An adaptive heuristic algorithm based on reinforcement learning for ship scheduling optimization

Runfo Li [a], **Xinyu Zhang** [a,b,*], **Lingling Jiang** [c], **Wenqiang Guo** [a]

[a] *Maritime Intelligent Transportation Research Team, Navigation College, Dalian Maritime University, Dalian 116026, China*

[b] *Shenzhen Research Institute of Dalian Maritime University, Shenzhen 116026, China*

[c] *College of Environmental Sciences and Engineering, Dalian Maritime University, Dalian, 116026, China*

Abstract: Due to the development of ship size and the traffic increase in port, ships having long turnaround time in port often result in port congestion, which seriously affects the efficiency and environmental sustainability of ship navigation. It has been evident that effective ship scheduling presents a solution of fundamental and strategic importance to port congestion. In this paper, a mixed-integer linear programming mathematical model is developed to realise the optimization of ship scheduling in port to minimize the total time spent by ships in port. Its methodological novelty is gained by an innovative adaptive genetic simulated annealing algorithm based on a reinforcement learning algorithm (GSAA-RL) to support the developed mathematical model, in which the genetic algorithm is considered as the basic optimization algorithm, and Q-learning with a unique property of selecting suitable parameters dynamically is developed to adjust the parameters of crossover and mutation to improve the search ability of the algorithm. Meanwhile, the dynamic parameter turning process is formulated into a Markov decision process (MDP) model with well defining the state, action, and reward function in GSAA-RL. Specifically, the state sets are proposed by analyzing the key factors affecting the scheduling efficiency and a new reward mechanism that can reduce the objective value significantly based on the quality of selected parameters is designed. The annealing operation is performed on some excellent individuals to further expand the search scope. Simulation experiments demonstrate that the proposed GSAA-RL algorithm can significantly shorten the total time spent by ships in port compared to existing approaches. The findings hence make contributions to ship owners for their improved operation efficiency and to port operators/authorities for the reduction of port congestions.

Keywords: Q-learning; Adaptive genetic simulated annealing algorithm; Ship traffic scheduling; Maritime transportation.

1. Introduction

Since the new century, we have witnessed the fast growth of the maritime industry to respond to the tremendous development of global trade. The volume of world maritime trade increased from 8.4 billion tons in 2010 to 10.6 billion tons in 2020 [1]. As an essential part of maritime transportation, ports undertake the vital mission of radiating and driving the economic development of coastal areas, hence receiving widespread attention from academia and industry. However, the rapid increases of both ship size and traffic in ports have posed ship delays and traffic congestion in port, leading to a significant increase in operating costs and a serious decline in service quality. It is extremely costly to alter a port layout and infrastructure after its initial establishment, rationally organizing and dispatching the inbound and outbound ships is deemed to be a realistic and effective solution to alleviation of traffic congestion and ships delay [2].

According to the surveys of port, it is found that the current ships entering and leaving the port

[*] Corresponding author.
E-mail addresses: 15352856102@163.com (Runfo Li), zhangxy@dlmu.edu.cn (Xinyu Zhang).

are mainly scheduled by VTS staff based on human experience under the guidance of port navigation rules. Obviously, this method takes more human factors into consideration and lacks a certain theoretical guidance. It is difficult to ensure the efficiency and rationality of scheduling, which further leads to the unnecessary waiting time of ships entering and leaving the port. Therefore, for port managers and ship owners, it is urgent to shift the current scheduling paradigm towards a new and efficient scheduling method which enables the optimal scheduling even under a complex traffic circumstance.

In light of this practical demand, many scholars have proposed different types of methods to study the complex problem of ship scheduling. Table 1 presents some classical methods in the relevant area. Some scholars have proposed precise approaches to provide optimal solutions. Specifically, it includes a branch and bound algorithm [3], a column generation algorithm [2,4], and a lagrangian relaxation algorithm [5]. However, the precise algorithms take long computational time, and it is difficult to obtain a global optimal solution even for small-scale calculation examples. Therefore, heuristic methods are proposed to solve the problem, including a simulated annealing multi-population genetic algorithm [6], a simulated annealing algorithm [7,8], and a hybrid algorithm combining heuristic rules and simulated annealing algorithm [9], a meta-heuristic algorithm [10,11,12], a non-dominated sorting genetic algorithm [13,14], a genetic algorithm [16,17,18], a tabu search algorithm [19], a large neighborhood search algorithm [15]. Although showing much attractiveness in ship scheduling optimization, previous studies still revealed some concerns of which the theoretical implications have yet been well addressed in the existing literature. Among the significant ones is the determination of the optimization algorithm parameters which is crucial in terms of the improvement of solution quality.

With the rapid development of machine learning technologies in recent years, a few scholars have put effort on reinforcement learning to adaptively adjust the parameters of heuristic algorithms. Compared with the limitations of heuristic algorithms that directly give specific parameters in a specified solution space, reinforcement learning has the advantage of being able to select suitable parameters by revealing the internal structure of the population [20,21]. Shahrabi et al. [22] proposed to use a Q-learning algorithm in reinforcement learning to improve the performance of the variable neighborhood search algorithm for the dynamic workshop scheduling problem of machine failure. In order to obtain a feasible shop scheduling sequence in a limited period, Cao et al. [23] proposed a knowledge-based cuckoo search algorithm, which incorporated the Sarsa algorithm into the cuckoo algorithm to effectively improve the cuckoo algorithm's performance. Pettinger et al. [24] proposed a hybrid system for the traveling salesman problem. The system used Q learning algorithm to estimate the state-action value, which is used to realize the advanced adaptive control of the genetic algorithm. Based on the reserve selection mechanism, Chen et al. [25] proposed an optimal reserve scale learning method based on reinforcement learning technologies and conducted experimental verification. The verification results proved that the proposed algorithm is effective in finding the optimal reserve scale. Meng et al. [26] proposed an improved reinforcement learning-based dynamic priority algorithm for parameter optimization to solve the problem of selecting scheduling performance index in a dynamic priority algorithm. The experimental results demonstrated that the improved reinforcement learning algorithm can not only optimize the weight parameters but also reduce the deadline error rate.

Although the methods of adjusting and optimizing heuristic algorithm parameters by reinforcement learning have been applied in the fields of job shop scheduling, product manufacturing, and power systems, there is little evidence to the authors' best knowledge that that the reinforcement learning has been used to optimise heuristic algorithm parameters in maritime transport and less in ship scheduling for the purpose of shortening the total time of ships for entering and leaving the port The key to using reinforcement learning to adjust the parameters of the heuristic algorithm lies in the setting of the Markov decision process (MDP) adjustment parameter model (e.g. state sets, action sets andreward functions). The MDP model constructed in other fields is obviously not appliable in the shipping field, due to its unique characteristics.

In this paper, an adaptive genetic simulated annealing algorithm based on a reinforcement learning algorithm (GSAA-RL) is designed. In GSAA-RL, the dynamic parameter turning process is formulated into a MDP model with well defining the state, action, and reward function. Specifically, by analyzing the key factors affecting ship scheduling, a state set that conforms to the realistic situation of ships entering and leaving the port is divided. A new reward mechanism that can significantly reduce the objective value based on the quality of selected parameters is developed to improve the learning efficiency of Q-learning, and reduce the number of iterations of the GSAA-RL algorithm. In addition, the annealing operation is performed on some excellent individuals to further expand the search scope. The proposed solution algorithm is evaluated and benchmarked against the First Come First Service (FCFS) strategy, CPLEX solver, genetic algorithm (GA), and genetic simulated annealing algorithm (GSAA), which have been frequently used as the norms for the ship scheduling literature.

The main contributions of this paper are summarised as follows:

First, a new GSAA-RL algorithm is proposed based on the characteristics of the specific shipping scheduling problem to solve a mixed-integer linear programming (MILP) model. Compared to the existing methods, the algorithm is new in a sense that it aids more reduction of the total time spent of ships entering and leaving the port.

Second, an MDP model with a property of turning parameter dynamically is constructed, where the state sets for the ship scheduling optimization problem are proposed through the analysis of the key factors affecting the scheduling efficiency. It is more consistent with the real situation to be modelled.

Third, a new reward mechanism is designed to effectively minimize the total time spent by ships in port based on the quality of selected parameters. It significantly improves the learning efficiency of Q-learning and reduces the number of GSAA-RL algorithm iterations.

The remainder of this paper is organized as follows. Section 2 describes the problem statement and modeling. Section 3 presents the details for the implementation of the GSAA-RL algorithm. Computational experiments are conducted in Section 4. Section 5 concludes the paper and provides the possible directions for future studies.

Table 1

Summary of methods for solving ship traffic scheduling problems.

AUTHORS	OM		
	EM	HM	RL
Jia et al. 2019	CG		
Liu et al. 2021	CG		
Li et al. 2019	LR		
Wu et al. 2021	BB		
Lalla et al. 2016		SA	
Pei et al. 2018		SA	
Zhang et al. 2016		GA+SA	
Zheng et al. 2018		SA+HR	

Meisel et al. 2019	MHM	
Andersen et al. 2021	MHM	
Soroush et al. 2020	MHM	
Zhang et al. 2020	GA	
Liu et al. 2021	GA	
Zhang et al. 2018	NSGA-II	
Zhang et al. 2019	NSGA-II	
Li et al. 2021	NSGA-II+TS	
Liu et al. 2021	LNSA	
This study	GA+SA	RL

OM: optimization method; EM: exact method; HM: heuristic method; RL: reinforcement learning; CG: column generation algorithm; BB: branch and bound algorithm; LR: lagrangian relaxation algorithm; SA: simulated annealing algorithm; GA: genetic algorithm; HR: heuristic rule; MHM: meta heuristic method; TS: tabu search; LNSA: large neighborhood search algorithm; NSGA-II: non-dominated sorting genetic algorithm II.

2. Problem statement and modeling

Ship operation in port is a complex process, involving a safe and feasible scheduling plan to anticipate and avoid ships facing urgent situations in the nearby waters (e.g. channels) in advance. The key to ship scheduling in one-way navigable ports is to reasonably arrange the sequence and time of ships entering and/or leaving the ports for ensuring the ships' safety and improving their navigation efficiency.

The ship scheduling problem in this paper can be described as follows: under the condition that the pre-docking berths of ships are known, all the ships expected to arrive and depart from the port in a fixed planning period are taken as the research objects, focusing on the avoidance of realistic constraints such as risky encounters and tidal time windows. It takes into account various ship driving service rules, while minimizing the total waiting time of all arriving ships as the optimization goal, and giving the best time for each ship to enter and leave the port. It should be noted that the basic ship data used in the ship scheduling model established in this paper are deterministic, leaving the effect of uncertainty in data against the relevant factors on ship scheduling to be separately presented.

2.1. Model assumptions

There are many complex factors that affect the ship traffic scheduling in port. The key factors are extracted with the following assumptions rationally made in this paper.

(1) For the in-wharf ships, the application time refers to the time when they apply to enter the port; For the out-wharf ships, the application time refers to the time when they apply for leaving the berth;

(2) The berths by incoming ships have been allocated in advance;

(3) The number of pilots is sufficient;

(4) During the ship scheduling process, the influence of weather, accidents, and other possible disturbances on the ship is not taken into account;

(5) All ships entering and leaving the port are in the same position away from the channel;

(6) When the ship applies for entering and leaving the port, the pilot and tugboat have been allocated and ready to use.

Table 2

The main components of the proposed mathematical model.

Sets and parameters	
V^{in}	set of all inbound ships, $i \in \{1, 2, ..., \lvert V^{in} \rvert\}$
V^{out}	set of all outbound ships, $i \in \{1, 2, ..., \lvert V^{out} \rvert\}$
V	set of all ships, $i \in \{1, 2, ..., \lvert V \rvert\}, V = V^{in} \cup V^{out}$
B	set of all berths, $i \in \{1, 2, ..., \lvert B \rvert\}$
d_{1i}	the distance for ship i from the entry of the channel to its anchor, $i \in V$
d_2	the distance from the entrance to the exit of the channel
d_{3i}	the distance for ship i from the exit of the channel to its berth, $i \in V$
v_i	the speed of ship i, $i \in V$
M	a sufficiently large positive integer
Decision Variables	
t_{si}	the starting time of ship i needs to enter or leave the port by the tide, $i \in V$
t_{ei}	the ending time of ship i needs to enter or leave the port by the tide, $i \in V$
$t_i^{a\_in}$	the application time of ship i, $i \in V^{in}$
$t_i^{a'\_in}$	the adjusted application time of ship i, $i \in V^{in}$
$t_{ib}^{s\_in}$	the beginning scheduling time of ship i which will berth in berth b when entering the port, $i \in V^{in}$, $b \in B$
$t_{ib}^{1\_in}$	the time when the ship i allocated to berth b approaches the channel, $i \in V^{in}$, $b \in B$
$t_{ib}^{2\_in}$	the time when the ship i associated to berth b leaves the channel, $i \in V^{in}$, $b \in B$
$t_{ib}^{f\_in}$	the time when the ship $i \in V^{in}$ finishes scheduling, which is the time of arriving at berth b, $i \in V^{in}$, $b \in B$
$t_i^{a\_out}$	the application time of ship i, $i \in V^{out}$
$t_i^{a'\_out}$	the adjusted application time of ship i, $i \in V^{out}$
$t_{ib}^{s\_out}$	the beginning scheduling time when ship i in berth b leaves the port, $i \in V^{out}$, $b \in B$
$t_{ib}^{1\_out}$	the time when the ship i allocated in berth b approaches the channel, $i \in V^{out}$, $b \in B$
$t_{ib}^{f\_out}$	the time when the ship i allocated in berth b finished scheduling, $i \in V^{out}$, $b \in B$
g_1	a minimum safe time interval is required for the ship to avoid an overtaking situation
g_2	a minimum safe time interval is required for the ship to avoid a cross-encounter or confrontation situation
T_i	binary, equal to 1 if ship i needs to take tide to the port, and 0 otherwise, $i \in V$
R_{ij}	binary, equal to 1 if ship j is scheduled after ship i when entering, $i, j \in V^{in}$
Z_{ij}	binary, equal to 1 if ship j is scheduled after ship i when leaving, $i, j \in V^{out}$
IO_i	binary, equal to 1 if ship i is entering the port, and 0 otherwise, $i \in V$

2.2. Model building

According to the sets, parameters, and decision variables shown in Table 2 and the above assumptions, the one-way ship traffic scheduling problem is modeled as an MILP model as follows.

Minimize

$$\text{OS} = \sum_{i\in V^{in}} \sum_{b\in B} ({t_{ib}}^{s\_in} - {t_i}^{a\_in}) + \sum_{i\in V^{out}} \sum_{b\in B} ({t_{ib}}^{s\_out} - {t_{ib}}^{a\_out}) \quad (1)$$

Subject to

$$t_i^{1\_in} = t_i^{s\_in} + \frac{d_{1i}}{v_i}, \forall i \in V^{in} \quad (2)$$

$$t_i^{2\_in} = t_i^{1\_in} + \frac{d_2}{v_i}, \forall i \in V^{in} \quad (3)$$

$$t_i^{f\_in} = t_i^{2\_in} + \frac{d_{3i}}{v_i}, \forall i \in V^{in} \quad (4)$$

$$t_i^{1\_out} = t_i^{s\_out} + \frac{d_{3i}}{v_i}, \forall i \in V^{out} \quad (5)$$

$$t_i^{f\_out} = t_i^{1\_out} + \frac{d_2}{v_i}, \forall i \in V^{out} \quad (6)$$

$$T_i = 1 \rightarrow t_i^{a'\_in} = \begin{cases} t_{si}, t_i^{a\_in} \le t_{si} \\ t_i^{a\_in}, t_{si} \le t_i^{a\_in} \le t_{ei} \end{cases}, \forall i \in V^{in} \quad (7)$$

$$T_i = 1 \rightarrow t_i^{a'\_out} = \begin{cases} t_{si}, t_i^{a\_out} \le t_{si} \\ t_i^{a\_out}, t_{si} \le t_i^{a\_out} \le t_{ei} \end{cases}, \forall i \in V^{out} \quad (8)$$

$$t_i^{s\_in} \ge \begin{cases} t_i^{a'\_in}, T_i = 1 \\ t_i^{a\_in}, T_i = 0 \end{cases}, \forall i \in V^{in} \quad (9)$$

$$t_i^{s\_out} \ge \begin{cases} t_i^{a'\_out}, t_i = 1 \\ t_i^{a\_out}, t_i = 0 \end{cases}, \forall i \in V^{out} \quad (10)$$

$$t_{ib}^{1\_in} - t_{jb}^{1\_in} + g_1 \le M(1 - R_{ij}), \forall i, j \in V^{in}, \forall b \in B \quad (11)$$

$$t_{ib}^{1\_out} - t_{jb}^{1\_out} + g_1 \le M(1 - Z_{ij}), \forall i, j \in V^{out}, \forall b \in B \quad (12)$$

$$\text{é}\, t_{jb}^{s\_out} + \frac{d_{1j}}{v_j} \text{- } (t_{ib}^{s\_in} + \frac{d_2}{v_i}) \text{- } g_2 \text{ù}\ (IO_i \text{ - } IO_j)^2\ {}^3\ 0, i\ \text{Î}\ V^{in}, j\ \text{Î}\ V^{out}, b\ \text{Î}\ B \quad (13)$$

$$\text{é}\, t_{jb}^{s\_in} + \frac{d_{1j}}{v_j} \text{- } (t_{ib}^{s\_out} + \frac{d_2}{v_i} + \frac{d_{3i}}{v_i}) \text{- } g_2 \text{ù}\ (IO_i \text{ - } IO_j)^2\ {}^3\ 0, j\ \text{Î}\ V^{in}, i\ \text{Î}\ V^{out}, b\ \text{Î}\ B \quad (14)$$

$$IO_i \in \{0,1\}, \forall i \in V \quad (15)$$

$$T_i \in \{0,1\}, \forall i \in V \quad (16)$$

$$R_{ij}\ \text{Î}\ \{0,1\}," \ i, j\ \text{Î}\ V^{in} \quad (17)$$

$$Z_{ij}\ \text{Î}\ \{0,1\}," \ i, j\ \text{Î}\ V^{out} \quad (18)$$

The objective function (1) aims to minimize the total time spent by ships entering and leaving the port, it includes sailing time and waiting time of all ships in port. Constraints (2)-(4) state the sailing continuity of in-wharf ships. The time at which the ship sails to the entrance of the channel, the exit of the channel and the berth at which it is berthed can be determined in turn. Constraints (5)-(6) guarantee the sailing continuity of out-wharf ships. The time when the ship sails to the exit of the channel and the entrance to the channel can be determined in turn. Constraints (7)-(8) ensure that the ship taking a tide adjusts the application time to enter or leave the port. Constraints (9)-(10) state ship can begin scheduling. Constraints (11)-(12) guarantee that a certain safety clearance should be maintained between ships in the same direction and Constraints (13)-(14) ensure that ships in different directions should maintain a safe clearance. Constraints (15)-(18) specify binary variables.

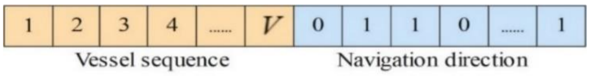
3. Adaptive genetic simulated annealing algorithm based on reinforcement learning (GSAA-RL)

3.1. Genetic simulated annealing algorithm

Because the ship traffic scheduling problem belongs to a combinatorial optimization problem, it is difficult to enumerate all the solutions by an enumeration method. Therefore, the optimization algorithm is required to have high computational efficiency. Genetic algorithm is widely used to solve combinatorial optimization problems due to their strong global searching ability and short computing time. However, the traditional genetic algorithm has the disadvantage of local convergence, while the simulated annealing algorithm has a strong local search ability [27] to further expand the search scope of solutions. A simulated annealing algorithm is first introduced in this paper to accept a worse solution with a certain probability.

3.1.1. Chromosome representation and population initialization

This process of generating a chromosome is regarded as chromosome initialization. In this paper, a chromosome includes two layers, namely ship number and navigation direction respectively. An example of a chromosome is shown in Fig. 1 and the population can be initialized according to the number of individuals *NIND*.



1	2	3	4	......	*V*	0	1	1	0	......	1

Fig. 1. An example of a chromosome

3.1.2. Genetic operations

This paper selects the objective function of the total time spent by ships entering and leaving the port as the fitness function (the first and fourth rows in Algorithm 1). According to the fitness function, the regenerated individuals are selected. The selection strategy in this paper adopts stochastic universal selection (line 9 in Algorithm 1), with the probability of *GGAP* selecting some individuals with higher fitness value from the parent as the offspring. After completing the selection operation, the selected excellent individuals are paired and crossed. A single-point crossover is used for the crossover operation (line 10 in Algorithm 1). Mutation operation plays an important role in improving the local search ability of the genetic algorithm, and it is also an important step to generate new individuals,. In the process of mutation operation, the genes of individuals in the population are randomly changed according to a certain probability. The reverse mutation is used to carry out mutation operation (line 11 in Algorithm 1).

3.1.3. Repair operation

The repair operation is mainly to repair the illegal chromosomes generated by some genetic operations. A ship *i* is about to dock at a berth *m* and another ship *i* is loading and unloading cargo on the berth *m*, the ship *i* can only be scheduled after the ship *j* has loaded and unloaded the cargo. However, after the initial chromosome has performed a series of genetic operations, the ship *i* may be scheduled earlier than the ship *j*. Therefore, it is necessary to design a repair operation to repair illegal chromosomes, that is, to readjust the order of ships entering and leaving the port. The repair method is mainly to first identify the situation where two ships entering and leaving the port are served by a berth at the same time, and then determine the scheduling order of the two ships. If ship *i* has priority over ship *j*, then the gene positions corresponding to the two ships in the chromosome are exchanged. Otherwise, it means that the chromosome is feasible and there is no need to repair it.

3.1.4. Simulated annealing operation

When a certain number of new individuals are generated after genetic and repair operations, an annealing operation is used to determine whether the new individual should replace the old one. The Monte Carlo criterion in the simulated annealing algorithm is defined as follows.

$$P=\begin{cases}1, & y_2(x)<y_1(x)\\ e^{\frac{-(y_2(x)-y_1(x))}{T}}, & y_2(x)\geq y_1(x)\end{cases} \tag{17}$$

Here, it is assumed that $y_2(x)$ is the target value of the new individual and $y_1(x)$ is the target value of the old individual. If the target value of the new individual is smaller than that of the old one, the new individual will replace the old; otherwise, the new individual is accepted with probability $e^{\frac{-(y_2(x)-y_1(x))}{T}}$. If $T>T_{end}$, the temperature is lowered through $T=q*T$ until the algorithm terminates.

3.2. MDP model construction

Q-learning is one of the most effective value-based RL algorithms. The construction of the MDP is regarded as the most important step in Q-learning [30]. The MDP can generally be described by four elements: state set S, action set A, reward function R, and strategy function π. In addition, the *Q* table is updated in Q-learning based on the comprehensive consideration of the experience state, the selected behavior, and the reward obtained by the agent, the *Q* table updated is performed by the following formula.

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha[r_t+\gamma\max_a Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \tag{18}$$

$Q(s_t,a_t)$ is the *Q* value after the agent performs a_t; r_t is the instant reward; α is learning rate; γ is discount rate; $\max_a Q(s_{t+1},a_{t+1})$ is when the agent is in the state s_{t+1}, after executing the action a_{t+1}, it expects to get the maximum *Q* value.

3.2.1. Design of state sets

In GSAA-RL, two key factors that affect the ship scheduling efficiency are selected to define the state sets. Through the comprehensive analysis of the in-wharf and out-wharf process, the number of ships (*n*) and the average total time spent by in port (AOS) are finally selected as the main factors. In order to limit the AOS value calculated in each generation to a certain range, we normalize it to eliminate the adverse effects caused by singular value. AOS' represents the average total time of entering and leaving the port normalized by the average total time spent by entering and leaving the port of the first generation population. AOS' can be calculated by the formula (19). Table 3 is the 27 state sets finally divided through multiple experiments.

$$\mathrm{AOS}'=\frac{\sum_{i=1}^{N}Fit(x_i^t)}{N}\bigg/\frac{\sum_{i=1}^{N}Fit(x_i^1)}{N} \tag{19}$$

Where $Fit(x_i^t)$ represents the fitness function of the *i*-th individual in the *t*-th iteration, $Fit(x_i^1)$

represents the fitness function of the i-th individual in the first iteration, and N is the number of individuals in the population.

Table 3

Definition of state sets.

State	ship number(n)	Average total time spent by ships in port (AOS')
s_1	$2 \le n \le 10$	$0.8 \le \text{AOS}' \le 0.85$
s_2	$2 \le n \le 10$	$0.85 \le \text{AOS}' \le 0.9$
s_3	$2 \le n \le 10$	$0.9 \le \text{AOS}' \le 0.95$
s_4	$2 \le n \le 10$	$0.95 \le \text{AOS}' \le 1.0$
s_5	$2 \le n \le 10$	$1.0 \le \text{AOS}' \le 1.05$
s_6	$2 \le n \le 10$	$1.05 \le \text{AOS}' \le 1.10$
s_7	$2 \le n \le 10$	$1.10 \le \text{AOS}' \le 1.15$
s_8	$2 \le n \le 10$	$1.15 \le \text{AOS}' \le 1.20$
s_9	$2 \le n \le 10$	$1.20 \le \text{AOS}' \le 1.25$
s_{10}	$11 \le n \le 19$	$0.8 \le \text{AOS}' \le 0.85$
s_{11}	$11 \le n \le 19$	$0.85 \le \text{AOS}' \le 0.9$
s_{12}	$11 \le n \le 19$	$0.9 \le \text{AOS}' \le 0.95$
s_{13}	$11 \le n \le 19$	$0.95 \le \text{AOS}' \le 1.0$
s_{14}	$11 \le n \le 19$	$1.0 \le \text{AOS}' \le 1.05$
s_{15}	$11 \le n \le 19$	$1.05 \le \text{AOS}' \le 1.10$
s_{16}	$11 \le n \le 19$	$1.10 \le \text{AOS}' \le 1.15$
s_{17}	$11 \le n \le 19$	$1.15 \le \text{AOS}' \le 1.20$
s_{18}	$11 \le n \le 19$	$1.20 \le \text{AOS}' \le 1.25$
s_{19}	$n \ge 20$	$0.8 \le \text{AOS}' \le 0.85$
s_{20}	$n \ge 20$	$0.85 \le \text{AOS}' \le 0.9$
s_{21}	$n \ge 20$	$0.9 \le \text{AOS}' \le 0.95$
s_{22}	$n \ge 20$	$0.95 \le \text{AOS}' \le 1.0$
s_{23}	$n \ge 20$	$1.0 \le \text{AOS}' \le 1.05$
s_{24}	$n \ge 20$	$1.05 \le \text{AOS}' \le 1.10$
s_{25}	$n \ge 20$	$1.10 \le \text{AOS}' \le 1.15$
s_{26}	$n \ge 20$	$1.15 \le \text{AOS}' \le 1.20$

s_{27}	$n \geq 20$	$1.20 \leq \text{AOS}' \leq 1.25$

3.2.2. Design of action sets and reward functions

During each iteration, the agent will choose different actions to get the appropriate crossover and mutation probabilities. For the crossover probability, we usually take the value from 0.4 to 0.9. The paper divides it into 10 intervals and the interval value is 0.05. As shown in Table 4. For the mutation probability, the value is usually from 0.01 to 0.21, which is divided into 10 intervals and the interval value is taken as 0.02, as shown in Table 5. For example, when an action a_1 is selected from action sets *Pc*, Pc is randomly selected from 0.4 to 0.46; Similarly, when an action a_2 is selected from action sets *Pm*, Pm is also randomly selected from 0.01 to 0.03. Next, the reward functions of crossover and mutation probabilities should be designed to evaluate whether the selection of their values is reasonable. Different reward functions may have different results under the same algorithm [28]. The setting of reward functions determines the convergence speed and efficiency of the algorithm. Chen et al. [29] set the following reward functions to evaluate the crossover and mutation probabilities in each iteration. They are defined as follows, respectively.

$$R_{\cos sover} = \frac{Fit_{best}(x_i^t) - Fit_{best}(x_i^{t-1})}{Fit_{best}(x_i^{t-1})} \tag{20}$$

$$R_{mutation} = \frac{\sum_{i=1}^{N} Fit(x_i^t) - \sum_{i=1}^{N} Fit(x_i^{t-1})}{\sum_{i=1}^{N} Fit(x_i^{t-1})} \tag{21}$$

Here, $Fit_{best}(x_i^t)$ represents the minimum fitness value of the *i*-th individual in the *t*-th generation, and $Fit_{best}(x_i^1)$ represents the minimum fitness value of the *i*-th individual in the first generation. But in the process of finding the target state, we must not only consider the situation of $Fit_{best}(x_i^t) < Fit_{best}(x_i^{t-1})$ and $\sum_{i=1}^{N} Fit(x_i^t)/N < \sum_{i=1}^{N} Fit(x_i^{t-1})/N$, but for the situation $Fit_{best}(x_i^t) \geq Fit_{best}(x_i^{t-1})$ and $\sum_{i=1}^{N} Fit(x_i^t)/N \geq \sum_{i=1}^{N} Fit(x_i^{t-1})/N$ that appears in each iteration, the agent must be punished and given a negative reward value to force its state to change towards a good trend. Therefore, the following improved segmented reward functions are generated in the paper.

$$R_{\cos sover} = \begin{cases} 1, & Fit_{best}(x_i^t) < Fit_{best}(x_i^{t-1}) \\ -1, & Fit_{best}(x_i^t) \geq Fit_{best}(x_i^{t-1}) \end{cases} \tag{22}$$

$$R_{mutation} = \begin{cases} 1, & \dfrac{\sum_{i=1}^{N} Fit(x_i^t) - \sum_{i=1}^{N} Fit(x_i^{t-1})}{\sum_{i=1}^{N} Fit(x_i^{t-1})} < 0 \\ -1, & \dfrac{\sum_{i=1}^{N} Fit(x_i^t) - \sum_{i=1}^{N} Fit(x_i^{t-1})}{\sum_{i=1}^{N} Fit(x_i^{t-1})} > 0 \end{cases} \tag{23}$$

Table 4

Definition of action sets Pc.

Action	Range of parameter Pc
a_1	$0.4 \leq Pc \leq 0.45$

a_2	$0.45 \le Pc \le 0.50$
a_3	$0.50 \le Pc \le 0.55$
a_4	$0.55 \le Pc \le 0.60$
a_5	$0.60 \le Pc \le 0.65$
a_6	$0.65 \le Pc \le 0.70$
a_7	$0.70 \le Pc \le 0.75$
a_8	$0.75 \le Pc \le 0.80$
a_9	$0.80 \le Pc \le 0.85$
a_{10}	$0.85 \le Pc \le 0.90$

Table 5

Definition of action sets Pm .

Action	Range of parameter Pm
a_1	$0.01 \le Pm \le 0.03$
a_2	$0.03 \le Pm \le 0.05$
a_3	$0.05 \le Pm \le 0.07$
a_4	$0.07 \le Pm \le 0.09$
a_5	$0.09 \le Pm \le 0.11$
a_6	$0.11 \le Pm \le 0.13$
a_7	$0.13 \le Pm \le 0.15$
a_8	$0.15 \le Pm \le 0.17$
a_9	$0.17 \le Pm \le 0.19$
a_{10}	$0.19 \le Pm \le 0.21$

3.2.3. Action selection strategy

This paper selects the $\varepsilon - greedy$ strategy of reinforcement learning as the action selection strategy. The strategy balances utilization and exploration, where the largest selected action value function is used, and other non-optimal actions still have a probability of being searched. The $\varepsilon - greedy$ strategy can be expressed by formula (24), where ε is the greedy rate and r is a random number between 0 and 1. When $\varepsilon \ge r$, the probabilities of crossover and mutation that maximizes Q value are chosen, when $\varepsilon \le r$, probabilities of crossover and mutation at random are chosen.

$$\pi(s_t, a_t) = \begin{cases} \max_a Q(s_t, a_t), \varepsilon \ge r \\ random\ \ value, \varepsilon < r \end{cases} \tag{24}$$

3.3. Design of GSAA-RL algorithm

This section describes the process of the Q-learning algorithm dynamically adjusting the crossover probability Pc and mutation probability Pm in detail.

Crossover and mutation operations play an extremely important role in the genetic algorithm, and the key parameters of crossover and mutation operations are Pc and Pm. In the iterative process of the algorithm, if Pc and Pm are too large, the solution will converge too slowly and if the values are too small, it will be difficult to generate new individuals [32]. However, reinforcement learning has the advantage of being able to select the suitable parameters dynamically. Therefore, reinforcement learning is introduced to adjust Pc and Pm, so that the solution effect can better meet the actual situation.

It can be divided into four steps by reinforcement learning to adjust two main parameters in the genetic algorithm. First, the agent obtains the state s_t of the time step t in the iterative process of the genetic simulated annealing algorithm (line 5 in algorithm 1). Secondly, it performs the corresponding action a_t according to the specified action selection strategy (line 6 in algorithm 1). Then, it is followed by the genetic, repair, and simulated annealing operations (lines 9-13 in algorithm 1). At this time, the state of the genetic simulated annealing algorithm is shifted to s_{t+1}, and the feedback is returned to the agent. Finally, the agent will conduct the action a_{t+1}. The agent records the learning process according to the existing state, action, the feedback received and updates the Q table at the same time (line 8 in algorithm 1). If the reward is positive, the action selection of the genetic simulated annealing algorithm will be strengthened; if the reward is negative, it will be weakened accordingly [31]. The process of continuously acquiring states, executing actions, obtaining feedback, and adjusting strategies constitutes of the reinforcement process.

After several iterations, the reinforcement learning process is activated, and the selection of Pc and Pm will be optimized based on the past and current learning experience.

Based on the above series of descriptions, the complete pseudo code of the GSAA-RL is shown in **Algorithm 1**, and the flow chart of GSAA-RL algorithm is presented in Fig. 2.

Algorithm 1. GSAA-RL

GSAA-RL parameters: population size (*NIND*), maximum iteration number (*MAXGEN*), probability of selection operation (*GGAP*), *Q* table, state set (*S*), action set (*A*).

```
Input: $Pop_{init}(t = 0)$
1: $F_t \leftarrow$ fitness value calculation for the current time step
2: $a_t, s_t \leftarrow$ choose action and state randomly
3: While $t<MAXGEN$ do
4:     $F_{t+1} \leftarrow$ fitness value calculation
5:     $s_{t+1} \leftarrow$ calculate the state of GSAA-RL according to Eq.(19)
6:     $a_{t+1} \leftarrow$ choose action according to $\varepsilon - greedy$
7:     $r_{t+1} \leftarrow$ calculate reward value for the next time step according to Eqs.(22-23)
8:     $Q(s_{t+1}, a_{t+1}) \leftarrow$ update $Q$ table according to Eq.(18)
9:     $Pop_{t+1} \leftarrow$ selection operation
10:    $Pop_{t+1} \leftarrow$ crossover operation
11:    $Pop_{t+1} \leftarrow$ mutation operation
12:    $Pop_{t+1} \leftarrow$ repair operation
13:    $Pop_{t+1} \leftarrow$ simulated annealing operation
14:    $t \leftarrow t+1$
```

```
End while
Output: Optimal schedule sequence
```

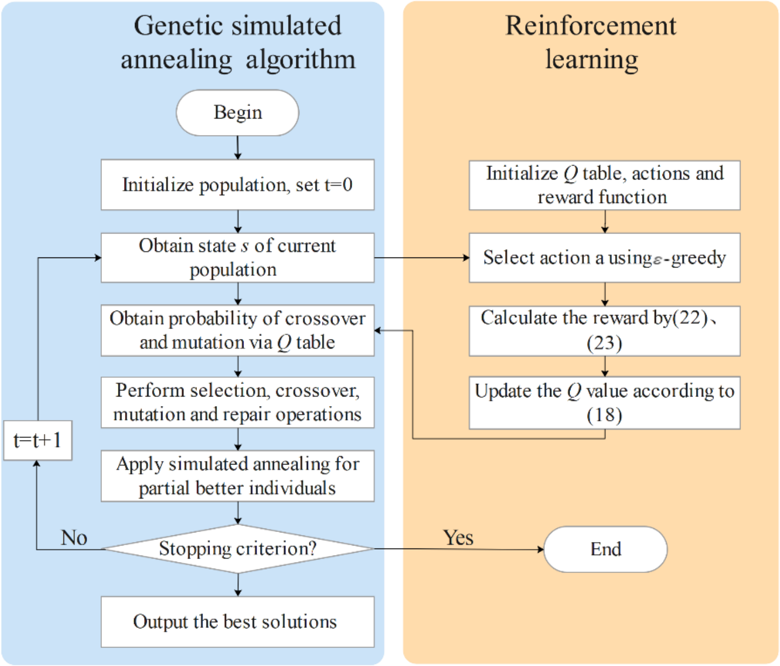

Genetic simulated annealing algorithm
Reinforcement learning
Begin
Initialize population, set t=0
Obtain state s of current population
Obtain probability of crossover and mutation via Q table
Perform selection, crossover, mutation and repair operations
Apply simulated annealing for partial better individuals
t=t+1
No
Stopping criterion?
Yes
Output the best solutions
Initialize Q table, actions and reward function
Select action a usingε-greedy
Calculate the reward by(22)、(23)
Update the Q value according to (18)
End

Fig. 2. Flow chart of the GSAA-RL algorithm

4. Computational experiments based on real cases

In this section, computational experiments are given to validate the effectiveness of the proposed model and solution algorithm using the field survey data of Comprehensive Port in Huanghua. Fig. 3. is the sketch of Comprehensive Port in Huanghua, with a 200,000 ton one-way narrow and long channel with a total length of 31 nautical miles. In particular, because of the limit of channel depth, ships with a draft of more than 18m and a length of more than 280m must take the tide to enter the port. Due to a large number of ore importers and small batches, in addition to 200,000 ton ships entering and leaving the port every day, 50,000-100,000 ton ships also occupy a certain proportion, which will cause port detention to a certain extent. Therefore, it is significantly important to find a feasible dispatching plan for ships entering and leaving the port to alleviate the congestion phenomenon. There are 8 anchorages and 15 berths in the Comprehensive Port in Huanghua. For the convenience of calculation, the anchorage numbers are marked as 1-8, and the berth numbers are marked as 1-15. The detailed information for the berth, anchorage of Comprehensive Port in Huanghua is presented in Appendix A. The scope of the numerical experiments also included a detailed evaluation of the convergence patterns and boxplot for the considered solution algorithms, which can be found in Appendix B.

There are two groups of instances used in this paper. The first group with different number of ships (6-35 ships for small-scale instances as well as 42-83 ships for large-scale instances) are considered to examine the efficiency of the proposed algorithm with GA, GSAA, CPLEX, the instances of 35 ships can be expressed as V35. The second group with different number of ships (5-80 ships) is used to compare the GSAA-RL with the real port scheduling scheme (FCFS). All experiments are conducted on a computer with a CPU of 3.5 GHz, RAM size of 64 GB, and running version 12.6.

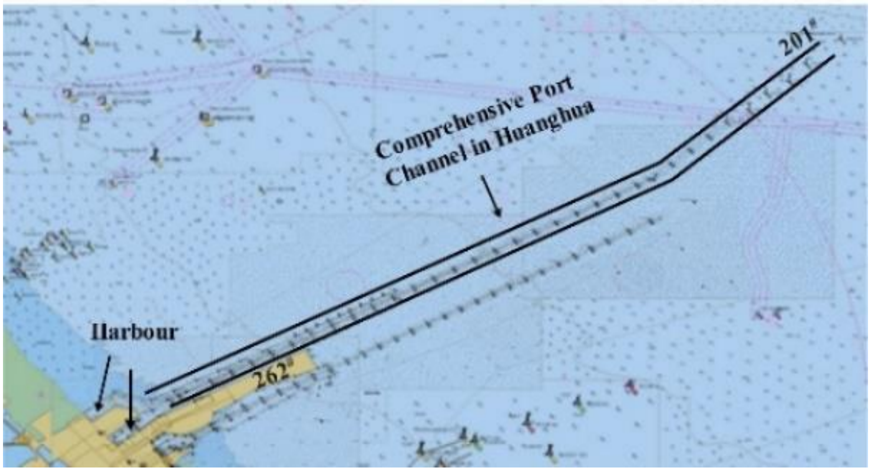


Fig. 3. The sketch of Comprehensive Port in Huanghua

4.2 Benchmark methods

To evaluate the performance of the proposed GSAA-RL algorithm, four benchmark methods are introduced to solve the MILP model and compared with our algorithm. The first benchmark method is the CPLEX solver, and the maximum computational time is set to the 3600s. The second benchmark method is the First-Come-First-Served (FCFS) strategy simulating the decision of a VTS operator, which is commonly used in ports at present. The other two methods are heuristic algorithms, namely genetic algorithm (GA) and genetic simulated annealing algorithm (GSAA).

4.3 Parameter settings

The developed GSAA-RL algorithm will be compared with 4 state-of-the-art methods (i.e., CPLEX solver, FCFS strategy, GA algorithm, and GSAA method), which have been frequently used in ship scheduling literature. Each of the considered algorithms has a set of parameters, which remain constant during the search progress. The parameters related to the GSAA-RL algorithm are set as follows: learning rate $\alpha = 0.6$, discount rate $\gamma = 0.65$, greedy rate $\varepsilon = 0.5$, crossover and mutation probabilities are adjusted dynamically according to Q-learning, and initial Q value are all 0. The parameters related to the GA and GSAA algorithms are set as follows: the maximum number of iterations $MAXGEN = 400$, the number of population $NIND = 200$, selection probability $GGAP = 0.9$, initial temperature $T_0 = 2$, cooling factor $\alpha_1 = 0.9$, crossover probability $Pc = 0.85$, mutation probability $Pm = 0.21$. The other parameters are set as follows: same-direction safety clearance $g_1 = 5$, different-direction safety clearance $g_2 = 5$. Each instance is run for 10 times.

4.4. Experimental results and analysis of 13 ships

In this subsection, the proposed model and algorithm are tested using the 13 ships in the Comprehensive Port in Huanghua on a certain day in May 2021. The basic data of 13 ships includes the direction of the ship entering and leaving the port, ship speed, berth ID, and anchorage ID, etc., as shown in Table 6. Table 7 reports the optimal scheduling scheme of the 13 ships solved by the proposed GSAA-RL algorithm. From the results shown in Table 7, it is revealed that the sequence of ships entering and leaving the port, the navigation continuity of all ships, and the time when the ships approach the channel, leave the channel and reach the berth, respectively. At the same time, it can be observed that the first ship scheduled is inbound no.1, its scheduling start time is at 0 min, and the last ship to leave the channel is inbound no.13 at 957 min. Table 8 is the chromosome with the shortest total time spent by the 13 ships entering and leaving the port. It is also clear that the optimal scheduling scheme and navigation direction of all ships by reading this chromosome from Table 8, for example, 121 indicates that ship no.12 in the state entering the port and the ranking is 8. To further show the scheduling results of the 13 ships in Table 8, the Gantt chart is drawn in Fig. 4. From Fig. 4, we better visualize the waiting time, sailing time, sailing direction of each ship.

The evolution process of parameters *Pc* and *Pm* is presented in Fig. 5. As shown in Fig. 5, in the initial and intermediate stages of the algorithm, *Pc* and *Pm* almost change between large and small values. After the rapid convergence of two phases, *Pc* and *Pm* only evolute within a small range. It can be observed that *Pc* and *Pm* are constantly searching for optimization until the objective value reaches the minimum. The experimental results of comparing the unimproved and new reward functions are shown in Fig. 6. Reward function 2 represents the convergence result of GSAA-RL without improving the reward function. The minimum total time spent by entering and leaving the port is found around 180 generations, and it is 2615min. Reward function 1 represents the convergence result of GSAA-RL with an improved segmented reward function, it can be obtained that after around 170 generations, the minimum total time of entering and leaving the port is found and the total time is 2546min. It is obvious that new segmented reward function 1 can improve the learning efficiency of Q-learning and reduces the number of GSAA-RL algorithm iteration.

Table 6

Data of 13 ships.

NO.	IO	Berth ID	Anchorage ID	Ship length (m)	Ship width (m)	Ship speed (kn)	Application time (min)	Ship draft (m)	Tidal time window(min)
1	1	13	2	225	32	13.65	0	10.35/10.75	
2	0	13	-	225	37	11.24	90	13/13.1	
3	0	5	-	292	45	14.24	153	6.36/8.7	
4	0	4	-	240	38	7.4	185	5.17/7.12	
5	1	5	5	229	32	13.65	225	12.88/12	
6	1	2	2	150	21	9.25	275	8.7	
7	1	4	3	190	32	12.12	302	12.88/13.13	
8	0	6	-	289	45	11.6	356	13.5/13.5	
9	0	7	-	295	46	9.01	396	6.5/8.6	
10	0	15	-	157	21	7.7	425	8.3	
11	1	6	2	292	45	12.31	470	18.10/18.22	[540,750]
12	1	15	6	122	21	8.48	493	6	
13	1	7	2	325	57	12.97	560	11.38/12.5	

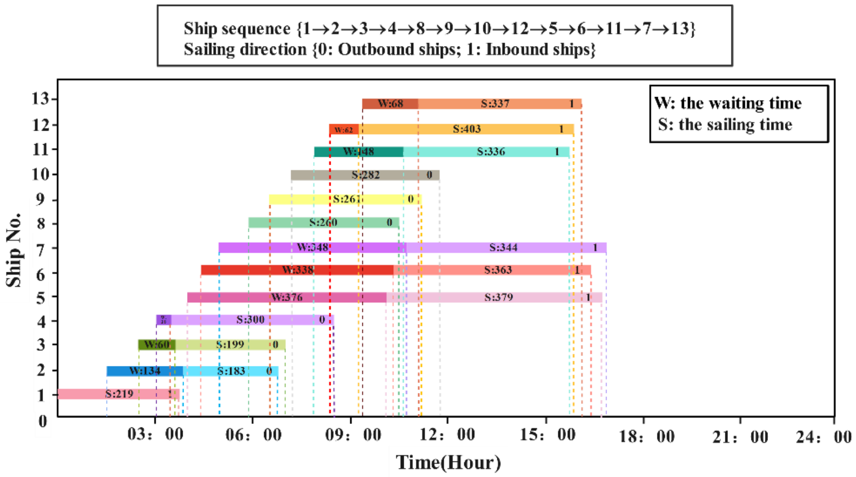


Fig. 4. Gantt diagram of the optimization result of 13 ships experiments

Table 7

Optimal results of 13 ships experiments using GSAA-RL algorithm.

No.	IO	Average speed (kn)	Adjusted average speed (kn)	Application time (min)	Begin scheduling time (min)	Approach channel time (min)	Leave channel time (min)
1	1	13.65	13.65	0	0	68	205
2	0	11.24	11.24	90	224	241	407
3	0	14.24	11.24	153	213	246	412
4	0	7.4	7.4	185	203	251	503
8	0	11.6	7.4	356	356	364	616
9	0	9.01	7.4	396	396	405	657
10	0	7.7	7.4	425	425	455	707
12	1	8.48	8.48	493	555	712	932
5	1	13.65	8.48	225	601	717	937
6	1	9.25	8.48	275	613	722	942
11	1	12.31	8.48	470	618	727	947
7	1	12.12	8.48	302	650	732	952
13	1	12.97	8.48	560	628	737	957

Table 8

Chromosome with the shortest total time spent by ships entering and leaving the port.

11	20	30	40	80	90	10	121	51	61	111	71	131

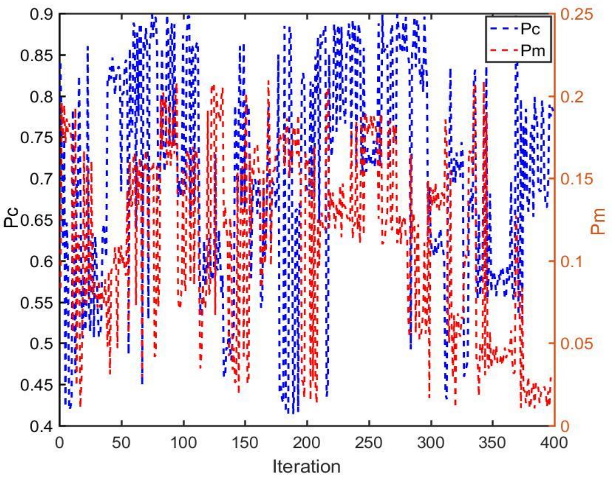


Fig. 5. The evolution process of parameter *Pc* and *Pm*.

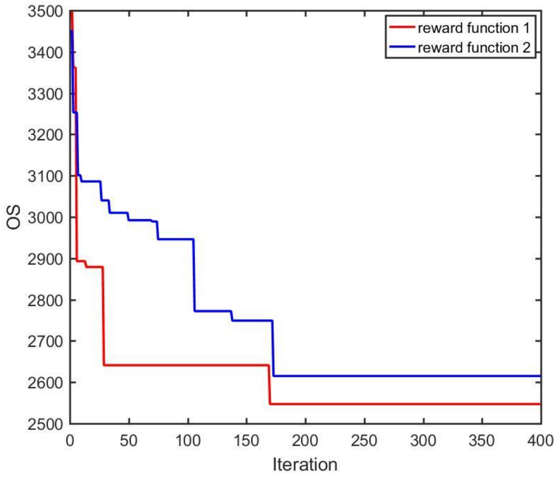


Fig. 6. Comparison of the optimization process with unimproved and of new reward functions.

4.5. The verification of the model rationality with the case of the13 ships

The rationality of the model can be verified from the perspective of constraints such as ship speed adjustment, tidal time window, safe clearance between the same and different direction ships, and navigation continuity in Table 7. Speed adjustment is to adjust the speed of ships successively in the same direction to avoid overtaking on the channel. For example, the outbound ship no.8 follows the outbound ship no.4 on the channel. The speed of ship no.8 cannot exceed that of ship no.4, which should be adjusted from 11.6kn to 7.4kn. In addition, ships can only be dispatched after they apply for entry and exit. With reference to this rule, an illustrative example is that only after the outbound ship no.4 approaches the channel, a safe clearance later, the outbound ship no.8 can approach the channel. However, the outbound ship no.8 has not applied for leaving the port at this time. Consequently, the outbound ship no.8 cannot be arranged to leave the port until the application is made at 356min. The inbound ship no.11 completes entering the port within the time window [540,750], meeting the constraint of the tidal time windows.

4.6. Effectiveness of the proposed GSAA-RL algorithm

4.6.1 Comparison of the results of the GSAA-RL with CPLEX and GSAA, GA

In this subsection, we first solve the MILP described in Section 2.2 using the proposed GSAA-RL algorithm and three benchmark methods, namely GSAA, GA, and CPLEX methods. We set the same parameters for all five methods, for example, $g_1 = 5, g_2 = 5$. We compare the results of GSAA-RL with GSAA, GA, and CPLEX in terms of the total time spent by ships entering and leaving the port, and the computational efficiency under different scale instances. All scale instances in the experiment are generated from the Comprehensive Port in Huanghua, each instance is run for 10 times. Table 9 and 10 presents the computational results of the proposed GSAA-RL, GA, GSAA, and CPLEX methods for small-scale and large-scale instances respectively, where the notation "OS_{IP}", "OS_{GA}", "OS_{GSAA}", "$\text{OS}_{\text{GSAA-RL}}$" denote the objective values of the MILP, the optimal solution generated by the GA algorithm, GSAA algorithm, and the GSAA-RL, respectively.

Table 9

Computational results for the small-scale instances.

Instance	CPLEX		GA		GSAA		GSAA-RL		Gap1(%)	Gap2(%)
	CPU(s)	OS_{IP}	CPU(s)	OS_{GA}	CPU(s)	OS_{GSAA}	CPU(s)	$\text{OS}_{\text{GSAA-RL}}$		
V6	6.72	1261	9.04	1288	25.75	1261	26.89	1261	2.10	0
V10	33.14	2038	13.37	2674	37.56	2071	39.22	2038	23.78	1.59
V13	665.12	2546	17.01	2867	49.07	2607	49.54	2546	11.20	2.34
V18	1534.23	3908	23.88	4234	56.89	4177	62.97	3722	12.09	10.89
V23	3245.21	5220	64.34	7385	82.16	6840	83.24	6649	9.97	2.79
V28	3600.00	-	85.25	13208	106.22	12054	107.25	11806	7.00	5.54
V35	3600.00	-	47.18	15442	161.27	15337	169.90	13508	12.52	11.93
Average	1812.06	-	37.01	6730	74.13	6335	77.00	5933	11.24	5.01

Notes: Gap1=($\text{OS}_{\text{GA}} - \text{OS}_{\text{GSAA-RL}}$)/$\text{OS}_{\text{GA}}$ x 100%; Gap2=($\text{OS}_{\text{GSAA}} - \text{OS}_{\text{GSAA-RL}}$)/ OS_{GSAA} x 100%;

It becomes obvious that compared with the GSAA-RL, the CPLEX is faster only when solving the instances with 6, 10, 13, 18 ships. The computational time of the CPLEX in solving NILP shows an exponential growth with the increase of the instance scale. In particular, CPLEX cannot even obtain any feasible solution for some large-scale instances with 63, 68, 73, 78, 83 ships due to the complexity of the problem. Therefore, such a solution method cannot meet the actual needs of the port. The GA and GSAA-RL algorithms can obtain a feasible solution under the one-hour time limit, but the quality of the solutions is relatively poor for the small-scale and large-scale instance. This is because the solution is obtained by fixing the values of key parameters in each iteration, thus leading to faster convergence. In contrast to these two methods, the solutions obtained by the GSAA-RL are not inferior to the solutions obtained by the GA and GSAA, and the advantage of the GSAA-RL becomes more remarkable with the increase of the instance scale. The solution obtained by GSAA-RL is 11.24% and 5.01% better than that of the GA and GSAA for small-scale instances, and 18.58% and 11.58% for large-scale instances, reflecting the superiority of the proposed algorithm. Concerning the time consumption, relatively higher computation time is recorded in GSAA-RL compared to that of GSAA, because the additional time is required to reselect the crossover and mutation probability values from the *Q* table to complete the evolution operations in each iteration in order to expand the search scope. However, the maximum GSAA-RL computational time did not exceed 15.75min (83 ships). Therefore, the port operators will be able to develop and revise ship scheduling plans in a timely manner. Based on the conducted computational experiments, the developed GSAA-RL algorithm outperformed the CPLEX solver, GA, and GSAA algorithms, demonstrating the effectiveness of the proposed algorithm.

4.6.2 Comparison of the results of the GSAA-RL and the real port scheduling schemes.

To validate the effectiveness of the model, the scheme provided by the GSAA-RL is compared to the real port scheduling schemes. According to the field surveys of Comprehensive Port in Huanghua, the real port scheduling scheme mainly adopts a FCFS rule for ship scheduling. Specifically, it arranges ships to enter and leave the port successively according to the application time of ships and preferentially assigns tide-dependent ships in the tidal time windows.

Table 11 shows the comparison among the results of the proposed scheme and the FCFS scheduling scheme. It is found that for all instances, the GSAA-RL is constantly better than the FCFS scheme, and the average gap arrives at 46.04%. It is because the order of ships has been determined in the FCFS strategy in advance, and it is difficult to arrange a reasonable scheduling order. This result indicates that the proposed model can significantly help port operators to reduce the total time spent by ships in port, thus achieving higher throughput and better emission performance.

Table 10

Computational results for the large-scale instances.

Instance	CPLEX		GA		GSAA		GSAA-RL		Gap1(%)	Gap2(%)
	CPU(s)	$\mathrm{OS}_{\mathrm{IP}}$	CPU(s)	$\mathrm{OS}_{\mathrm{GA}}$	CPU(s)	$\mathrm{OS}_{\mathrm{GSAA}}$	CPU(s)	$\mathrm{OS}_{\mathrm{GSAA\text{-}RL}}$		
V42	3600	-	62.06	25479	219.96	22176	233.41	20410	19.89	7.77
V48	3600	-	79.85	33234	272.78	32314	281.40	28454	14.38	11.95
V55	3600	-	195.25	41095	367.55	39125	377.10	36135	12.07	7.67
V63	3600	-	249.71	49283	452.32	48188	455.19	47381	3.86	1.67
V68	3600	-	305.88	68996	510.93	58417	526.71	52737	23.57	9.72
V73	3600	-	420.66	83429	660.42	80082	672.13	67574	19.00	15.62
V78	3600	-	506.07	128196	723.44	112967	735.27	85254	33.50	24.53
V83	3600	-	745.45	138458	856.23	124584	945.12	107545	22.33	13.68
Average	3600	-	320.62	71021	507.95	64731	528.29	55686	18.58	11.58

5. Conclusion and scope of future work

The rapid development of ships tonnage has caused great challenges on the capacities of ports, leading to frequent ship delays and heavy port congestion. Thus, effective ship scheduling scheme is required to cope with the phenomenon. This article presents an MILP mathematical model for the ship schedule problem with minimizing the total time spent by ships in port. The novel GSAA-RL algorithm is designed to solve the MILP model. In the proposed GSAA-RL, the genetic algorithm is considered as the basic optimization algorithm, and Q-learning with a unique property of selecting suitable parameters dynamically is developed to adjust the parameters of crossover and mutation to improve the search ability of the algorithm. In addition, given the fact that the genetic algorithm may fall into local optimum, a simulated annealing operation is implemented to some excellent individuals after genetic operations to further enhance the search ability of solution space.

To verify the effectiveness of the proposed GSAA-RL algorithm, taking the data of Comprehensive Port in Huanghua as an example, we compare the computational results of the proposed solution algorithm in this paper with those of a MILP/CPLEX solver, a FCFS strategy, a GA method, and a GSAA algorithm. Computational experiments demonstrate that the GSAA-RL algorithm proposed significantly outperforms three existing methods (i.e., the CPLEX, GA, and GSAA) in terms of solution quality when solving the small-scale and large-scale instances. In contrast to the real port scheduling schemes (i.e., FCFS strategy), the scheme obtained by the proposed GSAA-RL algorithm can reduce the total time spent by ships entering and leaving the port by an average of 43.91%. These computational performances highlight the effectiveness of the proposed solution algorithm in the practical applications. The algorithm proposed in this paper provides a new tool for ship traffic scheduling and makes up for the shortcomings of some existing scheduling optimization methods.

Table 11

Comparison of the results of the GSAA-RL and the real port scheduling schemes.

Instance	FCFS		GSAA-RL		Gap(%)
	CPU(s)	OS_{FCFS}	CPU(s)	$OS_{GSAA\text{-}RL}$	
V5	0.34	1095	25.30	1078	1.55
V9	0.35	2362	37.98	1910	19.14
V15	0.35	5490	58.80	3317	39.58
V20	0.35	9478	77.65	5784	38.97
V25	0.39	15026	100.47	8597	42.79
V30	0.38	21460	124.03	11037	48.57
V36	0.39	30846	173.18	13950	54.78
V43	0.37	44134	231.63	22906	48.10
V49	0.39	58919	295.71	28829	51.07
V54	0.36	75304	355.24	31001	58.83
V60	0.39	98214	423.92	39117	60.17
V65	0.37	119935	468.75	46345	61.36
V70	0.38	142863	548.05	55727	60.99
V75	0.36	167731	636.11	75478	55.00
V80	0.38	195503	821.45	98452	49.64
Average	0.37	65891	291.88	29569	46.04

Notes: Gap=(OS_{FCFS}-$OS_{GSAA\text{-}RL}$)/OS_{FCFS} x 100%;

Current research can be expanded from the following two aspects. First of all, from the perspective of the model, taking into account the actual situation, some input parameters (such as the ship's expected arrival time) may be affected by the weather and its own mechanical failure, which may cause the arrival time to be uncertain and affect the initial ship scheduling plan. Therefore, the ship scheduling problem under the uncertain conditions has practical significance, and this part is being studied as a key content of the authors' current work and will be presented in a separate cover in the future. Secondly, from the perspective of algorithm applications, the GA-RL algorithm can be applied to the optimization problem of ship scheduling in two-way, compound and restricted channels, in order to shorten the total waiting time of ships in port and improve the efficiency of channel navigation. Finally, methodologically, ues of reinforcement learning to adjust the key parameters of other optimization algorithms, such as tabu search algorithm, particle swarm optimization algorithm, can be further investigated to improve the search performance of the algorithm.

CRediT authorship contribution statement

Runfo Li: Conceptualization, Methodology, Data curation, Software, Writing – original draft, Writing – review & editing. **Xinyu Zhang:** Conceptualization, Supervision, Project administration, Funding acquisition, Writing – review & editing, Validation. **Lingling Jiang:** Conceptualization, Methodology, Investigation, Writing – review & editing**. Wenqiang Guo:** Conceptualization, Methodology, Writing – review & editing.

Declaration of Competing Interest

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

Acknowledgments

This work was financially supported by the Central Guidance on Local Science and Technology Development Fund of Shenzhen (grant no.2021Szvup014), and the National Natural Science Foundation of China (grant no. 51779028). The authors thank the referees for helpful comments and suggestions.

Appendix A: berth, and anchorage information for Comprehensive port in Huanghua

The length of the planning horizon in this paper is set to 2 days for small-scale instances (6-35 ships) and 3 days for large-scale instances (42-83 ships) to show the applicability of the model. We set both the safe navigation distance and the unit of the time step to 10 minutes. Table A1 records the distance between the anchorage and buoy no.201, and the distance between the berth and the buoy no.262 is depicted in Table A2.

Table A1

Distance between the anchorage and the buoy no.201.

Anchorage ID	Distance(nm)
1	15.472
2	15.396
3	11.459
4	16.381
5	16.256
6	22.111
7	17.123
8	10.39

Table A2

Distance between the berth and the buoy no.262.

Berth	Berth ID	Distance(nm)	Berth	Berth ID	Distance(nm)
HG1	1	4.691	GC2	9	1.159
HG2	2	4.798	GC3	10	2.372
HG3	3	5.617	GC4	11	3.337
HG4	4	5.845	GC5	12	2.893
HG5	5	6.002	GC6	13	3.164

K01	6	0.914	GC7	14	3.395
K02	7	1.083	GC8	15	3.622
GC1	8	1.035	-	-	-

Appendix B: convergence curve and boxplot comparison for different algorithms

The analysis of convergence patterns allows keeping track of the objective function value improvements from one generation to another (in case of the considered GA, GSAA, GSAA-RL). The convergence patterns were shown only for the some small-scale and large-small instances [V13, V18, V23, V28, V35, V42, V48, V55, V63] and presented in Fig. 7. Based on the convergence pattern analysis, it can be noticed that GSAA-RL was able to identify the promising solutions of the search space much faster as compared to the GA, and GSAA algorithms, and allows effective exploration of the search space and identification of the domains with high-quality solutions. Besides, the boxplot figure of OS value (i.e., the total time spent by ships in port) from GA, GSAA, GSAA-RL is shown in Fig. 8 for the some small-scale and large-small instances [V10, V13, V18, V23, V28, V35, V42, V48, V55], which could further validate the effectiveness of the GSAA-RL algorithm. It can be observed from Fig. 8 that the OS values obtained by GSAA-RL algorithm have smaller median and range, which further illustrates the effectiveness of the proposed GSAA-RL in this paper for solving the ship scheduling problem.

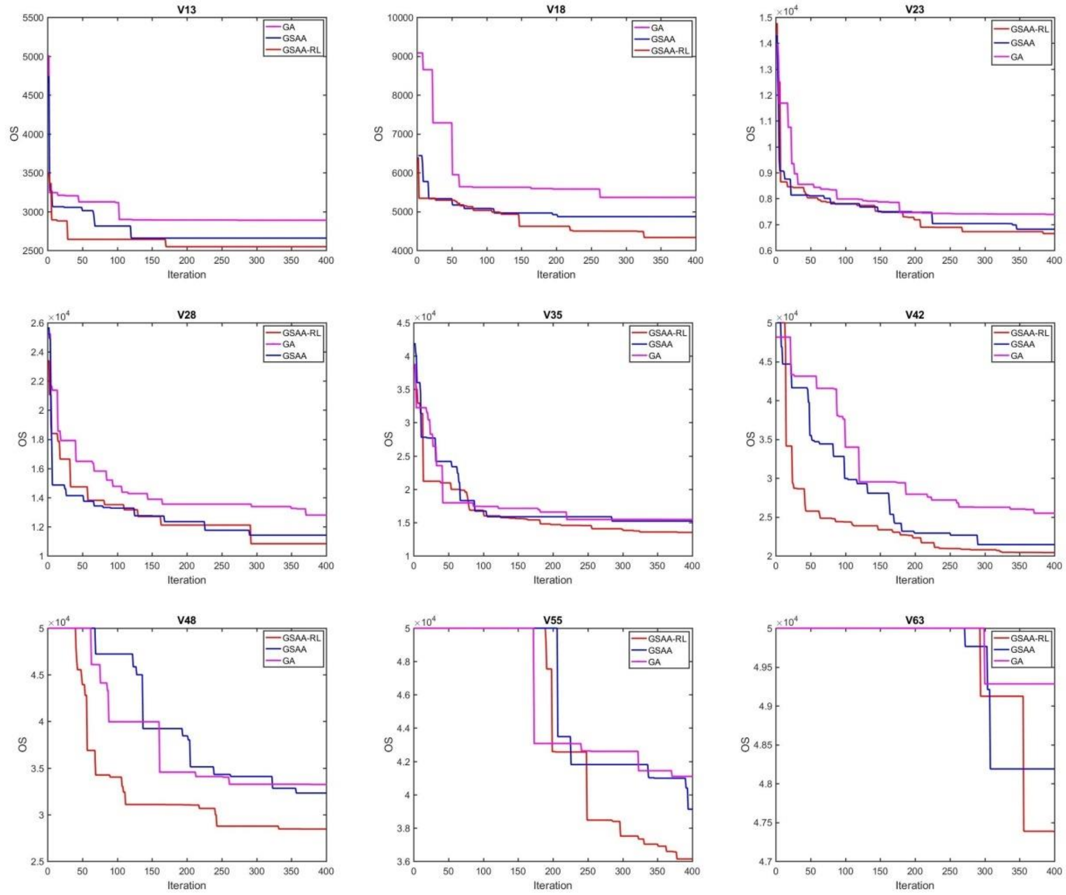


Fig. 7. The convergence curves of different algorithms for small-scale and large-scale instances [V13, V18, V23, V28, V35, V42, V48, V55, V63].

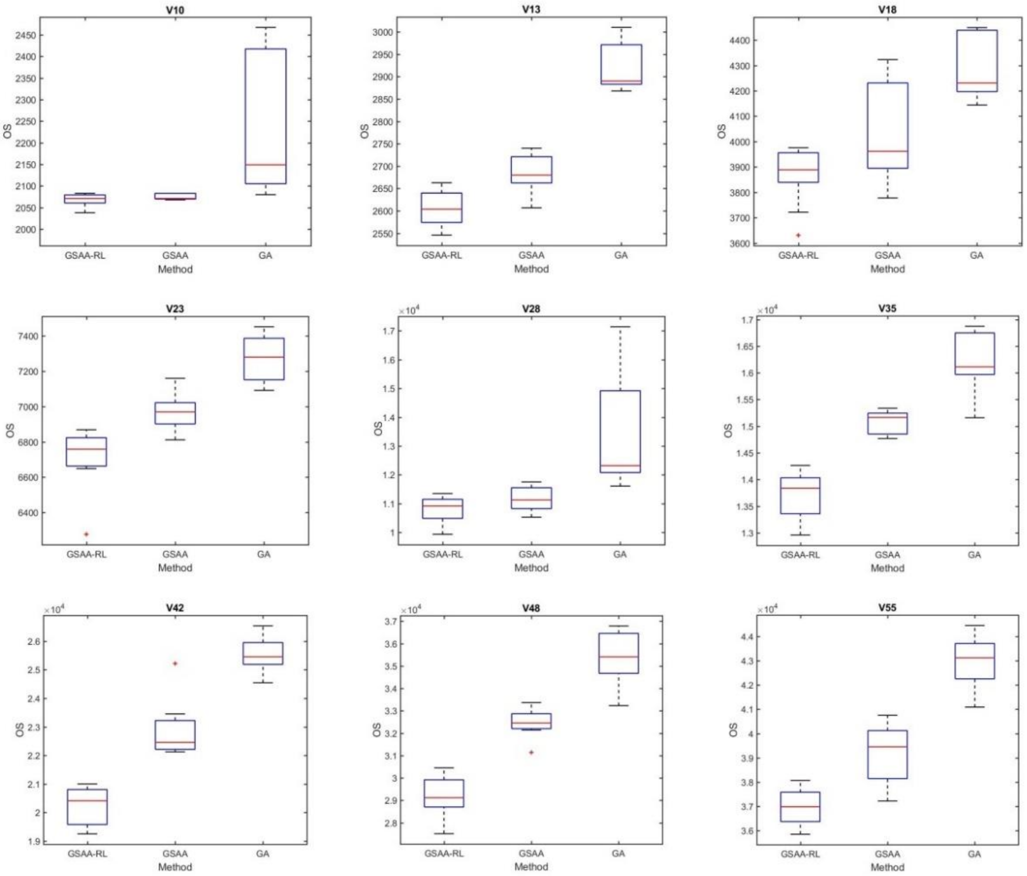


Fig. 8. The boxplot of three algorithms for for small-scale and large-scale instances [V10, V13, V18, V23, V28, V35, V42, V48, V55].

Reference

[1] UNCTAD. Review of maritime transport, 2017. United Nations Conference on Trade and Development.

[2] Jia S, Li CL, Xu Z, 2019. Managing navigation channel traffic and anchorage area utilization of a container port. J. Transportation Science. 53(3):728-745.

[3] Wu YQ, Zhang R, 2021. Integrated optimization of continuous berth allocation and ship dispatching under one-way channel. J. Computer Engineering and Applications.

[4] Liu BL, Li ZC, Wang YD, Sheng DA. 2021 Short-term berth planning and ship scheduling for a busy seaport with channel restrictions. J. Transportation Research Part E .154.

[5] Li SQ, Jia S, 2019. The seaport traffic scheduling problem: Formulations and a column-row generation algorithm. J. Transportation Research Part B. 128:158-184.

[6] Zhang XY, Lin J, Guo ZJ, Liu TS, 2016. Vessel transportation scheduling optimization based on channel-berth coordination. J. Ocean Engineering. 112:145-152.

[7] Lalla-Ruiz E, Shi XN, Voß SX, 2018. The waterway ship scheduling problem. J. Transportation Research Part D. 60:191-209.

[8] Pei YL, Chen DJ, 2018. Study on the optimal dispatching algorithm of ships in and out of tidal two-way channel. C. IEEE 4th Information Technology and Mechatronics Engineering Conference. IEEE. 324-329.

[9] Zheng HX, Liu BL, Deng CY, Feng PP, 2018. Ship scheduling optimization in one-way channel bulk harbor. J. Operations Research and Management Science. 27(12):28-37.

[10] Meisel F, Fagerholt K, 2019. Scheduling two-way ship traffic for the Kiel Canal: Model, extensions and a matheuristic. J. Computers & Operations Research. 106:119-132.

[11] Soroush FA, Reza TM, Dorsa A, Behdin VN, 2020. Simultaneous waterway scheduling, berth allocation, and quay crane assignment: A novel matheuristic approach. J. International Journal of Production Research.

[12] Andersen T, Hove JH, Fagerholt K, Meisel F, 2021. Scheduling ships with uncertain arrival times through the Kiel Canal. J. Maritime Transport Research. 2.

[13] Zhang XY, Li RJ, Lin J, Xu CB, 2018. Optimization modeling of vessel traffic scheduling for Y shaped bifurcated compound waterway. J. Journal of Dalian Maritime University. 44(2):1-14.

[14] Zhang XY, Li RJ, Lin J, Chen X, 2018. Vessel scheduling optimization in two-way traffic ports. J. Navigation of China. 41(2):36-40.

[15] Liu BL, L ZC, Sheng D, Wang YD, 2021. Integrated planning of berth allocation and vessel sequencing in a seaport with one-way navigation channel. J. Transportation Research Part B. 143:23-47.

[16] Zhang B, Zheng ZY, Wang DQ, 2020. A model and algorithm for vessel scheduling through a two-way tidal channel. J. Maritime Policy & Management. 47(2):188-202.

[17] Liu DD, Shi GY, Hirayama K, 2021. Vessel scheduling optimization model based on variable speed in a seaport with one-way navigation channel. J. Sensors. 21(16):5478.

[18] Zhang XY, Chen X, Ji MJ, Yao S, 2017. Vessel scheduling model of a one-way port channel. J. Journal of Waterway Port Coastal & Ocean Engineering. 143(5):04017009.

[19] Li JJ, Zhang XY, Yang BD, Wang NN, 2020. Vessel traffic scheduling optimization for restricted channel in ports. J. Computers & Industrial Engineering. 152(3):107014.

[20] Zhao J, Liu SX, Zhou MC, Guo XW, Qi L, 2018. Modified cuckoo search algorithm to solve economic power dispatch optimization problems. J. IEEE-CAA Journal of Automatic Sinica. 5(4):794-806.

[21] Zhang YW, Wang L, Wu QD, 2014. Dynamic adaptation cuckoo search algorithm. J. Control Decision. 29(4):617-622.

[22] Shahrabi J, Adibi MA, Masoud M, 2017. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. J. Computers & Industrial Engineering. 110:75-82.

[23] Cao ZC, Lin CR, Zhou MC, 2021. A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. J. IEEE Transactions on Automation Science and Engineering. 18(1):56-69.

[24] Pettinger JE, Everson RM, 2002. Controlling genetic algorithms with reinforcement learning. C. Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation.

[25] Chen Y, Hu JL, Hirasawa K, Yu SN, 2007. Optimizing reserve size in genetic algorithms with reserve selection using reinforcement learning. C. Proceedings of Sice Annual Conference. 1-8.

[26] Meng SS, Zhu Q, Xia F, Lu JF, 2020. Research on parameter optimization of dynamic priority scheduling algorithm based on improved reinforcement learning. J. IET Generation, Transmission & Distribution. 14(16):3171-3178.

[27] Fang YL, Ming H, Li MQ, Liu Q, Pham DT, 2019. Multi-objective evolutionary simulated annealing optimization for mixed-model multi-robotic disassembly line balancing with interval processing time. J. International Journal of Production Research. 58(3):846-862.

[28] Xue BP, Berseth G, Michiel VDP, 2016. Terrain-adaptive locomotion skills using deep reinforcement learning. J. Acm Transactions on Graphics. 35(4):1-12.

[29] Chen RH, Yang B, Li S, Wang SL, 2020. A Self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. J. Computers & Industrial Engineering. 149.

[30] Bellman R, 1957. Dynamic programming and lagrange multipliers. C. Proceedings of the National Academy of Sciences of the United States of America. 42:767-769.

[31] Emary E, Zawbaa HM, Grosan C, 2018. Experienced gray wolf optimization through reinforcement learning and neural networks. J. IEEE Transactions on Neural Networks and Learning Systems. 29:681-694.

[32] Bashir MB, Nadeem A, 2017. Improved genetic algorithm to reduce mutation testing cost. J. IEEE Access. 5:3657-3674.