Check for updates

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–11
© The Author(s) 2024
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/03611981241242352
journals.sagepub.com/home/trr
Sage

Gate Assignment Algorithm for Airport Peak Time Based on Reinforcement Learning

Chenwei Zhu[1], Zhenchun Wei[1,2,3], Zengwei Lyu[1,2,3], Xiaohui Yuan[4], Dawei Hang[1], and Lin Feng[1]

Abstract

In existing airport gate allocation studies, little consideration has been given to situations where gate resources are limited during peak periods. Under such circumstances, some flights may not be able to make regular stops. In this paper, the airport gate assignment problem under peak time is investigated. We propose a gate pre-assignment model to maximize the gate matching degree and the near gate passenger allocation rate. Besides, to minimize the pre-assignment gate change rate, we propose a dynamic reassignment model based on the pre-assignment model. By considering the non-deterministic polynomial hard (NP-hard) property of this problem, a gate assignment algorithm based on proximal policy optimization (GABPPO) is proposed. The simulation results show that the algorithm can effectively solve the gate shortage problem during the airport peak period. Compared with the adaptive parallel genetic, deep Q-network, and policy gradient algorithms, the target value of solutions obtained by the proposed algorithm in the near gate passenger allocation rate is increased by 5.7%, 3.6%, and 7.9%, respectively, and the target value in the gate matching degree is increased by 10.6%, 4.9%, and 11.5% respectively.

Keywords

airfield and airspace capacity and delay, airport operations, aviation

With the rapid development of the civil aviation industry in recent years, the number of passengers and flights has been increasing, and the reasonable assignment of airport gates has become an important issue, which is especially prominent during peak periods. How to allocate a suitable gate for each arriving flight under limited airspace resources to enhance passenger satisfaction and achieve the service efficiency of the airport is called "gate assignment." The gate assignment problem is generally a matter of pre-assignment and reassignment. The issue of gate assignment plays a key role in the efficient use of gates and, therefore, the assignment of gates needs to be carefully arranged, which greatly affects the efficient operation of the airport and passenger satisfaction. The gate pre-assignment process considers the type of flight, flight attributes, arrival/departure time, the number of passengers, airport rules, and so forth. All incoming flights are pre-allocated for the next day, and passengers are informed of the gate availability in advance. The gate reassignment problem is that, in real-time operations, flight arrival and departure times frequently change for various reasons. When a delayed flight is identified, the gate must be reassigned for the delayed flight. Effective gate reassignment methods are critical to aviation scheduling problems. In current airline scheduling, the gate reassignment process is mainly carried out by manually adjusting the current reallocation plan, using computer-aided tools. The gate assignment problem has been a hot research topic and it is a core aspect of airport operations and management (*1*, *2*).

[1]School of Computer Science and Information Engineering, Hefei University of Technology, Heifei, China
[2]Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei, China
[3]Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei, China
[4]Department of Computer Science and Engineering, University of North Texas, TX, USA

Corresponding Author:
Zhenchun Wei, weizc@hfut.edu.cn

There are various hard and soft constraints in the gate assignment process. Hard constraints refer to the rules that must be followed in the gate assignment process, such as uniqueness constraints and safety buffer time constraints. Soft constraints are some constraints that can be violated under certain circumstances, such as access conflict constraints, aircraft type matching constraints, and so forth.

Many scholars have studied the problem of gate pre-assignment and reassignment. Cai et al. proposed a bi-objective constrained optimization problem to minimize the total walking distance of passengers and maximize robustness (*3*). They proposed a two-stage neighborhood search algorithm to solve it. Benlic et al. established multiple objective functions and proposed a heuristic algorithm based on a breakthrough local search framework to solve this problem (*4*). Zhang and Klabjan established multi-commodity network flow models and proposed a heuristic algorithm to solve the problem (*5*). Pternea and Haghani proposed a new binary integer programming model to solve the gate reassignment problem, which evaluates the success rate of passenger transfers as the optimization objective (*6*). In a study considered from the joint passenger and airport perspectives, Deng et al. used a weighted approach to transform the four optimization objectives—minimum walking distance for passengers, minimum variance of idle time for each parking space, minimum number of flights to stop on the apron, and the most reasonable utilization of large parking spaces—into a single objective optimization problem and solved it using an improved particle swarm algorithm (*7*). L'Ortye et al. considered airport airside and landside constraints on facility capacity, and they finally obtained a robust gate assignment scheme based on a mixed integer linear approach (*8*). To simultaneously guarantee the robustness of the airport operation plan as well as to improve the efficiency of gate utilization, Kim et al. introduced an overlapping opportunity constraint and solved the trade-off between the two objectives by a designed branch-and-price algorithm (*9*).

Liu et al. considered that, with the increase in air transportation volume, the problem of insufficient airport boarding gates has become increasingly prominent (*10*). Unreasonable gate assignments will affect resource utilization, reduce passengers' satisfaction, and even lead to chaotic airport operations, restricting the rapid development of airports. Therefore, it is extremely important to adopt a reasonable and efficient method to optimize the assignment of existing gates. This study reviews the current research status of gate optimization and summarizes three main research methods—model solving based on 1) accurate algorithms, 2) heuristic algorithms, and 3) meta-heuristic algorithms. Tang proposed a gate reassignment model to handle temporary gate shortages and stochastic flight delays, which is implemented by allowing violations of the gate reassignment constraints (*11*). The effectiveness was verified by conducting numerical tests on the operational data of an international airport. Few existing studies use reinforcement learning methods to solve the problem of gate assignment. Muhafız Yılıdz et al. used Q-learning to solve the problem of gate assignment (*12*). Q-learning maintains a state-action value function (Q-value function) and learns the optimal strategy by continuously updating this table. It is suitable for small-scale, discrete state and action space reinforcement learning problems, but faces limitations in algorithm expansion.

The above study did not take into account the possibility that some flights may not have a gate during peak traffic hours. This problem is a sequential decision-making problem that is highly dynamic and suitable for deep reinforcement learning to solve. If soft constraints are violated without affecting normal airport operations, gate resources can be utilized more efficiently (*13*, *14*). Although existing research considers the impact of soft constraints on airport flights, few studies maximize resource utilization by violating soft constraints during peak traffic. In addition, many existing studies use heuristic algorithms and integer programming models to solve the gate assignment problem, while there are fewer papers using reinforcement learning algorithms to solve this problem (*12*, *15*, *16*).

The contributions of this article include:

1) The gate assignment problem is a sequential decision-making problem. This paper presents a Deep Reinforcement Learning (DRL)-based gate assignment algorithm to solve the gate assignment.
2) From the perspective of passengers and the airport, we use the weighting method to transform the bi-objective optimization problem of each gate matching degree and the near gate allocation rate into a single-objective optimization problem in the gate pre-assignment stage. According to the operation rules of different airports, different weights are selected, and the proposed method is universal.
3) Based on pre-assignment, we construct a real-time gate dynamic re-assignment model. The change rate of pre-assigned gates is taken as the optimization objective, and the preference for each objective is expressed by setting different weighting coefficients.

The remainder of this paper is organized as follows. The next section presents our system model and the corresponding optimization objectives. In the section after that, the reinforcement learning-based gate assignment

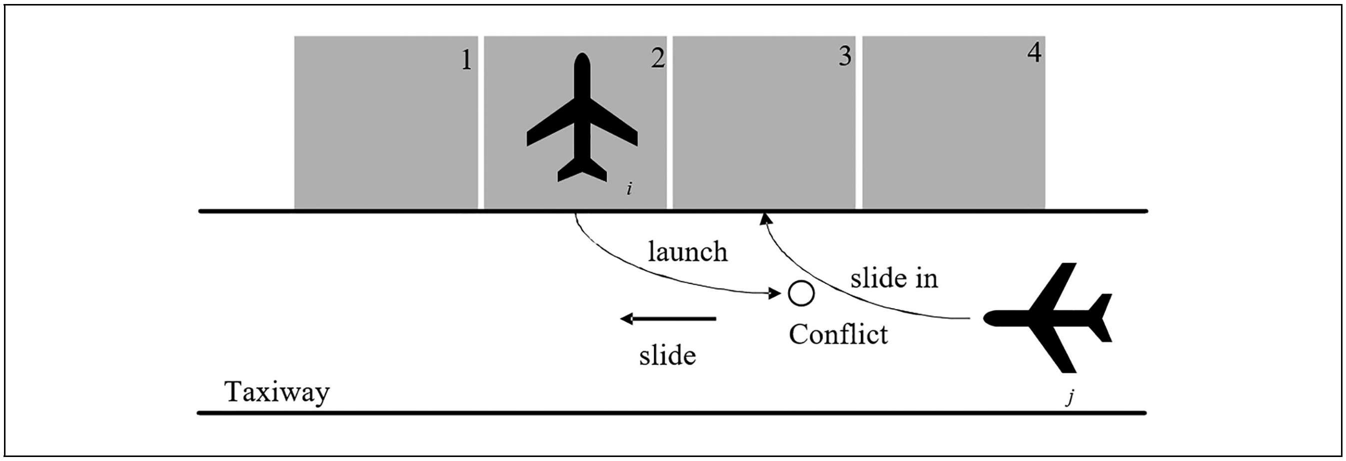

Figure 1. Flight in and out conflict.

algorithm based on proximal policy optimization (GABPPO) is proposed and the Markov decision process is described in detail. The penultimate section gives experimental settings and comparisons of GABPPO with several more advanced algorithms on benchmark problems. Conclusions are drawn in the final section.

System Model

Parameters Definition

The set of flights is denoted as $A = \{a_1, a_2, ..., a_N\}$ and N is the number of flights.

The set of gates is denoted as $G = \{g_1, g_2, \cdots g_M\}$ and M denotes the number of gates.

PR_i and PD_i represent the scheduled arrival and departure times of flight i, respectively.

The variable x_{ik} denotes whether flight i is assigned to gate k. If flight i is assigned to gate k, then x_{ik} is 1, otherwise 0.

The variable O_i indicates the type of flight i. $O_i = 1$ indicates that flight i is a large aircraft, while $O_i = 0$ indicates that it is a small aircraft.

Similarly, if gate k is large, then $G_k = 1$, otherwise $G_k = 0$.

If gate k is adjacent to gate l, then $z_{kl} = 1$, otherwise $z_{kl} = 0$.

The variable E_{ik} denotes the moment when flight i enters gate k, $E_{ik} = \mathrm{PR}_i x_{ik}$.

The variable D_{jl} denotes the moment when flight j leaves gate l, $D_{jl} = \mathrm{PD}_j x_{jl}$.

Model Constraints

To ensure the safe arrival of flights, it is necessary to fully consider the safety requirement rules, operation rules, and other information in the process of gate assignment. In this paper, the constraints required for gate assignment for flights are as follows:

$$\sum_{k=1}^{M} x_{ik} = 1, \forall i = 1, ..., N, k = 1, ..., M \tag{1}$$

$$\left(PR_j - PD_i - \alpha\right)\left(PR_i - PD_j - \alpha\right)x_{ik}x_{jk} \leqslant 0 \tag{2}$$

$$(G_k - O_i)x_{ik} \geqslant 0, \forall i = 1, ..., N, k = 1, ..., M \tag{3}$$

$$\left|E_{ik} - D_{jl}\right| \geqslant \beta x_{ik}x_{jl}z_{kl} \tag{4}$$

$$\left|D_{jl} - D_{ik}\right| \geqslant \beta x_{ik}x_{jl}z_{kl} \tag{5}$$

$$\left|E_{jl} - E_{ik}\right| \geqslant \beta x_{ik}x_{jl}z_{kl} \tag{6}$$

Equation 1 represents the uniqueness constraint, that is, a flight must be assigned to a gate when it enters the airport, and it can be assigned to only one gate, where x_{ik} denotes whether flight i is assigned to gate k.

Equation 2 indicates that a specific time interval must be maintained between neighboring flights stopping at the same gate, that is, the minimum safety buffer time α, where PR_i and PD_i denote the expected time of arrival and expected time of departure of flight i, respectively.

Equation 3 represents the type matching constraint, that is, large flights can be assigned to large gates only, and small flights can be assigned to large or small gates, where G_k and O_i denote the type of gate and the type of flight, respectively.

Equation 4 represents the access conflict constraint, as shown in Figure 1. Flight i and j are assigned to adjacent gate 2 and gate 3. Flight i leaves gate 2 at the time D_{i2} and flight j enters gate 3 at the time E_{j3}, and, if the time interval between them is less than the minimum safety interval β, an access conflict occurs, where z_{kl} denotes gate k is adjacent to gate l.

Equations 5 and 6 represent the dual-out and dual-in conflict constraints, as shown in Figure 2. Flight i and j

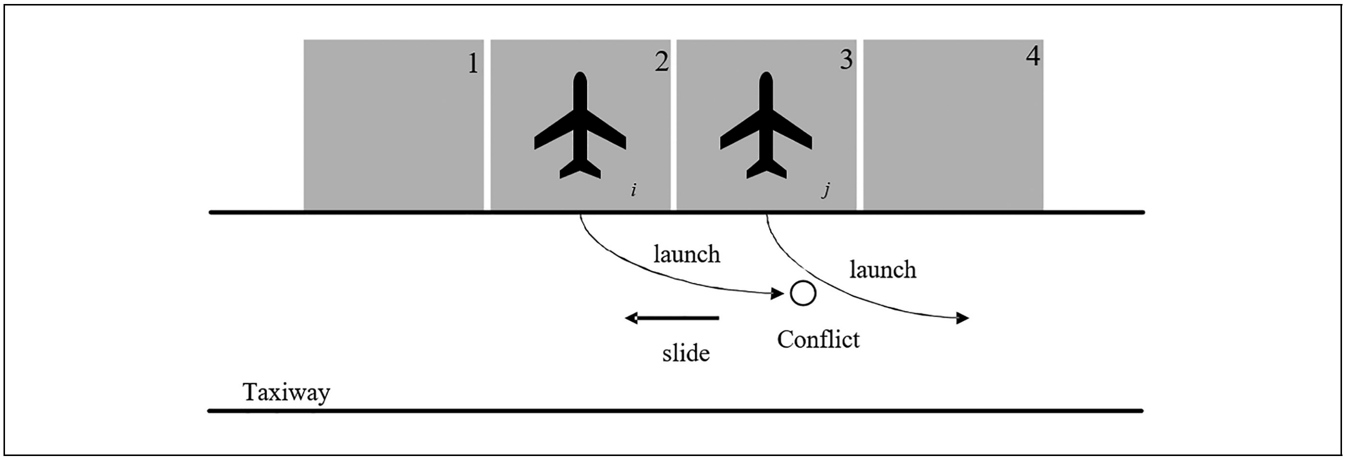

Figure 2. Flight dual-out conflict.

are assigned to adjacent gate 2 and gate 3. Flight i leaves gate 2 at the time D_{i2} and flight j leaves gate 3 at the time D_{j3}, and, if the time interval between the departure of two flights is less than the minimum safety interval β, a dual-out conflict occurs. Similarly, the time interval between two flights assigned to adjacent gates must also be greater than the minimum safety interval β.

Pre-Assignment Objective Function

To ensure the regular operation of the airport and the availability of a gate for each flight, soft constraints should be satisfied as much as possible without violating the hard constraints. Meanwhile, the near gate passenger allocation rate is an important indicator of airport passenger satisfaction. Therefore, this paper considers the gate matching degree and the near gate passenger allocation rate as the optimization objectives.

The degree of satisfaction of the soft constraints by the allocation scheme is defined as the gate matching degree, which is used to reflect the degree of impact caused by the gate assignment scheme on the airport operation. The gate matching degree consists of three parts: 1) access conflict constraint, 2) dual-out and dual-in conflict constraints, and 3) aircraft type matching constraint. In some cases, violation of soft constraints can alleviate the problem of airport resource shortage. But, at the same time, we must also fully ensure the operational efficiency of the airport. In other words, we do not allow violations of soft constraints to happen at discretion. Constructing the gate matching degree weight matrix $Z = Z_1 + Z_2 + Z_3$:

$$Z_1 = \begin{bmatrix} o_{1,1} & \cdots & o_{1,M} \\ \vdots & \ddots & \vdots \\ o_{N,1} & \cdots & o_{N,M} \end{bmatrix} \tag{7}$$

$$Z_2 = \begin{bmatrix} h_{1,1} & \cdots & h_{1,M} \\ \vdots & \ddots & \vdots \\ h_{N,1} & \cdots & h_{N,M} \end{bmatrix} \tag{8}$$

$$Z_3 = \begin{bmatrix} l_{1,1} & \cdots & l_{1,M} \\ \vdots & \ddots & \vdots \\ l_{N,1} & \cdots & l_{N,M} \end{bmatrix} \tag{9}$$

where

$o_{i,k}$, $h_{i,k}$, and $l_{i,k}$ indicates the values of row i and column k in Z_1, Z_2, and Z_3, respectively,

$o_{i,k}$ and $h_{i,k}$ indicates whether the flight i allocated to the gate k meets the access conflict constraint and the dual entry dual exit conflict constraint, respectively (1 if satisfied, 0 otherwise), and

$l_{i,k}$ indicates if small flight i is assigned to a large gate, it is 0; if small flight i is assigned to a small gate or large flight i is assigned to a large gate, it is 1, indicating that the aircraft type matching is high at this point. (It is worth noting that in our model, the large flight is not allowed to be assigned to the small gate.)

The gate matching degree F_1 is defined as follows, maximizing the gate matching degree as the first objective function. Our goal is to minimize the number of violations of soft constraints while fully ensuring the operational efficiency of the airport.

$$F_1 = \frac{\sum_{i=1}^{N} \sum_{k=1}^{M} x_{ik} Z(i,k)}{3N} \tag{10}$$

where

$Z(i,k)$ indicates the value of the ith row and kth column in the matrix Z.

The higher the gate matching degree, the smaller the impact on airport operation, and vice versa.

Passenger satisfaction is one of the most important indicators for evaluating the results of gate assignment. Passenger satisfaction will be higher when the flight is assigned to a near gate because the walking distance of passengers is shorter. In this paper, the second objective function is to maximize the near gate passenger allocation rate.

$$F_2 = \frac{\sum_{i=1}^{N}\sum_{k \in G_n}(p_{i0} + p_{0i})x_{ik}}{\sum_{i=1}^{N}(p_{i0} + p_{0i})} \tag{11}$$

where

G_n indicates the set of near gates (we assume that the gate where the passengers are located is a binary event, that is, the gate where the passengers are located either belongs to the set of near gates or to the set of far gates), and

p_{i0} and p_{0i} indicates the number of passengers entering and departing flight i, respectively.

According to the elaboration of the optimization objectives as well as the constraints, the interests of the airport and passengers are considered comprehensively. Maximizing the near gate allocation rate and maximizing the gate matching degree are taken as the combined optimization objectives, and the combined optimization objectives are expressed as follows:

$$\begin{aligned}\max F_{pre} = W\frac{\sum_{i=1}^{N}\sum_{k \in G_n}(p_{i0} + p_{0i})x_{ik}}{\sum_{i=1}^{N}(p_{i0} + p_{0i})} \\ + (1 - W)\frac{\sum_{i=1}^{N}\sum_{k=1}^{M}x_{ik}Z(i,k)}{3N} \\ \text{s.t}: (1), (2), (3), (4), (5), (6)\end{aligned} \tag{12}$$

Reassignment Problem Definition

Flight arrival and departure times frequently change for various reasons in real-time operations. The gate must be reassigned for the delayed flight and subsequent flights when a delayed flight is identified. Therefore, in this paper, based on the two optimization objectives of pre-assignment, the change rate F_3 of the pre-assignment gate is added as the optimization objective, that is, the original gate assignment result is kept unchanged as much as possible.

$$F_3 = \frac{\sum_{i=1}^{N}\sum_{k=1}^{M}x_{ik}^{num}x_{ik}}{N} \tag{13}$$

where

x_{ik}^{num} i indicates whether flight i in the pre-assignment and re-assignment has the same gate, 1 if the same, 0 if different.

Therefore, the optimization objective of gate re-assignment combination is expressed as follows:

$$\begin{aligned}\max F_{re} = W\frac{\sum_{i=1}^{N}\sum_{k \in G_n}(p_{i0} + p_{0i})x_{ik}}{\sum_{i=1}^{N}(p_{i0} + p_{0i})} + V\frac{\sum_{i=1}^{N}\sum_{k=1}^{M}x_{ik}Z(i,k)}{3N} \\ + (1 - W - V)\frac{\sum_{i=1}^{N}\sum_{k=1}^{M}x_{ik}^{num}x_{ik}}{N} \\ \text{s.t}: (1), (2), (3), (4), (5), (6)\end{aligned} \tag{14}$$

where

W and V indicates weighting coefficients that can be set according to the importance of each objective.

Gate Assignment Algorithm Based on Reinforcement Learning

Most of the previous studies have adopted traditional algorithms for solving the problem. However, traditional algorithms are difficult to explore from one locally optimal solution to another, so they easily fall into local optima. In contrast, deep reinforcement learning algorithms are extremely suitable for solving complex sequential decision problems (*17*). They can converge and obtain an approximate global optimal solution under finite iterations by continuously learning action strategies through the interaction between an agent and its environment (*18*, *19*). Reinforcement learning algorithms are particularly suitable for the gate assignment problem. Each aircraft lands at the airport in turn. The agent decides to allocate gates for each aircraft and obtains the corresponding rewards. The goal of the gate assignment problem is to have the agent continuously trained so that the cumulative reward value obtained after multiple decision-making steps is maximized. The proximal policy optimization algorithm is more suitable for solving discrete action problems among the reinforcement learning algorithms (*20*). Therefore, this paper proposes a proximal policy optimization-based gate assignment algorithm to solve the aircraft space pre-assignment and reassignment problems.

Markov Decision Process Modeling

When building a Markov decision model, it is first necessary to define the time step. The interaction process between the agent and the environment can be divided into several discrete time steps. The moment of arrival of a flight at the airport is defined as a time step, so the time step is the set of a series of discrete moments, denoted as $T = \{1, 2, \ldots, t, t+1, \ldots, N\}$. There are three key elements of the reinforcement learning approach that need to be identified, namely the state space, the action space, and the definition of immediate reward (*21*, *22*).

1) The State Space

The state space at time step t is defined as $S_t = \ <B(t), G^{pro}, AD(t), FP(t)>$.

$B(t)$ indicates the occupation time status of each gate resource at time step t, that is, $B(t) = (B_1(t), B_2(t), \cdots, B_M(t))$.

If the gate k of time step t is not occupied, then $B_k(t) = 0$; otherwise $B_k(t)$ represents the time that the gate k still needs to be occupied after time step t.

G^{pro} refers to the attributes of each gate, that is, distance attribute and size attribute.

$AD(t)$ indicates the arrival and departure time status of the flight t at time step t.

$FP(t)$ indicates the flight attribute status at time step t.

The flight attribute $FP(t) = (an_t, dn_t, m_t)$, where an_t and dn_t are the number of people boarding and disembarking from flight t, respectively, and denotes the model size attribute of flight t. Together, they form the state space S_t at moment t.

2) The Action Space

The action space describes the set of actions a_t that can be taken by the agent at time step t, where $a_t \in \{1, 2, \ldots, M\}$ means that flight t must select an action from the set $\{1, 2, \ldots, M\}$. If $a_t = k$, it means that the current flight t is assigned to the kth gate, where the agent can take action a_t that needs to be scaled down according to the hard constraints.

3) The Immediate Reward

For each action performed by the agent, an immediate reward r is obtained, and the immediate reward should be related to the optimization goal, so the immediate reward of the pre-assignment model at time step t is:

$$r_t = W \frac{\sum_{k \in G_n} (p_{t0} + p_{0t}) x_{tk}}{\sum_{t=1}^{N} (p_{t0} + p_{0t})} + (1 - \mathrm{W})/3 * \sum_{k=1}^{M} x_{tk}(o_{t,k} + h_{t,k} + l_{t,k}) \tag{15}$$

Similarly, the immediate reward of the re-assignment model at time step t is defined as:

$$r_t = W \frac{\sum_{k \in G_n} (p_{t0} + p_{0t}) x_{tk}}{\sum_{t=1}^{N} (p_{t0} + p_{0t})} + \frac{(1-W-V)}{N} \sum_{k=1}^{M} x_{tk}^{num} x_{tk} + V \sum_{k=1}^{M} x_{tk}(o_{t,k} + h_{t,k} + l_{t,k})/3 \tag{16}$$

where
W and V indicates weighting factors.

Gate Assignment Algorithm Based on Proximal Policy Optimization (GABPPO)

When the tth flight arrives at the airport, it needs to be assigned to a gate. As shown in Figure 3, firstly, the agent observes the environment state s_t, and follows the policy network π_θ to select the action a_t for the flight, to assign a gate from the action space. Then, the environment will move to the next state s_{t+1} according to the taken action and issue an immediate reward r_t to the agent. The process of assigning a gate for subsequent flights repeats the above steps. The agent will get an immediate reward after assigning a gate, so the accumulated reward of the agent from moment t is defined as:

$$G_t = r_t + \gamma r_{t+1} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{17}$$

where
$\gamma \in (0, 1]$ indicates the discount factor.

The GABPPO algorithm is based on the actor-critic framework, which uses multiple training before updating and reuses past experience samples to improve the learning efficiency, thus using two copies of the strategy π_θ and π_{old}. π_{old} is a backup of the π_θ strategy—each training only updates the network parameters in the π_θ strategy. The actor network outputs as large an action advantage as possible, introducing the advantage function and is defined as:

$$\hat{A}_t = r_t + \gamma r_{t+1} + \cdots + \gamma^{n-1} r_{t+n+1} + \gamma^n v(s') - v(s) \tag{18}$$

The ratio of the two strategies is defined as:

$$\mathrm{ratio}_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\mathrm{old}}(a_t|s_t)} \tag{19}$$

The resulting loss function of the actor network is obtained as follows:

$$L_{\mathrm{a}} = \hat{E}_t\left[\min\left(\mathrm{ratio}_t(\theta)\hat{A}_t, \mathrm{clip}(\mathrm{ratio}_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{20}$$

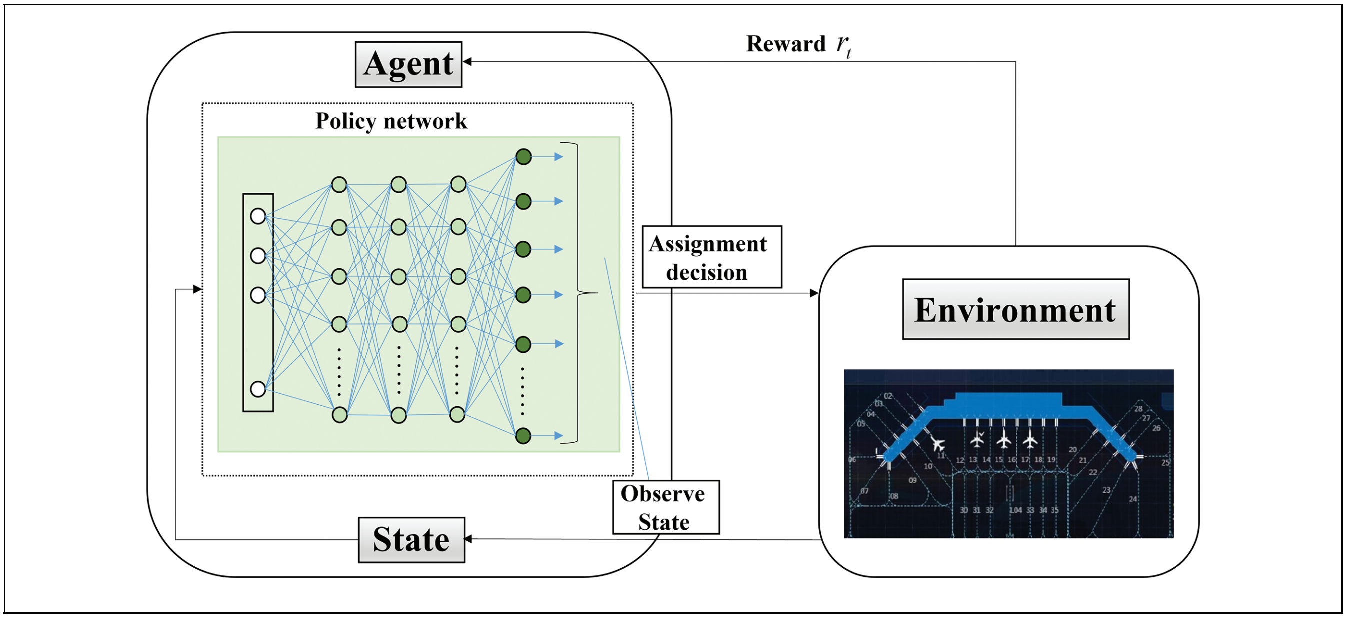

Figure 3. Reinforcement learning process.

where
ϵ indicates an adjustable parameter whose function is to limit the value of ratio_t.

The critic network can directly estimate the state value $v(s)$ of current state s, so G_t can be used as the true value of the critic network estimate—the loss function of the critic network is the difference between the calculated discounted reward and the critic network estimate, that is, Temporal Difference (TD)-error. The critic network loss function is defined as follows:

$$L_c = \frac{1}{n}\sum_{t=1}^{n}(G_t - v(s_t))^2 \tag{21}$$

where
n indicates a total of n time steps of updates are currently being performed.

Reinforcement learning aims to find the optimal policy network π_θ that maximizes the cumulative reward G_t obtained by the agent in any state. The policy network π_θ is continuously trained according to the loss function during the reinforcement learning process until a fit is achieved. The pre-assignment algorithm is shown in Algorithm 1.

Firstly, the agent obtains the state from the environment. According to the state space modeling, the flight and gates information is modeled as a two-dimensional matrix for input into the policy network. Secondly, the state space is input into the policy network π_θ, which outputs the probability of taking each action in the action space and filters all gates that violate the hard constraints. Finally, the policy network uses a roulette to decide the action based on the probability of each action as the result of assigning a gate to that flight.

When a flight is delayed, gate reassignment is required. Based on the pre-assignment model, the optimization objective of increasing the change rate of gates is proposed as a reassignment model and modeled as a Markov decision model to obtain an immediate reward r_t. When the pre-assignment result of a delayed flight affects assigning gates of subsequent flights, the pre-assignment process shown in Algorithm 1 is executed again to obtain the reassignment result.

Algorithm 1: GABPPO

Input: Flight information sheet, gate occupancy information
Output: Pre-assignment results
1: Model the gate pre-assignment model as a Markov decision model. The immediate rewards r_t after each time step can be obtained according to the pre-assignment model.
2: Initialize the actor network parameters π_θ and the critic network parameters π_ϕ.
3: Initialize $ep \leftarrow 1, t \leftarrow 1$.
4: While $ep \leqslant EP\_MAX$ do
5: While $t \leqslant N$ do
6: Input state s_{t-1} into the actor network π_θ and collect the state, action, and immediate reward at the tth time step as $\{s_t, a_t, r_t\}$.
7: $t \leftarrow t + 1$
8: End while
9: Calculate L_a by using the collected states, actions, and immediate rewards for N time steps and update π_θ using gradient descent multiple times.
10: Calculate L_c, and similarly update the critic network π_ϕ several times.
11: $ep \leftarrow ep + 1$
12: End while

Table 1. Pre-Assignment Calculation Results of Four Algorithms under Different Evaluation Indexes

Evaluation index	W = 0.3	W = 0.4	W = 0.5	W = 0.6	W = 0.7
GABPPO					
Gate matching degree	**95.56%**	**91.11%**	**92.22%**	90.62%	**90.22%**
Near gate allocation rate	71.48%	**73.78%**	**76.61%**	**78.42%**	**81.37%**
DQN					
Gate matching degree	94.44%	**91.11%**	91.11%	**92.32%**	88.89%
Near gate allocation rate	**72.80%**	71.48%	73.01%	73.21%	75.48%
APGA					
Gate matching degree	89.23%	89.02%	87.24%	88.86%	86.48%
Near gate allocation rate	68.32%	68.62%	69.26%	70.83%	70.86%
Policy gradient					
Gate matching degree	91.11%	88.89%	86.67%	85.62%	87.77%
Near gate allocation rate	66.18%	67.96%	68.70%	68.90%	71.20%

Note: APGA = adaptive parallel genetic algorithm; DQN = deep Q-network; GABPPO = gate assignment algorithm based on proximal policy optimization. The bold entries in Table 1 indicate that the algorithm performs best on the corresponding evaluation index with different weighting factors W (columns).

Experiments and Evaluation

Experiment Settings

This section describes the experimental simulation environment and the parameter settings of GABPPO. The python version is 3.7.0, and the computing platform uses an AMD R5 series six-core processor with 16G memory. The specific parameters of this experiment are set as follows: minimum safe buffer time $\alpha = 15\,\text{min}$, the safety interval $\beta = 3\,\text{min}$ to avoid conflict, and the maximum number of iterations $EP\_MAX = 10000$. The actor and critic learning rates are set to 0.001 and 0.002, respectively. The parameter ε in the clip function is set to 0.2, and the discount factor γ is set to 0.9.

In this paper, a medium-sized airport in China is used for the study, and the flight data from 8:00 a.m. to noon during the peak period of a certain day are selected for model validation. In the experiment, there are a total of 30 sets of flight data in the 4 h period. The flight data include flight number, scheduled arrival time, scheduled departure time, passenger number, flight type, and so forth. There are ten near gates and five far gates in the airport. We assume flights arrive at the airport with sufficient gate resources for allocation, without considering the resource constraint problem.

During the peak period of the airport on a weekday, some flight schedules are changed as a result of air traffic control, weather, aircraft failure, and other factors requiring the reassignment of aircraft to gates. The process of reassignment is carried out after the delays have been gathered. We can determine the maximum amount of delay in advance of 4 h.

Pre-Assignment Simulation Results and Comparative Analysis

Simulation experiments are performed to evaluate the performance of the GABPPO algorithm and compare it with the adaptive parallel genetic algorithm (APGA) (*23*), the deep Q-network (DQN) algorithm (a representative deep reinforcement learning method), and the policy gradient algorithm (*24*). The evaluation metrics were selected as the gate matching degree and the near gate allocation rate, corresponding to Equation 10$F1$ and Equation 11$F2$, respectively. The target data obtained by running the four algorithms 20 times for different evaluation metrics are shown in Table 1. The observation and analysis show that, with different values of W, the aircraft matching degree and passenger near-airport assignment rate show different up and down trends. In the actual application, various settings of W can achieve different degrees of allocation requirements for the gate matching degree and near-flight passenger allocation law. In this paper, W is selected as 0.5 as the weight coefficient of the pre-assignment model.

Figure 4 shows the convergence performance of the proposed proximity policy optimization based gate assignment algorithm. As can be seen from the figure, in the GABPPO algorithm, the objective function value increases with the number of episodes, and the initial objective function value is about 0.710, and, after more than 8,000 episodes, the objective function value gradually converges and maintains about 0.845, which shows that the GABPPO algorithm training effect is significant.

We first apply the GABPPO algorithm based on the established pre-assignment and its Markov decision model to obtain the Gantt chart for the pre-assignment of gates shown in Figure 5. Each flight has a corresponding number, where Nos. 0–9 are near gates and Nos. 10–14 are far gates. An incomplete match indicates that the flight violated a soft constraint when assigning gates.

In response to the assignment result, it can be seen that more flights tend to be assigned to the near gates, and the passenger assignment rate in the near gates has reached 76.61%, which can improve passenger

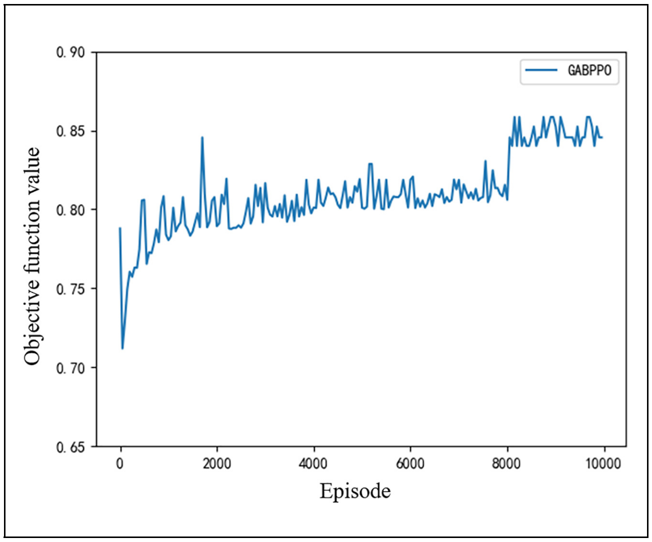

Figure 4. Convergence performance of gate assignment algorithm based on proximal policy optimization (GABPPO) algorithm.

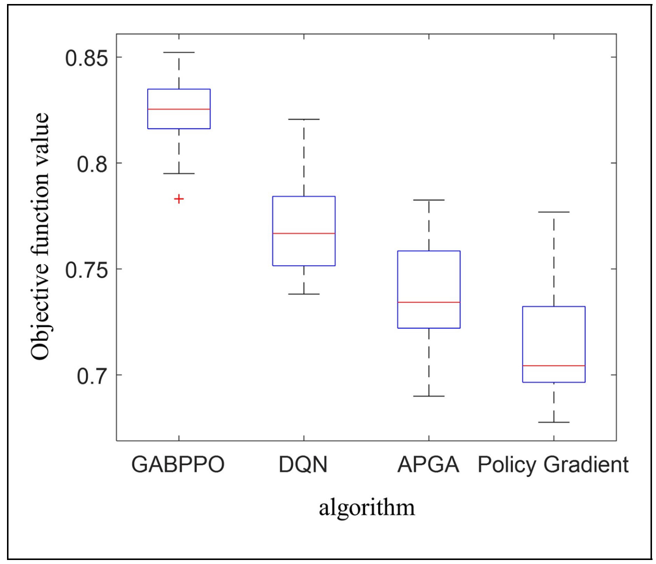

Figure 6. Comparison of objective function value of four algorithms in gate pre-assignment.

Note: APGA = adaptive parallel genetic algorithm; DQN = deep Q-network; GABPPO = gate assignment algorithm based on proximal policy optimization.

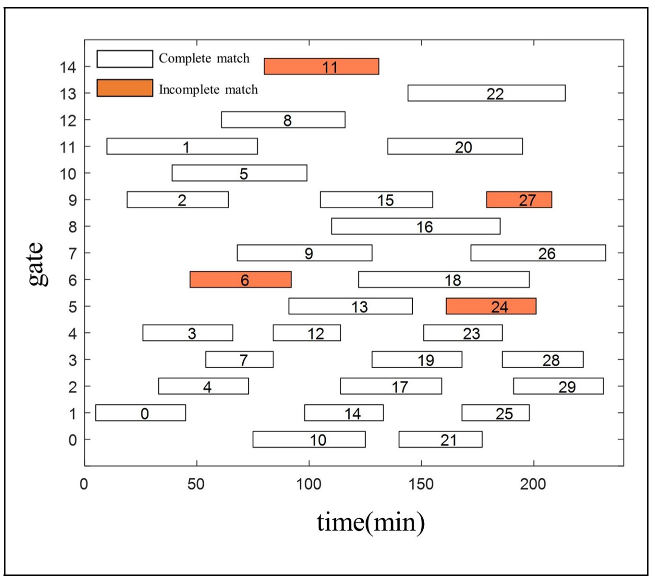

Figure 5. Gantt diagram of gate pre-assignment.

satisfaction. Secondly, under the premise of ensuring that every aircraft is assigned to a gate, the matching degree of gates reaches 92.22%, which minimizes the impact of the allocation scheme on airport operation and can effectively solve the problem of shortage of aircraft space resources.

To compare the four algorithms more intuitively, this section compares the four algorithms in relation to objective function values through box plots. From Figure 6, it can be seen that the GABPPO algorithm obtains higher quality solutions, so the GABPPO algorithm proposed in this paper can effectively improve passengers' satisfaction and effectively solve the problem of shortage of gates during the peak period of the airport.

As shown in Table 1, the solution obtained by the GABPPO algorithm is 5.7%, 3.6%, and 7.9% better than APGA, DQN, and policy gradient algorithms, respectively, in relation to gate matching degree, so the proposed algorithm can effectively solve gate resources shortage problem in the airport peak traffic. At the same time, the solution obtained by the GABPPO algorithm is 10.6%, 4.9%, and 11.5% higher than APGA, DQN, and policy gradient algorithms, respectively, in relation to near gate allocation rate, so the proposed algorithm can improve passenger satisfaction.

Reassignment Simulation Results and Comparative Analysis

During the peak period of the airport, some flight plans were changed because of weather and other factors, so the result of gate assignment was changed. In this section, $W = 0.3$ and $V = 0.3$ are selected as the weighting coefficients for the reassignment model, and the results of the gate reassignment are represented in a Gantt chart, as shown in Figure 7.

The results of the comparison between the GABPPO algorithm and the three typical algorithms for gate reassignment are shown in Table 2. The change rate of gates—that is, the third objective value—of solutions obtained by the GABPPO algorithm is smaller. It can

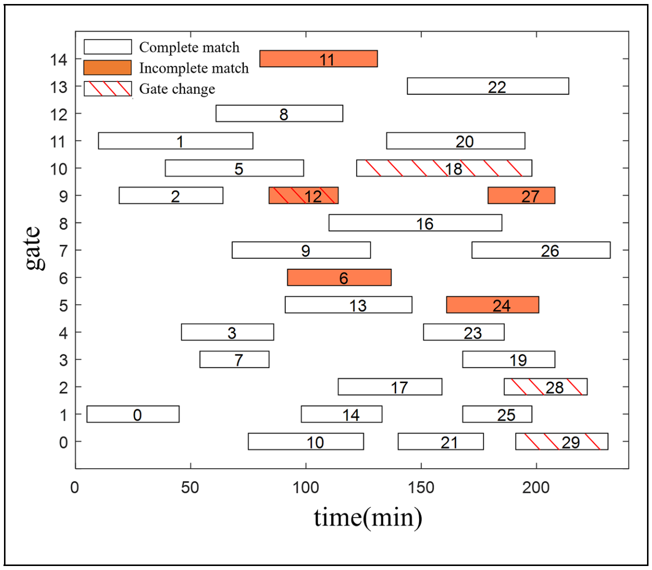

Figure 7. Gantt diagram of gate reassignment.

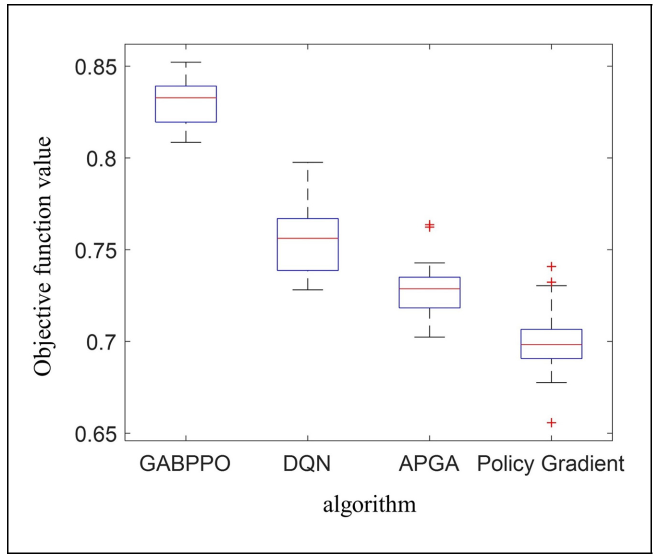

Figure 8. Comparison of objective function value of four algorithms in gate reassignment.
Note: APGA = adaptive parallel genetic algorithm; DQN = deep Q-network; GABPPO = gate assignment algorithm based on proximal policy optimization.

Table 2. Comparison of Optimal Solutions of Four Algorithms

Algorithm	Gate matching degree	Near gate allocation rate	Change rate of gates
GABPPO	**94.44%**	**75.98%**	**13.33%**
DQN	89.28%	73.30%	17.85%
APGA	87.21%	70.68%	16.42%
Policy gradient	85.71%	68.90%	17.85%

Note: APGA = adaptive parallel genetic algorithm; DQN = deep Q-network; GABPPO = gate assignment algorithm based on proximal policy optimization. The bold entries in Table 2 indicate that the algorithm (rows) performs best on the corresponding evaluation index(columns).

also be seen that the objective value of solutions obtained on the gate matching degree and the near gate passenger allocation rate is larger than that of the solution obtained by the other algorithms. Solutions obtained by the GABPPO algorithm are better than solutions obtained by the other three algorithms.

The box plot is used to compare the four algorithms in relation to objective function values. From Figure 8, it can be seen that the GABPPO algorithm obtains solutions with higher quality than the other three algorithms, no matter the maximum value, minimum value, or median.

Conclusions

For the problem of shortage of gate resources during the peak period of the airport, and taking into account the interests of passengers, a gate pre-assignment model is constructed. Meanwhile, to solve the problem of gate occupation conflict caused by flight delays, a dynamic reassignment model is constructed based on the pre-allocation model. The GABPPO algorithm is proposed to solve the above optimization problem effectively. The experimental simulation results show that the GABPPO algorithm proposed in this paper outperforms the baseline algorithms and can significantly improve passenger satisfaction while solving the problem of the shortage of gate resources.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: C. Zhu; data collection: Z. Wei; analysis and interpretation of results: C. Zhu, Z. Lyu; draft manuscript preparation: X. Yuan, D. Hang, L. Feng. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Key Research and Development Program of China (2023YFB3907302).

ORCID iDs

Zhenchun Wei https://orcid.org/0000-0003-2751-6501
Xiaohui Yuan https://orcid.org/ 0000-0001-6897-4563

References

1. Daş, G. S., F. Gzara, and T. Stützle. A Review on Airport Gate Assignment Problems: Single Versus Multi Objective Approaches. *Omega*, Vol. 92, 2020, p. 102146.
2. Kim, S. H., and E. Feron. Impact of Gate Assignment on Departure Metering. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 2, 2013, pp. 699–709.
3. Cai, X., W. Sun, M. Misir, K. C. Tan, X. Li, T. Xu, and Z. Fan. A Bi-Objective Constrained Robust Gate Assignment Problem: Formulation, Instances and Algorithm. *IEEE Transactions on Cybernetics*, Vol. 51, No. 9, 2019, pp. 4488–4500.
4. Benlic, U., E. K. Burke, and J. R. Woodward. Breakout Local Search for the Multi-Objective Gate Allocation Problem. *Computers & Operations Research*, Vol. 78, 2017, pp. 80–93.
5. Zhang, D., and D. Klabjan. Optimization for Gate Re-Assignment. *Transportation Research Part B: Methodological*, Vol. 95, 2017, pp. 260–284.
6. Pternea, M., and A. Haghani. Mathematical Models for Flight-to-Gate Reassignment with Passenger Flows: State-of-the-Art Comparative Analysis, Formulation Improvement, and a New Multidimensional Assignment Model. *Computers & Industrial Engineering*, Vol. 123, 2018, pp. 103–118.
7. Deng, W., H. Zhao, X. Yang, J. Xiong, M. Sun, and B. Li. Study on an Improved Adaptive PSO Algorithm for Solving Multi-Objective Gate Assignment. *Applied Soft Computing*, Vol. 59, 2017, pp. 288–302.
8. L'Ortye, J., M. Mitici, and H. Visser. Robust Flight-to-Gate Assignment with Landside Capacity Constraints. *Transportation Planning and Technology*, Vol. 44, No. 4, 2021, pp. 356–377.
9. Kim, J., B. Goo, Y. Roh, C. Lee, and K. Lee. A Branch-and-Price Approach for Airport Gate Assignment Problem with Chance Constraints. *Transportation Research Part B: Methodological*, Vol. 168, 2023, pp. 1–26.
10. Liu, Y., J. Liu, L. Jiang, and X. Zhang. Summary of Research on Optimum Configuration of Airport Boarding Gate. *Proc., Seventh International Conference on Electromechanical Control Technology and Transportation (ICECTT 2022)*, Vol. 12302, SPIE, Washington, D.C., 2022, pp. 1315–1321.
11. Tang, C.-H. A Gate Reassignment Model for the Taiwan Taoyuan Airport Under Temporary Gate Shortages and Stochastic Flight Delays. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 41, No. 4, 2010, pp. 637–650.
12. Muhafız Yılıdz, M., U. Avcı, M. A. Örnek, and C. Öztürk. Flight Gate Assignment Problem with Reinforcement Learning. *Proc., International Conference on Intelligent and Fuzzy Systems*, Springer, New York, 2023, pp. 189–196.
13. Xiaoyun, S., and Y. Huiwen. Multi-Objective Airport Gate Assignment Based on Stratified Sequencing Method. *Journal of Civil Aviation University of China*, Vol. 37, No. 2, 2019, p. 43.
14. Lee, J., H. Im, K. H. Kim, S. Xi, and C. Lee. Airport Gate Assignment for Improving Terminals' Internal Gate Efficiency. *International Journal of Industrial Engineering*, Vol. 23, No. 6, 2016, pp. 431–444.
15. Yu, C., and H. Y. Lau. Airport Gate Reassignment Based on the Optimization of Transfer Passenger Connections. *Proc., 4th International Conference on Traffic and Logistic Engineering*, Orlando, 2015.
16. Morrison, D. R., S. H. Jacobson, J. J. Sauppe, and E. C. Sewell. Branch-and-Bound Algorithms: A Survey of Recent Advances in Searching, Branching, and Pruning. *Discrete Optimization*, Vol. 19, 2016, pp. 79–102.
17. Zhan, W., C. Luo, J. Wang, C. Wang, G. Min, H. Duan, and Q. Zhu. Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing. *IEEE Internet of Things Journal*, Vol. 7, No. 6, 2020, pp. 5449–5465.
18. Wei, Y., F. R. Yu, M. Song, and Z. Han. Joint Optimization of Caching, Computing, and Radio Resources for Fog-Enabled Iot Using Natural Actor–Critic Deep Reinforcement Learning. *IEEE Internet of Things Journal*, Vol. 6, No. 2, 2018, pp. 2061–2073.
19. Chen, C., M. Cui, F. Li, S. Yin, and X. Wang. Model-Free Emergency Frequency Control Based on Reinforcement Learning. *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 4, 2020, pp. 2336–2346.
20. Yang, J., J. Zhang, and H. Wang. Urban Traffic Control in Software Defined Internet of Things Via a Multi-Agent Deep Reinforcement Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 6, 2020, pp. 3742–3754.
21. Wei, Q., L. Wang, Y. Liu, and M. M. Polycarpou. Optimal Elevator Group Control Via Deep Asynchronous Actor–Critic Learning. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 12, 2020, pp. 5245–5256.
22. Nguyen, N. D., T. T. Nguyen, P. Vamplew, R. Dazeley, and S. Nahavandi. A Prioritized Objective Actor-Critic Method for Deep Reinforcement Learning. *Neural Computing and Applications*, Vol. 33, 2021, pp. 10335–10349.
23. Liang, B., Y. Li, J. Bi, C. Ding, and X. Zhao. An Improved Adaptive Parallel Genetic Algorithm for the Airport Gate Assignment Problem. *Journal of Advanced Transportation*, Vol. 2020, 2020, pp. 1–17.
24. Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, et al. Human-Level Control Through Deep Reinforcement Learning. *Nature*, Vol. 518, No. 7540, 2015, pp. 529–533.