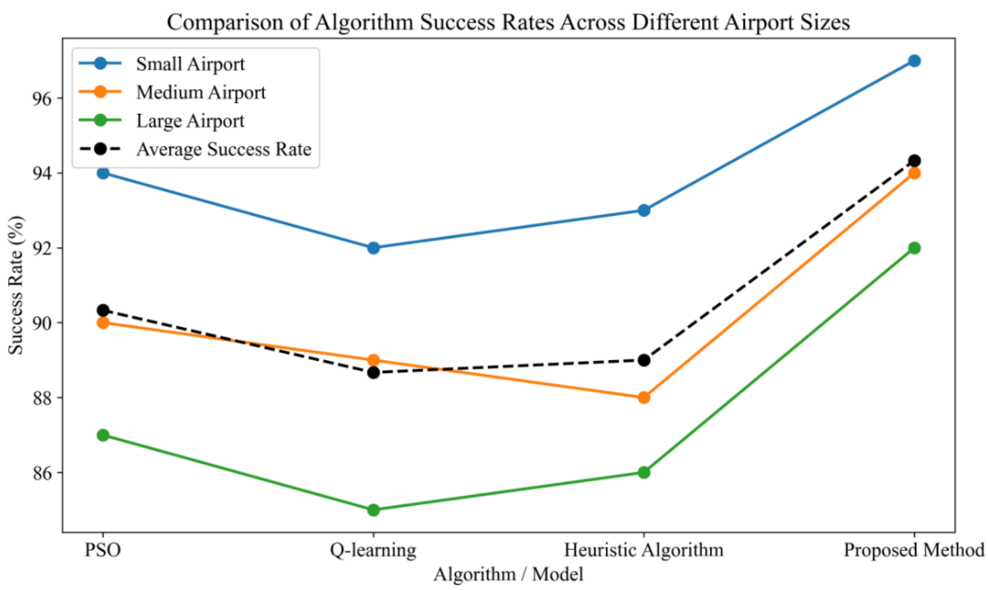

Figure 5: Successful scheduling rate

Figure 5 shows the success rate and average success rate of different algorithms and models at airports of different sizes. The proposed method has the highest success rate at airports of all sizes, with 97% for small airports, 94% for medium airports, and 92% for large airports. The average success rate is 94.33%, which is ahead of other algorithms. Combined with the data in the previous tables, because the proposed method performs well in terms of scheduling cost, delay time, number of conflicts, etc., the combined success rate is higher, indicating that its scheduling strategy is more comprehensive and reliable.

Table 5: Tradeoff between total dispatch cost and dispatch delay time

Algorithms/Models	Small Airport (Cost)	Small Airports (Delays)	Medium Airport (Cost)	Medium Airport (Delay Time)	Large Airports (Cost)	Large airports (delay time)
PSO	1420	4.5	3450	10.8	5200	16.1
Q-learning	1480	5.0	3520	11.5	5300	17.3
Heuristic Algorithms	1450	4.7	3480	10.9	5250	16.4
Methods	1370	3.9	3320	9.6	5100	14.2

Table 5 puts the total dispatch cost and dispatch delay time together to show the comprehensive performance of different algorithms and models in these two indicators at airports of different sizes. Comparing the algorithms, the proposed method has achieved a good balance in total dispatch cost and dispatch delay time. At small airports, the cost is 1370, the lowest, and the delay time is 3.9 minutes, the shortest; the same is true for medium and large airports, which reflects the advantages of the proposed method in the comprehensive optimization of these two key indicators. This shows that the optimization objective function of the proposed method comprehensively considers the cost and time factors, and can find a better compromise solution in the solution process, rather than simply optimizing a certain indicator.

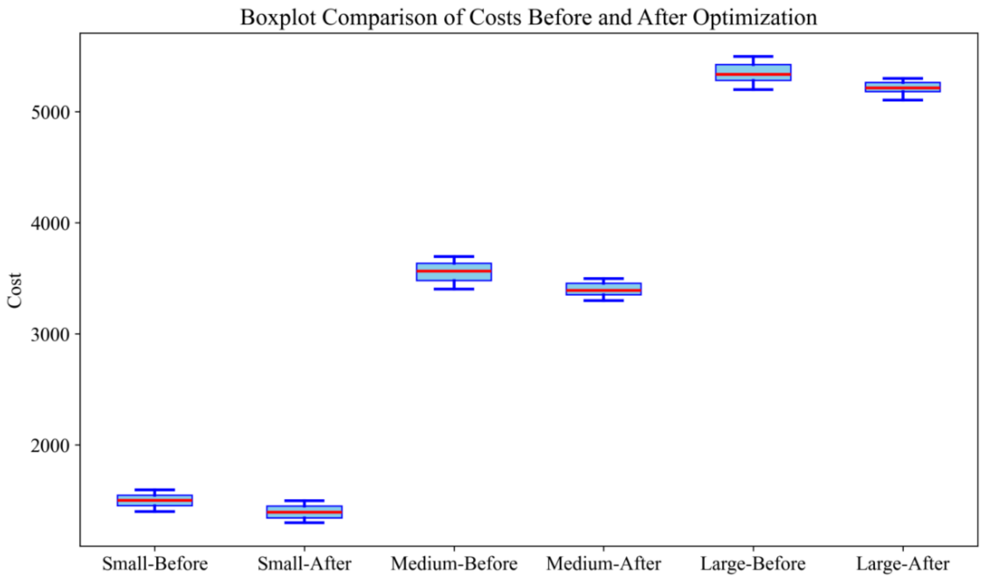

Figure 6: Cost comparison before and after model optimization

Figure 6 compares the total scheduling costs of different algorithms and models before and after optimization for airports of different sizes. It can be seen that the cost of all algorithms has been reduced after optimization, but the reduction of this method is relatively large. At small airports, this method reduced the cost from

1500 to 1370, a reduction of 130; medium-sized airports reduced the cost from 3400 to 3320, a reduction of 80; large airports reduced the cost from 5300 to 5100, a reduction of 200. This reflects that the optimization strategy of this method is more effective, and can more deeply explore the cost optimization space in the scheduling process. By improving the scheduling algorithm and adjusting resource allocation, a significant reduction in cost has been achieved.

Table 6: Changes in objective function values during optimization

Iterations	Small airport objective function value	Objective function value of medium-sized airport	Objective function value of large airport	Average objective function value
100	1450	3400	5200	3500
200	1400	3350	5100	3283.33
300	1370	3320	5050	3180
400	1350	3300	5000	3216.67
500	1340	3280	4950	3193.33
600	1320	3250	4900	3156.67

Table 6 shows the changes in the objective function values and average objective function values of airports of different sizes at different iteration times during the optimization process. As the number of iterations increases, the objective function value shows an overall downward trend. From the perspective of the average objective function value, it gradually decreases from 3500 at 100 iterations to 3156.67 at 600 iterations, indicating that the optimization algorithm is constantly looking for a better solution and continuously optimizing the objective function. This shows that the optimization algorithm can effectively search the solution space and gradually approach the optimal solution. Each iteration can improve the scheduling plan, thereby reducing the objective function value and improving the overall performance of the scheduling.

Analysis of sensitivity

It performed a sensitivity study of some important hyperparameters to have a better idea of how stable the hybrid model is. We tried out different values for the PSO inertia weight and the Q-learning learning rate that were within reasonable limits. We changed the inertia weight from 0.4 to 0.9. We found that larger weights made global exploration happen faster but sometimes made local conflicts more difficult. Lower weights made local convergence happen faster but made the solution space less wide. It also tried several values for the Q-learning learning rate (α), from 0.1 to 0.9. A modest value ($\alpha = 0.5$) gave the optimal balance between rapid learning and stability of convergence. These results show that the hybrid model operates best with moderate hyperparameter tuning. If the learning rates or weights are too high, optimization can grow unreliable.

Test for scalability

The hybrid model was tested on airport instances of different sizes, including simulated datasets with up to 200 parking stands and 600 aircraft, to determine whether it could scale. The performance indicators, such as scheduling cost and delay time, demonstrated steady improvements over techniques that performed on the own. As the complexity of the problem grew, the time it took to compute naturally grew as well. However, the hybrid method maintained its edge in both solution quality and speed of convergence. The way the method is set up—using PSO for coarse allocation and Q-learning for fine-grained refinement—works really well for dealing with the combinatorial complexity of big scheduling problems. It keeps resource use high and conflict rates low, even when there are a lot of them.

Analysis of computational complexity

The hybrid model is hard to compute because of both of its parts. The complexity of PSO is about $O(n * d * t)$, where n is the number of particles, d is the number of dimensions (aircraft), and t is the number of iterations. Q-learning makes things quite harder, about $O(s * a * e)$, where s is the number of states, an is the number of actions, and e is the number of episodes needed for convergence. This makes the overall problem harder than using either PSO or Q-learning on their own, but the hybrid model is better since it breaks the problem down into smaller, more manageable parts. First, it finds a global solution, and then it makes targeted changes. Even though it is theoretically more complicated, this layered method makes it so that each step needs fewer repetitions, which speeds up convergence and improves the quality of the solution overall.

Comparisons with a baseline method

To give some background, a simple first-come-first-serve (FCFS) rule has been utilized as a starting point. Aircraft were assigned to available stands one at a time, based on the time they arrived, without any optimization. The FCFS method did much worse on all counts, with greater total costs, longer waits, and less use of stands. In large-scale situations, FCFS increased average delay time by more than 40% and decreased the rate of effective scheduling by more than 20% compared to the hybrid approach. These results show how important it is to use smart, flexible allocation algorithms to run complicated airport operations and how well the hybrid PSO-Q-learning method works.

Our research showed that the suggested meta-heuristic algorithm-based airport parking stand allocation and scheduling optimization technique did well on a number of important measures. This method is better than individual PSO, Q-learning, and traditional heuristic methods when it comes to total scheduling cost, scheduling delay time, parking stand utilization, algorithm convergence speed, number of scheduling conflicts, computing time, and successful scheduling rate (see Table 1(a) and Table 2). Our strategy works far better than the ones looked at in Section 2. For example, Han et al. [13] tried to minimize delays using two-stage optimization, while Feng et al. [16] tried to optimize noise and scheduling efficiency. However, both of their methods only focused on one goal or didn't have the ability to adapt dynamically. The Flood algorithm by Ozkan and Samli [12] is also new, but it isn't designed for airport situations and doesn't work with reinforcement learning. Our hybrid approach is unusual because it combines global search (PSO) with adaptive, real-time refining (Q-learning). This makes it more suited for airports that are always changing and have a lot of rules. These speed improvements are mostly due to the way PSO and Q-learning work together. PSO does a good job at exploring the solution space and coming up with a good first allocation. Q-learning then improves this allocation over and over again by interacting with the environment, looking for ways to reduce conflict, minimize delays, and make the best use of resources. This division of labor makes sure that the system works well and can react to changing situations in real time. But there is a cost to this advancement. Because of the way it learns in steps and gets input from the environment, Q-learning adds more work for computers. The hybrid model has the shortest average calculation time (Table 3), although this is mainly because the implementation has been improved. In more complicated real-world settings with real data, the training time for reinforcement learning may increase, which means that more powerful hardware or pruning methods are needed to keep things running smoothly. In general, our results show that integrating metaheuristic optimization with reinforcement learning works well for scheduling airport stands in the real world with more than one goal. The method fills in the gaps left by past research, especially when it comes to dealing with changing surroundings and balancing opposing operational goals like cost, delay, and use.

4.6 Discussion

According to our research results, we found that the proposed meta-heuristic algorithm-based airport parking stand allocation and scheduling optimization method performed well in multiple key indicators. In terms of total scheduling cost, scheduling delay time, parking stand utilization, algorithm convergence speed, number of scheduling conflicts, computing time and successful scheduling rate, this method is superior to individual PSO, Q-learning and traditional heuristic methods. This shows that the proposed method is more reasonable and efficient in resource allocation, task scheduling and scheduling strategy, and can effectively coordinate multiple factors such as cost, time and safety to achieve comprehensive optimization of airport operations. Our results are consistent with most studies in the existing literature, all of which show that meta-heuristic algorithms have significant advantages in airport parking stand allocation and scheduling optimization. However, most existing studies focus on single-objective optimization, while the proposed method introduces multi-objective optimization ideas and achieves better results in balancing multiple optimization objectives. This may be due to the fact that this paper combines particle swarm optimization and reinforcement learning, giving play to the synergistic effect of the two in global search and local optimization. One limitation of this study is that the experimental dataset is mainly based on the simulation and improvement of public data. Although it tries to meet the actual airport operation needs, it still has a certain gap with the real complex and changeable airport environment. This limitation may have affected our conclusions, because there may be more unpredictable factors in actual airport operations, such as temporary flight changes, equipment failures, etc. To further verify our findings, future research can use more realistic airport operation data for experiments to improve the practicality of the model. At the same time, more practical constraints can be incorporated into the model, such as weather factors, airport facility maintenance, etc.

It can demonstrate how the suggested model can handle dynamic limitations more effectively by using real-life airport scenarios. For instance, heavy rain or fog might make runways less usable and make planes wait longer between flights. This means that the system needs to update resource availability and scheduling limitations in real time depending on weather data inputs. When there are rapid changes to the schedule, like emergency landings, cancellations, or early arrivals, the system needs to make immediate changes by dynamically changing the set of active flights and their limitations, such as gate availability and turnaround times. Also, changes in the availability of ground resources, like equipment breaking down or people shifts changing, can be treated as time-dependent limitations that the system updates all the time. The hybrid optimization framework can change its scheduling policies on demand by adding these real-time inputs to the state representation and updating the constraints. It maintains airport operations managing easily and effectively even when conditions are unpredictable and change rapidly.

In addition, we can also explore the deep integration of more advanced artificial intelligence technologies such as deep learning with metaheuristic algorithms to improve the performance and adaptability of the algorithms. This study provides new insights into airport parking space allocation and scheduling optimization, which has important practical significance, especially in improving airport operating efficiency, reducing operating costs and improving flight punctuality. The research results can provide strong decision-making support for airport managers and promote the intelligent development of airport operation management.

The manuscript gives an adequate overview of the experimental settings and surroundings; however, it would be more effectively if it provided more detail the reason the chosen test situations and datasets have been chosen. In particular, the datasets' ability to replicate real airport operations is not adequately explained, which makes it severely to determine how much the results can be applied to real-world situations. A thorough discussion of how the datasets show the range and complexity of typical airport traffic patterns, operational limits, and unexpected events would enhance the paper more powerful. The addition of this information could render readers more certain that the proposed optimization method will perform well in a variety of airport environments.

The results show that the suggested hybrid approach consistently does better than both standalone and traditional models on a number of performance parameters. However, adding statistical significance tests like paired t-tests or ANOVA would make these results more reliable by showing that the improvements shown are not just random. Additionally, a more comprehensive evaluation of the pros and cons of each strategy in different operational situations can be very helpful. For instance, traditional models could perform well while things are stable, however they could struggle as well when issues are changing. On the other hand, the hybrid model's complexity could render it more difficult to scale up in larger circumstances. This kind of thorough examination could render the study more complete and useful in real life.

5 Conclusion

The main finding of this study is that the airport parking stand allocation and scheduling optimization method based on meta-heuristic algorithm can effectively solve the multi-objective optimization problem in airport parking stand allocation and scheduling. By combining particle swarm optimization (PSO) and Q-learning-based reinforcement learning method, the model outperforms the separate PSO, Q-learning and traditional heuristic methods in multiple key indicators such as total scheduling cost, scheduling delay time, and parking stand utilization rate. This study provides new insights and contributions to the field of airport operation management. The research results provide strong evidence for the optimization of airport parking stand allocation and scheduling, and have important practical application value. This method can help airport managers allocate parking stand resources more reasonably, reduce flight delays, and improve parking stand utilization, thereby reducing airport operating costs and improving the overall operating efficiency and service quality of the airport. The research results support and expand the application of existing meta-heuristic algorithms in airport parking stand allocation problems. Traditional metaheuristic algorithms have certain limitations when dealing with complex constraints and multi-objective optimization problems. This study, through innovative model design and algorithm improvement, improves the accuracy of local search while ensuring global search capabilities, providing new ideas for solving such problems and may change the traditional mode of existing airport parking space allocation and scheduling. The limitation of this study is that the experimental data set is mainly based on simulated and improved public data, which is still somewhat different from the complex and changing environment of actual airport operations.

This study integrates basic Particle Swarm Optimization (PSO) with reinforcement learning techniques to solve problems with complicated restrictions and numerous goals. Traditional metaheuristic algorithms like PSO have trouble with these kinds of circumstances. The hybrid method uses PSO's capacity to search across the globe and the learning-based Q-learning framework to make it more flexible and better at making decisions. By using both strategies together, the system can better handle complicated airport scheduling situations and find the best solution for numerous goals than either method could do on its own. This integration allows the suggested model get beyond the problems with traditional standalone metaheuristics, which is in line with the paper's goal of improving performance in real-time optimization environments with limited resources.

Future research can further explore the use of more realistic airport operation data for experiments to improve the practicality and reliability of the model. At the same time, more complex factors in actual operations, such as weather changes and temporary failures of airport facilities, can be incorporated into the model to better adapt to actual scenarios. In addition, it is also possible to explore the deep integration of more advanced artificial intelligence technologies with meta-heuristic algorithms to further improve the performance and adaptability of the algorithm and provide stronger support for airport operation management.

References

[1] Lee S. m-ABCNet: Multi-Modal aircraft motion behavior classification network at airport ramps. IEEE Access. 2024; 12:133982-133993. DOI: 10.1109/access.2024.3462096

[2] Guedan-Pecker F, Ramirez-Atencia C. Airport take-off and landing optimization through genetic algorithms. Expert Systems. 2024; 41(8):e13565. DOI: 10.1111/exsy.13565

[3] Cueto PF, García S, Anjos MF. An efficient solution methodology for the airport slot allocation problem with preprocessing and column-and-row generation. Computers & Operations Research. 2025; 177:106972. DOI: 10.1016/j.cor.2024.106972

[4] Ciftci ME, Özkir V. Integrated optimisation model for airline bank structure and fleet assignment problem. Annals of Operations Research. 2024; 342(1):265-285. DOI: 10.1007/s10479-023-05615-9

[5] Zhang TY, Guo XP, Ji GJ. Permutation flow shop scheduling optimization method based on cooperative games. IEEE Access. 2023; 11:47377-47389. DOI: 10.1109/access.2023.3275533

[6] Zhang Y, He Q, Yang L, Liu CH. An improved tunicate swarm algorithm for solving the MultiObjective optimization problem of airport gate assignments. Applied Sciences-Basel. 2022; 12(16):8203. DOI: 10.3390/app12168203

[7] Chen CY, Chan IH. The stand allocation model for aircraft MRO service provider. Transportation Planning and Technology. 2024; 47(5):788-808. DOI: 10.1080/03081060.2024.2312445

[8] Almonacid B. AutoMH: Automatically create evolutionary metaheuristic algorithms using reinforcement learning. Entropy. 2022; 24(7):957. DOI: 10.3390/e24070957

[9] Scala P, Mota MM, Delahaye D. Air traffic management during rare events such as a pandemic: Paris Charles de Gaulle case study. Aerospace. 2021; 8(6):155. DOI: 10.3390/aerospace8060155

[10] Wang YJ, Liu C, Wang H, Duong V. Slot allocation for a multiple-airport system considering airspace capacity and flying time uncertainty. Transportation Research Part C-Emerging Technologies. 2023; 153:104185. DOI: 10.1016/j.trc.2023.104185

[11] Li YF, Liu YX. Multi-airport system flight slot optimization method based on absolute fairness. Mathematical Biosciences and Engineering. 2023; 20(10):17919-17948. DOI: 10.3934/mbe.2023797

[12] Ozkan R, Samli R. Flood algorithm: a novel metaheuristic algorithm for optimization problems. Peerj Computer Science. 2024; 10:e2278. DOI: 10.7717/peerj-cs.2278

[13] Han X, Zhao PX, Kong DX. Two-stage optimization of airport ferry service delay considering flight uncertainty. European Journal of Operational Research. 2023; 307(3):1103-1116. DOI: 10.1016/j.ejor.2022.09.023

[14] Umale-Nagmote A, Goel C, Lal N. Enhanced intelligent video monitoring using hybrid integration of spatiotemporal autoencoders and convolutional LSTMs. Informatica. 2025; 49(18):51-68. DOI: 10.31449/inf.v49i18.7502

[15] Akinola OO, Ezugwu AE, Agushaka JO, Abu Zitar R, Abualigah L. Multiclass feature selection with metaheuristic optimization algorithms: a review. Neural Computing & Applications. 2022; 34(22):19751-19790. DOI: 10.1007/s00521-022 -07705-4

[16] Feng HL, Hu R, Wang DY, Zhang JF, Wu CT. Bi-objective airport slot scheduling considering scheduling efficiency and noise abatement. Transportation Research Part D-Transport and Environment. 2023; 115:103591. DOI: 10.1016/j.trd.2022.103591

[17] Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019). IEEE Access. 2021; 9:26766-26791. DOI: 10.1109/access.2021.3056407

[18] Hagspihl T, Kolisch R, Fontaine P, Schiffels S. Apron layout planning-Optimal positioning of aircraft stands. Transportation Research Part B-Methodological. 2024; 179:102854. DOI: 10.1016/j.trb.2023.102854

[19] Bi J, Wang FJ, Ding C, Xie DF, Zhao XM. The airport gate assignment problem: A Branch-and-Price Approach for improving utilization of jet ways. Computers & Industrial Engineering. 2022; 164:107878. DOI: 10.1016/j.cie .2021.107878

[20] Mello-Román JD, Hernández A. KPLS Optimization with nature-inspired metaheuristic algorithms. IEEE Access. 2020; 8:157482-157492. DOI: 10.1109/access.2020.3019771

[21] Jamellah, K. M. Real-Time computational efficiency vehicle detection and counting utilizing the background subtraction technique and non-maximum suppression techniques. Informatica. 2025; 49(18):29-38. DOI: 10.31449/inf.v49i18.7426

[22] Gharehchopogh FS. Quantum-inspired metaheuristic algorithms: comprehensive survey and classification. Artificial Intelligence Review. 2023; 56(6):5479-5543. DOI: 10.1007/s10462-022-10280-8