

Solving Multi-robot Task Assignment Problem based on Improved Genetic Algorithm

Zhibin Wang, Jianjun Zhu

School of Jilin Institute of Chemical Technology, Jilin, Jilin, 132022, China

Abstract: Aiming at the low efficiency of robot task distribution and unbalanced task distribution in the process of warehouse task scheduling, this paper aims at minimum total task path and balanced task distribution, establishes a warehouse robot scheduling scheme and mathematical model, and proposes an improved genetic algorithm for task allocation. Firstly, greedy algorithm is introduced to optimize the initial population and improve the quality of the initial population. Secondly, the heuristic bidirectional triple crossover operator is designed to expand the search range of the population and improve the accuracy of the optimal solution. Finally, the multi-variant strategy is introduced to solve the problem that the algorithm falls into local optimality. The experimental results show that the improved algorithm has better convergence and balanced distribution ability.

Keywords: Genetic Algorithm; Multi-Robot; Task Allocation; Three Crossover Operators; Variation Strategy.

1. Introduction

With the improvement of people's living standards and the innovation of information technology, e-commerce[1]has developed rapidly, which in turn has driven the transformation of the logistics industry and provided huge development space for the warehousing industry. The traditional model of people arriving at the goods is inefficient, costly and error-prone, and can no longer meet the growing demand for orders. Therefore, the establishment of an intelligent warehousing platform and the realization of the goods-to-people operation model are of great significance to the development of the warehousing industry.

As one of the important cores of intelligent warehousing, multi-robot task scheduling has attracted wide attention at home and abroad. Many scholars have studied it and achieved certain results. Rohini .M et al. [2]used linear hybrid programming to improve the genetic algorithm, with the goal of minimizing the total distance and minimizing the idle time of the robot; Yang Guihua et al. [3]proposed a warehouse robot task allocation algorithm based on the immune ant colony optimization algorithm, with the robot's total path as the goal, to achieve multi-objective task allocation; Zhou Hang et al.[4] proposed a hybrid genetic taboo search

algorithm, with the robot delivery cost as the goal, to improve the overall operation efficiency of the warehouse; Deng Fuqin et al. [5]proposed an algorithm combining genetic algorithm and rolling scheduling, with the time to complete the task as the goal, to improve the efficiency of multi-robot task allocation; Zhu Bowen et al. [6]proposed a multi-strategy optimized sparrow search algorithm, with the robot's total path as the goal, to improve the performance of multi-robot task scheduling.

Based on the above research content, this paper proposes an improved genetic algorithm for the warehouse multi-robot task allocation problem. First, the greedy algorithm is introduced in the population initialization process to improve the quality of the initial solution in the population; secondly, the bidirectional three-crossover operator is introduced to improve the quality of the offspring in the solution process; finally, a multi-variation strategy is designed, combined with the Metropolis criterion, to accept individuals with poor fitness with a certain probability, to ensure the diversity of the population, and to improve the ability of the algorithm to jump out of the local optimum.

2. Problem Description and Model Building

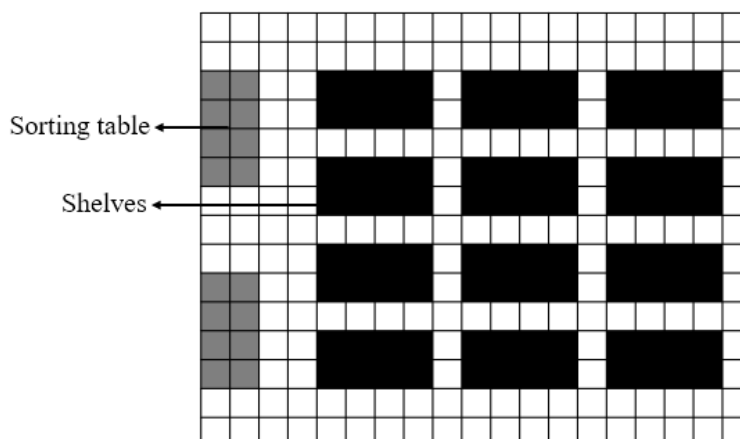


Figure 1. Intelligent Warehousing Grid Map

Multi-robot task allocation[7] refers to the allocation of several tasks to each robot without duplication, and the robot achieves the optimal result under the premise of satisfying the constraints. Smart warehousing consists of several sorting tables, shelves and a certain number of robots. For the multi-robot task allocation model, a smart warehousing grid map is established, as shown in Figure 1. The gray area in the model is the sorting table, the black area in the middle is several shelves, and the remaining white area is the passable area.

Based on the intelligent warehouse grid map, a multi-robot task allocation problem model is established [8] Assume that the number of robots is n , the robot set is $R = \{r_1, r_2, r_3, \dots, r_n\}$, the number of tasks to be assigned in the warehouse is m , and the task set is $T = \{t_1, t_2, t_3, \dots, t_m\}$, where $n \ll m$. R_i^i is used to represent the task point executed by the i -th robot, and t^i is the number of tasks that the robot needs to perform. The total distance S_i that each robot runs is as follows:

$$S_i = D_{R_i^1, S_i^0}^i + \sum_{k=1}^{t^i-1} D(R_i^k, R_i^{k+1}) + D_{R_i^i, S_i^0}^i \quad (1)$$

Where: S_i represents the total distance traveled by the i -th robot; $D_{R_i^1, S_i^0}^i$ represents the distance from the initial position of the i -th robot to the first task; $\sum_{k=1}^{t^i-1} D(R_i^k, R_i^{k+1})$ represents the total distance of the i -th robot completing t^i tasks; $D_{R_i^i, S_i^0}^i$ represents the distance from the last task point of the i -th robot to the initial position. The distance formula is as follows:

$$D(R^k, R^{k+1}) = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \quad (2)$$

In addition, the path distances taken by each robot to execute the task sequence are similar. In order to ensure that all robots have a high utilization rate, the variance of the path distance is used to reflect the degree of discreteness. The formula is as follows:

$$B = \sum_{i=1}^n \left(\frac{\sum_{i=1}^n S_i}{n} - S_i \right)^2 / n \quad (3)$$

The objective function is established with the goal of minimizing the sum of all robot paths and minimizing the task balance cost. The objective function expression is as follows:

$$F = \min \left(\sum_{i=1}^n S_i + B \right) \quad (4)$$

The constraints are:

$$\begin{cases} \sum_{i=1}^m t^i = m, i \in [1, m] \\ \text{mintask} = m' \end{cases} \quad (5)$$

Where: F is the objective function, which represents the total distance that all robots need to complete all tasks during

the task allocation process; $\sum_{i=1}^m t^i = m$ means that all tasks are allocated and there is no unallocated task; mintask is the minimum number of tasks performed by each robot.

3. Improved Genetic Algorithms

The genetic algorithm is a meta-heuristic optimization algorithm based on Darwin's "survival of the fittest" theory[9]. The algorithm generates an initial population and then iterates through operations such as selection, crossover, and mutation to search in the solution space to find the optimal solution to the problem. However, the genetic algorithm still has problems such as poor initial population quality and slow convergence in the solution process.

This paper adopts an improved genetic algorithm to solve the problem. First, a greedy algorithm is introduced to optimize the initial population to improve the quality of the initial solution; second, a heuristic two-way three-crossover operator is introduced to expand the search range to obtain high-quality local optimal solutions; finally, a multi-variation strategy is used in combination with the Metropolis criterion to perform a global search, and the diversity of the population is improved by accepting inferior solutions so that the algorithm can jump out of the local optimum and obtain the global optimal solution. The flowchart of the improved genetic algorithm is shown in Figure 2.

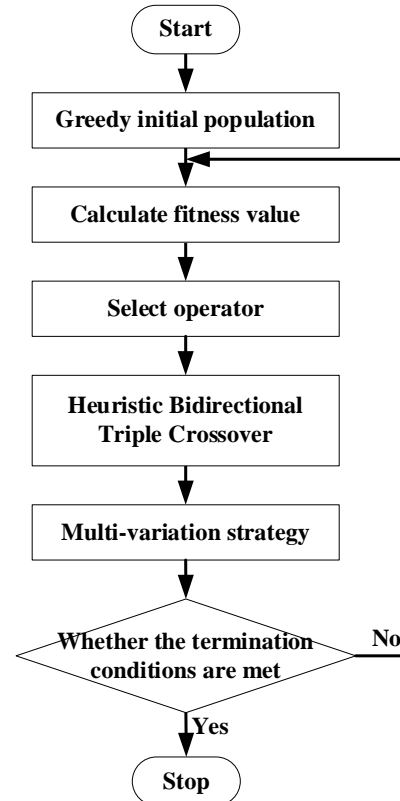


Figure 2. Flowchart of improved algorithm

3.1. Greedy Initial Population

3.1.1. Encoding

Common encoding methods for genetic algorithms include two-part chromosome design and double chromosome

design[10]. The search space of the above encoding methods is relatively small, and they are prone to fall into local optimality when solving large-scale problems[11]. This paper adopts single chromosome encoding based on virtual points. This encoding method is simple to design and easy to implement.

In order to explain the above encoding method, this article takes 3 robots and 8 tasks as an example. The chromosome individual is shown in Figure 3. 1~8 represent eight task points, and the two gray parts are virtual points. The virtual points are equivalent to the starting point to segment the robot. Therefore, the task route of the first robot is: $4 \rightarrow 6 \rightarrow 3 \rightarrow 4$; similarly, the task route of the second robot is: $4 \rightarrow 2 \rightarrow 1 \rightarrow 4$; finally, the task route of the third robot is: $4 \rightarrow 8 \rightarrow 5 \rightarrow 7 \rightarrow 4$.

4	6	3	9	2	1	10	8	5	7
---	---	---	---	---	---	----	---	---	---

Figure 3. Chromosome decoding based on virtual points

3.1.2. Greedy Algorithm

Table 1. Greedy algorithm

Algorithm: Greedy algorithm
$unvs_task = \{r_1, r_2, r_3, \dots, r_n\}$
$current_task = Random(1, task_num - 1)$
$path = [current_task]$
$vs_task = \{current_task\}$
Remove $current_task$ from $unvs_task$
While $unvs_task$ is not empty
$next_task = \min\{current_task_{dis} \text{ to } unvs_task_{dis}\}$
Append $next_task$ to $path$
Remove $next_task$ from $unvs_task$
Add $next_task$ to vs_task
$current_task = next_task$
$m = task_num // car_num$
for i in $car_num - 1$
if $i < m$ then
$new_path.EXTEND(path[:m])$
$new_path.APPEND(dummy_point)$
$i += 1$
$new_path.EXTEND(path)$

The initial population of traditional genetic algorithms is usually randomly generated by computers. The solution quality of the initial population generated in this way is poor, resulting in slow convergence of the algorithm [12]. To address the above problems, this paper uses a greedy algorithm to optimize and improve the quality of the initial population.

The process of generating individuals by the greedy algorithm is as follows: randomly select a starting point C_{strat} from the entire task list and add it to the individual, then find the task point C_{strat} closest to the starting point C_{next} in the remaining task list, add C_{next} to the individual

and continue to traverse the remaining tasks as the new starting point, and at the same time add virtual points to complete the segmentation according to the constraints and task balance constraints until all task points are traversed to form a complete chromosome. The pseudo code of the greedy algorithm is shown in Table 1.

3.2. Heuristic Bidirectional Triple Crossover Operator

Crossover strategy is the main method for genetic algorithms to generate new individuals. Different crossover strategies are suitable for different problems and optimization objectives. Common crossover strategies used in traditional genetic algorithms include sequential crossover and partial matching crossover[13]. However, these crossover strategies cannot obtain the global optimal solution during the solution process and may also destroy excellent individuals in the parent generation, resulting in a decrease in the fitness of the offspring. To address the above problems, this paper introduces a heuristic two-way three-crossover method, which selects multiple parents for crossover to obtain excellent offspring individuals and improve the solution accuracy.

This article takes 8 task points and 3 robots as an example to explain the process of heuristic two-way three-way crossover in detail, as shown in Figure 4. From the chromosome of parent 1, we can know that the number of tasks of each robot is 3, 2 and 3 respectively. Therefore, the task allocation method of the offspring is the same as that of parent 1. The selected parent is cross overed according to the following steps:

1) Select three parents, record the number of tasks assigned to parent 1, remove the virtual points and only keep the chromosomes of the task points for crossover.

2) Select the first task point of parent generation 1 as the starting point, as shown in Step 2 of Figure 4, and 4 as the starting point of the child generation. First, find the three task points 6 and 7 that are adjacent to the parent generation task point 4 in the forward direction; secondly, select the task point 7 that is closest to task point 4; finally, use the newly added task point as the starting point to continue looking for adjacent task points until the first robot task path is completed, as shown in Step 4.

3) After the first task sequence is completed, add a virtual point as a judgment, use the virtual point to represent task point 4, re-plan the task path for the second robot, find the closest task point, until the second task path is completed, as shown in Step 7. Similarly, complete the task path of the third robot and obtain the forward daughter chromosome.

4) Similarly, the reverse search is opposite to the forward search, and the same rules are used to obtain the reverse daughter chromosomes.

By performing crossover in the above-mentioned crossover method, the heuristic two-way three-way crossover can obtain a task path without repeated points. The offspring obtained by this crossover method not only inherits the excellent genes of the three parents at the same time, which is conducive to the production of high-quality offspring, but also can improve the solution speed of the algorithm.

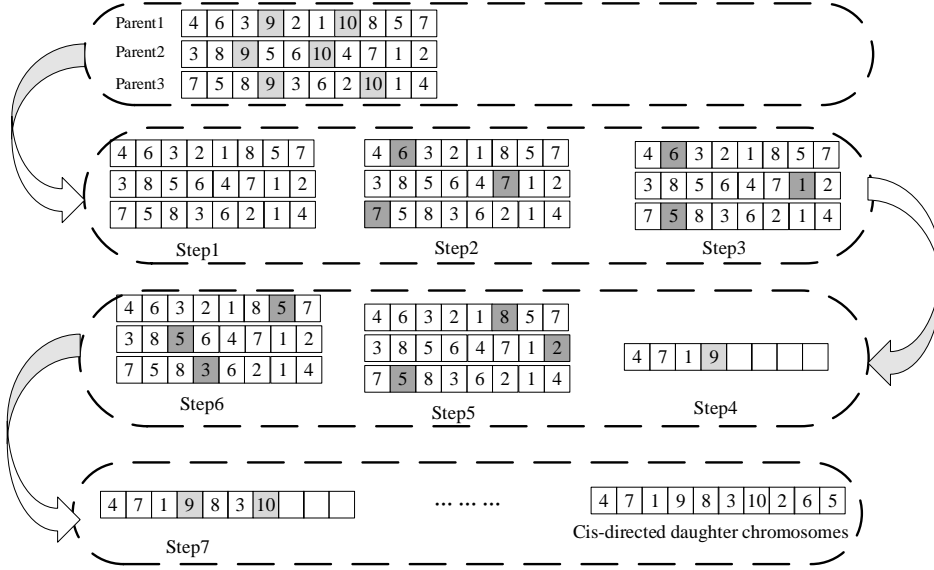


Figure 4. Heuristic two-way three-crossover process

3.3. Multi-variation Strategy

The main function of the mutation operator is to prevent the algorithm from falling into the local optimum during the iterative search process[14]. Traditional genetic algorithms only use single-point random mutation methods, which cannot guarantee that the algorithm will jump out of the local optimum. To address the above problems, this paper introduces insertion mutation and heuristic reversal mutation to improve the diversity of the population and enhance the ability of the algorithm to jump out of the local optimum.

The insertion mutation method randomly selects two points from the parent individuals and inserts the point with the front position behind the other point. This method can not only retain the excellent gene fragments of the parent, but also effectively increase the diversity of the population. Insertion mutation is shown in Figure 5.

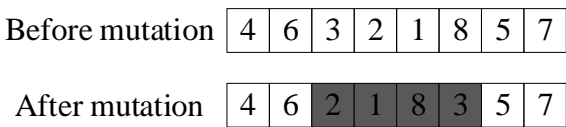


Figure 5. Insertion mutation

The heuristic reversal mutation method randomly selects a gene sequence in the parent generation, reverses its order, obtains a new individual, and then compares the fitness values before and after the mutation to decide whether to keep the offspring individual. The heuristic reversal mutation operation process is shown in Figure 6.

By adopting a multi-variation strategy, not only the excellent gene fragments in the parent generation are retained, but also the diversity of the population is improved, thereby enhancing the algorithm's ability to escape from the local optimum.

4. Simulation Results and Analysis

In order to further verify the performance of the algorithm proposed in this paper in solving the robot task allocation

problem, simulation experiments were carried out on it. The experimental environment was Python3.6. Three groups of simulation experiments were carried out and analyzed by taking the system to allocate 30, 50 and 100 tasks as examples. According to the tasks of different scales, the number of robots was 3, 5 and 10 respectively. The algorithm proposed in this paper was compared with the traditional genetic algorithm [15] (Genetic Algorithm, GA) and simulated annealing algorithm (Simulated Annealing, SA) in the experiment.

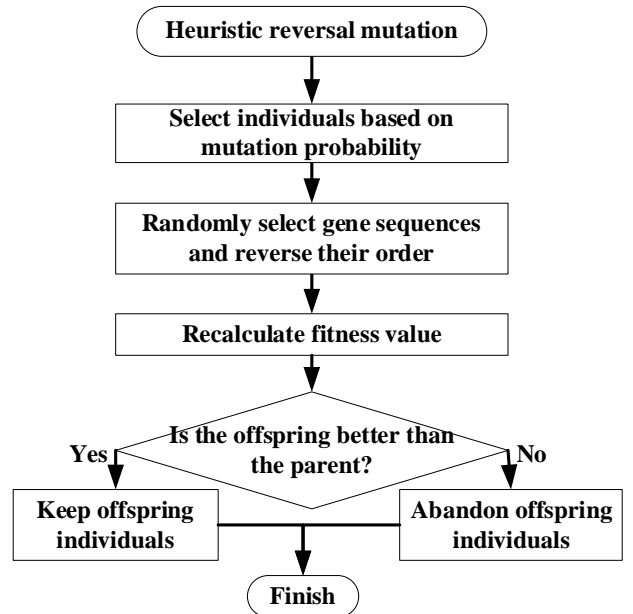


Figure 6. Heuristic reversal mutation process

The algorithm proposed in this paper is used to assign 30 tasks to 3 robots, and the task assignment results of each robot are shown in Table 2. According to the execution task sequence in the table, it can be seen intuitively that the task allocation scheme of the algorithm in this paper tends to be balanced and more reasonable.

Table 2. Task allocation results

Robot Number	Execute Task Sequence	Maximum single robot path /m	Total path cost /m
Robot 1	25,28,30,27,29,24,26,2,13,14	247	721
Robot 2	1,11,9,7,10,12,5,8,5,4,3		
Robot 3	15,20,17,18,21,23,19,21,16		

In order to fully verify the feasibility of the improved algorithm in this paper, the average values of the three algorithms running 20 times under three different task scales

are compared. The average values of the robot's total path, the maximum single-vehicle path and the number of algorithm iterations are shown in Table 3.

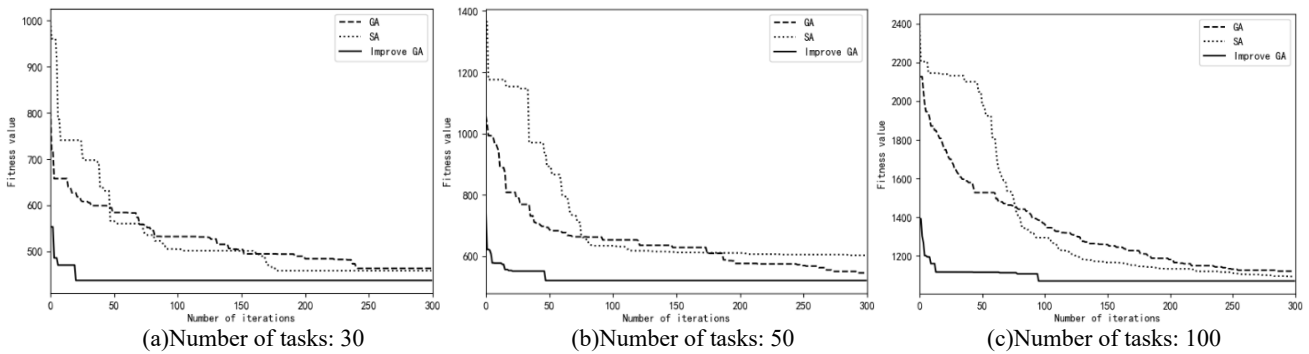
Table 3. Experimental results of different algorithms and different task scales

Number of tasks	Algorithm	Average total path /m	Average maximum single robot path /m	Average number of iterations
30	GA	845.7	320	200
	SA	1069.7	357.4	180
	Improve GA	731.3	247.4	35
50	GA	1019.4	211.4	220
	SA	998.4	199.5	140
	Improve GA	852	182.5	52
100	GA	1850	507	280
	SA	1776.8	225	200
	Improve GA	1724.5	205.8	101

It can be seen from Table 3 that the improved GA algorithm proposed in this paper has certain improvements in convergence speed and solution accuracy. It can be seen from the maximum single-vehicle path that the solution of the

improved GA algorithm is more reasonable.

Figure 7 shows the convergence curves of the GA, SA and improved GA algorithms when assigning tasks of different sizes.

**Figure 7.** Convergence curves of different scale task allocation algorithms

From the convergence curve and fitness value in Figure 7, we can see that the improved GA algorithm proposed in this paper performs well in solving the initial population. It expands the search range through heuristic two-way three-crossover, and the multi-mutation strategy enables the algorithm to jump out of the local optimum.

5. Conclusion

This paper proposes an improved genetic algorithm to solve the problems of low efficiency and unbalanced task allocation of traditional genetic algorithms in multi-robot task allocation in intelligent warehousing. First, the greedy algorithm is used to generate the initial population to improve the quality of the initial population of the genetic algorithm; secondly, a heuristic two-way three-crossover operator is designed to improve the search ability of the algorithm; finally, a multi-mutation strategy is introduced to enhance the ability of the algorithm to jump out of the local optimum. Through experimental comparison, the algorithm in this paper has improved the balance of task allocation and solution efficiency, can adapt to the task allocation of warehousing environments with different degrees of task numbers, and has obvious advantages in solving multi-robot task allocation.

Acknowledgments

Jilin Province Science and Technology Development Project (YDZJ202201ZYTS555).

References

- [1] Li Yansheng, Wan Yong, Zhang Yi, et al. Path planning of storage robot based on artificial bee colony-adaptive genetic algorithm [J]. Chinese Journal of Scientific Instrument, 2022, 43 (04): 282-290.
- [2] M. Rohini., B. Manohari. and S. Adhithyan., Genetic Algorithm Based Optimization in Solving Multi Robot Task Allocation Problems, 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 344-350.
- [3] Yang Guihua, Wei Jiale. Warehouse task scheduling based on immune ant colony optimization algorithm[J]. Modular Machine Tools and Automated Processing Technology, 2023, (01):179-183.
- [4] Zhou Hang, Qin Shihong, Fang Jingcheng. Multi-robot task allocation based on hybrid genetic taboo search algorithm[J]. Automation & Instrumentation, 2023, 38(11):35-39.

- [5] Deng Fuqin, Huang Huanzhao, Tan Chaoen, et al. Multi-robot task allocation algorithm combining genetic algorithm and rolling scheduling[J]. *Journal of Computer Applications*,2023, 43 (12):3833-3839.
- [6] Zhu Bowen, Cui Fengying. Warehouse robot task scheduling algorithm integrating multi-strategy optimization SSA[J]. *Combined Machine Tool and Automation Processing Technology*, 2024,(05):183-187+192.
- [7] Hamza C, François G ,Edouard L ,et al .Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art[J].*Robotics and Autonomous Systems*, 2023, 168.
- [8] Wang Jing, Zhang Gong, Zheng Jiahong, et al. Research review and prospect of multi-robot optimization layout and task allocation [J]. *Machine Tool and Hydraulics*,2021,49(16):161-167.
- [9] Sun Bing, Wang Chuan, Yang Qiang, et al. Evolutionary algorithm for multi-starting point equilibrium multi-traveling salesman problem[J]. *Computer Engineering and Design*, 2023, 44 (07): 2030-2038.
- [10] S. Yang et al., A Novel Maximin-Based Multi-Objective Evolutionary Algorithm Using One-by-One Update Scheme for Multi-Robot Scheduling Optimization. in *IEEE Access*, 2021, vol. 9,121316-121328.
- [11] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J] .*Systems Science& Control Engineering*, 2020,8(1):22-34.
- [12] CHEN K-C, LIN S-C, HSIAO J-H, et al. Wireless networked multirobot systems in smart factories [J]. *Proceedings of the IEEE*, 2020, 109(4): 468-494.
- [13] KIM J, SON HI. AVoronoi diagram-based workspace partition for weak cooperation of multi-robot system in orchard [J]. *IEEE Access*, 2020, 8: 20676-20686.
- [14] Yang Wei, Li Ran, Zhang Kun. Optimization of multi-automatic guided vehicle task allocation based on variable neighborhood simulated annealing algorithm[J]. *Journal of Computer Applications*,2021,41(10):3056-3062.
- [15] Katoch, S., Chauhan, S.S. & Kumar, V. A review on genetic algorithm: past, present, and future. *Multimed Tools Appl*, 2021, 80, 8091–8126.