International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 5115-5125

https://doi.org/10.70517/ijhsa464353

Elite Strategy Enhances NSGA-II for Multi-Robot Task Allocation in Orbital Bolting Operations

Yanni Shen[1] and Jianjun Meng[1,2,3,4,*]
[1] School of Mechanical Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu, 730070, China
[2] Institute of Mechanical and Electrical Technology, Lanzhou Jiaotong University, Lanzhou, Gansu, 730070, China
[3] Gansu Engineering Technology Research Center of Logistics and Transportation Equipment Informatization, Lanzhou, Gansu, 730070, China
[4] Gansu Logistics and Transportation Equipment Industry Technology Center, Lanzhou, Gansu, 730070, China
Corresponding authors: (e-mail: 13220039@ stu.lzjtu.edu.cn).

Abstract Reasonable task allocation not only improves the efficiency of task execution, but also reduces the total working time and energy consumption of the robot system. In this paper, an improved NSGA-II algorithm based on elite strategy is proposed for the multi-robot task allocation problem in orbital bolt operations. By combining elite selection, congestion ranking and adaptive cross-variance probability, this algorithm is able to achieve a better balance in multi-objective optimization. Experimental results show that the improved algorithm can significantly reduce the total distance traveled by the multi-robot system and effectively reduce the path deviation when dealing with different capacity datasets. For example, on the Kro_A100 dataset, the maximum path deviation is 0.14%, which is much lower than the traditional method. Through simulation experiments, when the algorithm runs in a space of 4000m×2000m, the path length of the shortest total time-consuming scheme is 42332.1 m, and the path length of the least power-consuming scheme is 32924.5 m. The results show that the improved NSGA-II algorithm not only improves the balanced robot path allocation, but also optimizes the task execution time and energy consumption. The method is highly scalable and applicable, and can provide an effective solution for practical multi-robot task allocation problems.

Index Terms Multi-robot task assignment, NSGA-II algorithm, elite strategy, path deviation, optimization, simulation

I. Introduction

As a major mode of transportation in China's transportation system, railroad transportation plays a very important role in China's economic and social development. Strengthening the construction of modern railroads, improving the transportation capacity of passengers and goods, building a modern comprehensive transportation system, and constructing a strong transportation country are of great significance in realizing the great rejuvenation of the Chinese nation [1], [2]. Based on this, with the help of artificial intelligence, cloud computing, Internet of Things and other technologies, the main industrial countries have put forward a strategic plan for intelligent manufacturing to boost the transformation and upgrading of the manufacturing industry from digital manufacturing to intelligent manufacturing [3], [4].

Railroad fastener is an important part to connect the railroad sleeper with the railroad, which has an extremely important role in ensuring the safety of train operation [5]. Although today's railroad construction has appeared large-scale mechanized equipment such as rail-laying cars, but in the operation area such as fastener assembly and bolt fastening is still used in the traditional manual operation [6]-[8]. Workers rely on experience to fasten bolts is very easy to appear “over-tightening” and “under-tightening” phenomenon, resulting in potential hidden dangers such as loose fasteners, breakage, and other more serious consequences may be incalculable [9]-[11]. Therefore, it is of great social value to apply intelligent assembly technology to railroad construction to improve the construction level and save human resources at the same time [12]. It is a general trend to study a high-precision, high-efficiency, and high-reliability automated fastener assembly and bolt tightening control algorithm and apply it to rail bolt work robots [13], [14].

In this paper, a multi-objective optimization framework based on the NSGA-II algorithm is adopted and improved by combining the elite strategy with the aim of enhancing the algorithm's search capability and stability in complex task environments. In the research methodology, the local search capability and global exploration performance of the algorithm are further improved by introducing the adaptive cross-variance probability and differential evolutionary variance strategies. Aiming at multiple constraints in task allocation, such as the limit of the number of robot tasks and the cost of the tasks, this paper designs a specific task allocation model suitable for orbital bolting operations by optimizing the objective function. The experimental results show that the improved algorithm can stably improve the efficiency and fairness of task execution in multiple task scenarios.

II. Improved NSGA-II algorithm based on elite strategy

II. A. Multi-robot task allocation in rail bolting operations

II. A. 1) Multi-robot task allocation problem description and classification

In the study of multiple robots in rail bolt operations, taking into account the performance of the robot, the robot's load and other parameter limitations, mainly used in the more general tasks related to task allocation between multiple mobile robots. The main mode relationships in the task allocation set are shown in Fig. 1, which mainly introduces the relationship between several allocation modes commonly used in task allocation.

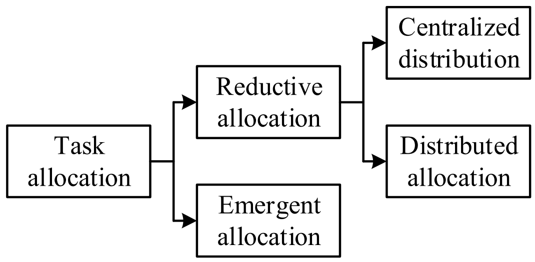

Figure 1: shows the relationships among several main patterns of task allocation

(1) Centralized assignment

By having a centralized robot in the environment, the centralized robot knows the task information in the environment, centralizes the processing of the collected task information, and assigns tasks to other robots so as to compute a set of optimal task sequence results.

(2) Distributed allocation

Part of the centralized central robot is removed, and each robot in the multi-robot system in the rail bolt operation is independent and equal, and all the robots make independent decisions, and with the change of the work scene, the distributed allocation can be better adapted, better handled, and more stable.

II. A. 2) Mathematical model of tasking

Multi-Robot Task Assignment in Orbital Bolt Operations Assume that there are n mobile robots in the system, denoted sequentially as $\{R_1, R_2, R_3, ..., R_n\}$. Perform m tasks, denoted sequentially as $\{T_1, T_2, T_3, ..., T_m\}$, and satisfy $n \le m$, the specific mathematical model is as follows.

(1) Constraint on the number of tasks performed by the robot

In the system, it is set that only one mobile robot is allowed to work at a target task point, and a robot can only execute one target task point at each time, as shown in equation (1).

$$Q_{ij} = 1 \text{ or } Q_{ij} = 0 \tag{1}$$

where, $Q_{ij} = 1$ - the i th robot performs the j th task, $Q_{ij} = 0$ - the i th robot does not perform the j th task, i - the serial number of the robot, j - the serial number of the task,

(2) Number of tasks for all robots constraints

It is required that all tasks must be executed as shown in equation (2).

$$\sum_{i=1}^{i=n} \sum_{j=1}^{j=m} Q_{ij} = m \tag{2}$$

where, m - total number of tasks, $i = 1, 2, 3, ..., n$ $j = 1, 2, 3, ..., m$.

(3) Robot execution task cost

At the end of execution of the assigned task point, each robot will have an execution task cost as shown in equation (3).

$$Cost = \{C_i \mid \mathrm{i} = 1, 2, ..., n\} \tag{3}$$

In the formula, $cost$ - robot execution task cost, C_i - the i th robot performs the task cost.

(4) Objective function

After calculating the target task points in the system for reasonable planning, the total execution task cost of all robots in the system is minimized as shown in equation (4).

$$\min[Sum(Cost)] = \sum_{i=1}^{n} C_i \tag{4}$$

where, $\min[Sun(Cost)]$ - the objective function, i.e., minimize the cost of the total execution of tasks by all robots.

II. A. 3) Task-allocation solving algorithms

Genetic algorithm parameters include crossover operation probability, mutation operation probability, population operation termination iteration number. The selection operation in the genetic algorithm is to pick out the offspring with high fitness function value for crossover operator and mutation operator operation, most of the current use of the carousel selection operator, which is a stochastic probabilistic method, the carousel is divided into a number of portions, and each portion represents an individual in the population, the carousel is rotated until it does not move, the hereditary to the next generation and carry out the next step is the portion of the pointer pointed to, while the probability of being selected on the carousel The size is related to the size of the fitness function value of each individual, the probability P_i of a specific individual i being selected is shown in equation (5).

$$P_i = \frac{f_i}{\sum_{i=1}^{M} f_i} (i = 1,2,3,...,M) \tag{5}$$

where, M - size of the population size, f_i - the value of the fitness function of individuals in the population.

II. B.NSGA and NSGA-II

II. B. 1) Fast non-dominated sorting methods

The non-dominated sorting method stratifies the individuals in the population according to the dominance relationship and guides the search towards the Pareto optimal solution, and the time complexity of non-dominated sorting in NSGA is $O(mN^3)$. Due to the high time complexity, the computational time of the algorithm is quite long when the population size is large and the number of reproduction generations is high [15]. In this regard NSGA-II proposes a fast non-dominated sorting method, which consists of the following two parts:

In the first part, two variables n_p and S_p are set and initialized for all the solutions in the population, where $p = 1,2,\cdots,N$, and n_p is used to count the number of solutions that dominate the solution p, and S_p is used to count the set of solutions that are dominated by solution p.

In the second part, the sorted individuals are stratified. Individuals with $n_p = 0$ (i.e., no other solution can dominate p) are put into the first layer and removed from the population. The number of layers is then increased by 1, and the above operation is continued until all individuals have been stratified.

In the first part a double traversal of all the solutions of the population is required to compute n_p and S_p for each solution, so the time complexity of the operation is $O(mN^2)$, and in the second part the operation has a time complexity of $O(N^2)$, so the time complexity of this sorting method is $O(mN^2) + O(N^2)$, which is $O(mN^2)$.

The specific process is as follows:

Step1. Parameterize all individuals n_p and S_p in the population such that $n_p = 0, S_p = \varnothing$, $p = 1,2,\cdots,N$.

Step2: Perform non-domination judgment on individuals in the population, let p,q be any two individuals in the middle mass, if p dominates q, then $S_p = S_p \cup \{q\}, n_q = n_q + 1$. If q dominates p, then $S_q = S_q \cup \{p\}$ and $n_p = n_p + 1$.

Step3. Set the initial value $k = 1$ for the current population stratification number k.

Step4. Remove the individuals in the population with $n_p = 0$ and add them to the stratification set F_k, i.e. $F_k = F_k \cup \{p\}$.

Step5: Judge whether F_k is empty, if not, reduce n_p corresponding to all individuals S_p in F_k by 1, and $k = k + 1$, jump to Step2, if empty then end the operation.

II. B. 2) Elite Selection Strategy

The method is to increase the diversity of the population by co-competition between the parent and the offspring. The schematic diagram of the elite selection process is shown in Fig. 2, in which parent P_i and offspring Q_i together form a new temporary population R_i, and non-dominated sorting stratification is performed on R_i, and finally the N best individuals are selected by the non-dominated sorting stratification number and crowding distance in R_i to form the next generation of the parent population.

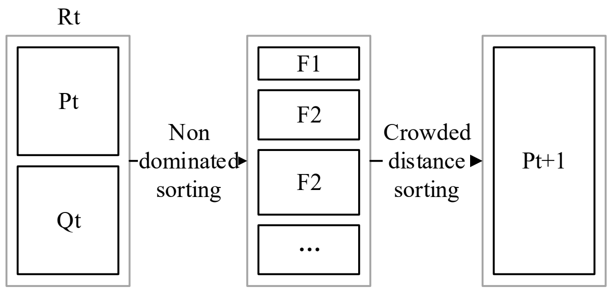

Figure 2: Schematic diagram of the elite selection process

II. B. 3) Crowding distance

The crowding distance is used to better understand the distribution of other solutions around each solution, and to prevent localized piling up of individuals and formation of local extremes during the evolutionary process. It indicates the density of individuals within the same non-dominated sorting stratum, and the crowding distance is a criterion for judging the superiority or inferiority of individuals within the same sorting stratum. Crowding distance is calculated by summing the absolute value of the difference between the distances of individuals within the same dominated stratum in each direction of searching for superiority and the distances of the two individuals close to them. For example, the crowding distance of individual i in the k th optimality-seeking direction f_k is $|f_k^{i+1} - f_k^{i-1}|, k = 1, 2, \cdots, m$, m is the number of the target direction, and f_k^{i+1}, f_k^{i-1} is the value of the difference between the two individuals that are close to individual i in the k th value of the two individuals that are close to each other on the target. Individual i crowding distance d_i is computed as in equation (6), where f_1, f_2 are two different optimization seeking directions.

$$d_i = \sum_{k=1}^{m} (|f_k^{i+1} - f_k^{i-1}|) \tag{6}$$

II. B. 4) Congestion selection operator

In order to select the optimal solution during the evolutionary process, the elite strategy requires the selection of individuals, and if the selected solution is in a different nondominated sorting stratum, then the solution with the lower stratum tier is selected. If the selected solution is in the same sorting stratum, then the solution with greater crowding, i.e., the solution in which the region of the solution is sparser, is selected.

II. B. 5) NSGA-II basic solution process

The solution process of NSGA-II is roughly the same as the traditional genetic algorithm, which is described as follows:

Step1: Initialize the running parameters of the algorithm, randomly generate an initial population $P_t, t = 0$ of size N in the specified search area, t is the number of generations of the population, and take P_t as the parent population.

Step2. Perform selection operation, simulated binary crossover operation, and polynomial variation operation on the parent population P_t in order, and use the resulting population Q_i as the child population.

Step3. The parent and offspring populations are subjected to an elite selection strategy, and the selected population is used as the parent population in the genetic process of the next generation, so that $t = t + 1$.

Step4. Determine whether the number of evolutionary times reaches the maximum value, if so, end the run and take the result of the last generation as the optimal solution. If not reached, jump to Step2.

II. C.Improved NSGA-II algorithm

II. C. 1) Adaptive cross-variance probabilities

The NSGA-II algorithm also has a cross-variance probability, which is set in the same way as the simple genetic algorithm, choosing an invariable cross-variance probability value. In the process of calculation, the value of the cross variance probability still has a certain influence on the calculation results, and has the disadvantages of easily falling into the local optimal solution and converging prematurely. In order to solve the computational error brought by the fixed parameters, the error is usually reduced by adaptive adjustment. Adaptive adjustment adopts setting two sets of cross-variation probabilities as the upper and lower bounds of change, and adjusts the cross-variation rate with the help of the information of the population at the time of evolution, so as to improve the search ability for the optimal solution. Currently, scholars use more adaptive cross-variation probability values set as shown in Equation (7) and Equation (8):

$$P_c(g) = P_{c\max} - (P_{c\max} - P_{c\min}) \times \frac{g}{gen} \tag{7}$$

$$P_m(g) = P_{m\max} - (P_{m\max} - P_{m\min}) \times \frac{g}{gen} \tag{8}$$

where g denotes the current number of evolved generations, $P_c(g)$ and $P_m(g)$ denote the magnitude of the crossover mutation probability at the time of evolution up to the current number of generations, gen denotes the total number of evolved generations, and $P_{c\max}$ and $P_{c\min}$ denote the value of the preset upper and lower bounds on the crossover probability. $P_{m\max}$ and $P_{m\min}$ denote the values of preset upper and lower bounds on the probability of variation.

In order to make the adaptive adjustment related to the population characteristics, the adjustment proposed in this paper is shown in Equation (9) and Equation (10):

$$P_m = \begin{cases} P_{m\max} - \dfrac{f_{\max} - f_{avg}}{f_{\max} - f_{\min}} \times (P_{m\max} - P_{m\min}) & else \\ P_{m\min} & f_{avg} = f_{\min} \end{cases} \tag{9}$$

$$P_c = \begin{cases} P_{c\max} - \dfrac{f_{\max} - f_{avg}}{f_{\max} - f_{\min}} \times (P_{c\max} - P_{c\min}) & else \\ P_{c\min} & f_{avg} = f_{\min} \end{cases} \tag{10}$$

where $f_{\max}, f_{\min}$ represent the maximum and minimum values of the objective function corresponding to the individuals in the current population, respectively, and f_{avg} is the average objective function value of all individuals in the current population [16]. In the early stages of evolution, there was a large difference between individuals, and this difference was mainly reflected in the difference in the value of the objective function. At this time, the maximum value of the individual objective function in the population $f_{\max}$ is quite different from the minimum value $f_{\min}$, and the number of individuals in the larger and smaller is about the same, at this time, f_{avg} is approximately equal to the average value of $f_{\max}$ and $f_{\min}$, $(f_{\max} - f_{avg})/(f_{\max} - f_{\min})$ is roughly 0.5, and the probability of cross-variation obtained is large, which is helpful for the algorithm to conduct a global search in the early stage to find the optimal solution set. When the objective function value obtained by most individuals in the population is roughly the same, f_{avg} is a value slightly greater than $f_{\min}$, because the number of individuals with a larger objective function value is very small, so the maximum and minimum difference between the average fitness is evenly distributed to each individual, and the value of f_{avg} and $f_{\min}$ is not much different. When all individuals have evolved to the optimal solution at this time, a limit condition is reached, that is, $f_{avg} = f_{\min}$, then the value of $(f_{\max} - f_{avg})/(f_{\max} - f_{\min})$ is close to 1, and the probability of cross-variation is relatively adjusted to a smaller number, which improves the ability of local optimization, which is consistent with the direction of individual optimization in the population. It is beneficial for the search of the optimal solution set.

II. C. 2) Improvement of the variational operator

After the differential evolutionary algorithm mutation operation of the population characteristics of better, help in the excellent solution to find more excellent solution of the individual.NSGA-II algorithm used in the mutation operator in the local optimization does not rely on the other individuals of the population information, has a certain degree of randomness, so here the differential evolutionary algorithm of the mutation strategy is used to improve the mutation operator as shown in Equation (11):

$$x_i(g+1) = x_{best}(g) + F \times (x_{r_1}(g) - x_{r_2}(g)) \tag{11}$$

where $x_{best}(g)$ is the best individual in the g th generation.

II. C. 3) Improvements to the intersection operator

The crossover operator used in NSGA-II has the characteristic of ensuring that the algorithm converges to a globally optimal solution, which guarantees that the offspring individuals retain some of the information of the parent individuals, as defined in Eq:

$$X_A^{t+!} = \alpha X_A^t + (1-\alpha) X_B^t \tag{12}$$

$$X_B^{t+!} = \alpha X_B^t + (1-\alpha) X_A^t \tag{13}$$

where α is the crossover factor, a deterministic constant. The advantage of this strategy is that the excellent individuals in the parent generation can be inherited to the offspring, but this strategy is relatively weak in the global search performance, which is easy to cause the overpopulation of the solutions with excellent, and can not ensure the population diversity well. So here, by redefining the value of α to improve the global search ability, try to leave the individuals with high non-dominated sorting rank, so the treatment of using a combination of non-dominated sorting rank and α is adopted, which enriches the diversity of the solution while trying to retain the individuals with high sorting rank.

$$\alpha = \frac{rankA}{rankA + rankB} \tag{14}$$

where rankA, rankB denote the non-dominated sorting level of individual A and individual B respectively, this treatment firstly ensures that this parameter is consistent with the range of the previous constant parameter. The second is to link the non-dominated sorting level and the crossover factor α, so that α can be changed with the help of the information of other individuals in the population, from increasing the proportion of individuals with low non-dominated sorting level in the progeny, and then improving the quality of individuals in the next generation of the population.

II. C. 4) Improvement of elite strategies

In order to further enrich the diversity of the population, the original elite strategy was modified and the improved elite strategy is shown in Figure 3.

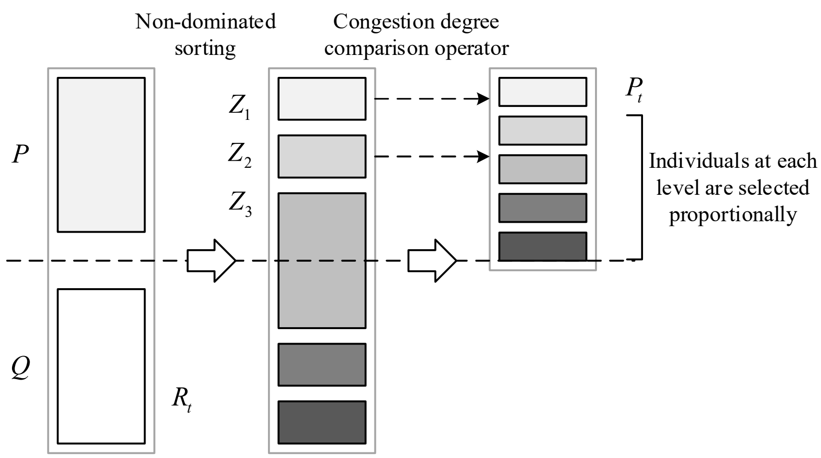

Figure 3: Improved Elite Strategy

The modified elite strategy still leaves intact the smallest individuals in the nondominant tier, i.e., those in the first tier. Individuals in other tiers are selected proportionally. The number of individuals selected for different tiers is shown in equation (15):

$$N_i = \frac{2(N - N_1)(M + 1 - i)}{(M + 2)(M - 1)} \quad (i >= 2) \tag{15}$$

where N is the size of the population, M is the maximum value of the non-dominated tier of the current population, i is the number of the tier in which individual selection is currently required, and N_i is the number of individuals selected from the i th tier. Equation (15) starts the selection from the second level of the non-dominated ordering because the first level individuals are optimal solutions and need to be kept intact. The selection of individuals from the second level should be done in such a way that as many individuals as possible from the lower levels of the sorting hierarchy are retained. When selecting individuals at lower sorting levels, Eq. $M+1-i$ is larger and more individuals are selected. When individuals are selected at a higher sorting hierarchy, $M+1-i$ is smaller and fewer individuals are selected at this hierarchy. With this improved elite strategy, it is ensured that individuals in the first tier of the non-dominated sort are retained and directly involved in the next evolutionary operation. It also selects different proportions of individuals from other tiers, which ensures the diversity of the population, enriches the population, provides a source of individuals for the next generation of cross-mutation operation, and further enhances the algorithm's local optimization seeking ability.

III. Experimental validation and result analysis

III. A. Algorithm testing

III. A. 1) Description of the algorithm

In this paper, the eil class (eil51) and Kro class (Kro_100, Kro_150, Kro_A200) of the TSPLIB dataset are used as the test datasets for side-by-side comparisons of different numbers of machines on datasets of different capacities, respectively.

III. A. 2) Calculations

Since the optimization objective is to minimize the total distance traveled by the multi-robot system and to reduce the distance variance of the multi-robot system, the percentage of the longest distance and the maximum path deviation of each group of robots are chosen as the judging criteria, which are used as a measure of the performance of the algorithm for the balanced performance of the task assignment and the longest execution time. Python was used for programming, and the tests were conducted on a desktop computer with a CPU of 3.5 GHz and 6 GB of RAM. The maximum sub-paths of different robots under 100, 150 and 200 nodes are shown in Fig. 4 to Fig. 6, respectively. With the increase in the number of participating transportation machines, the transportation path length of each transportation robot can be effectively reduced, thus reducing the transportation time and effectively reducing the task load of each robot.

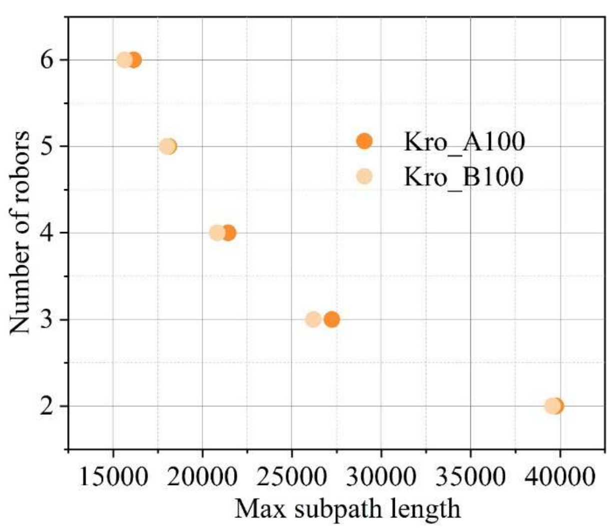

Figure 4: The maximum subpath of different robots under 100 nodes

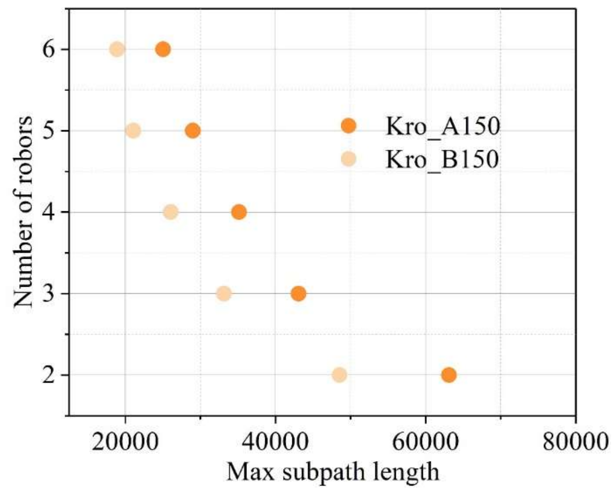

Figure 5: The maximum subpath of different robots under 150 nodes

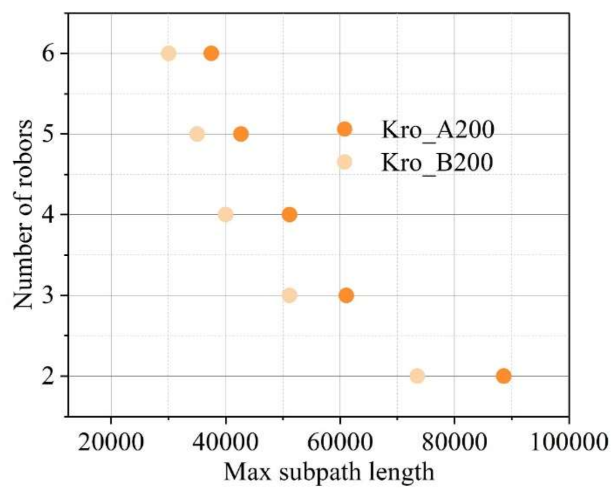

Figure 6: The maximum subpath of different robots under 200 nodes

The percentage of deviation from the maximum path for each robot is shown in Table 1. From the statistics in the table, it can be seen that the difference between the maximum path and the average path of each robot is less than 2% of the average path, thus ensuring that the probability of the smaller robots with smaller motion paths being idle due to the long working time of the robots with larger motion paths is low.

Table 1: The maximum path deviation of each machine is the percentage

Number of robots	2	3	4	5	6
Kro_A100	0.14	0.37	0.65	1.23	1.54
Kro_A150	0.13	0.26	0.55	0.7	1.15
Kro_A200	0.08	0.18	0.41	0.66	0.86
Kro_B100	0.16	0.35	0.66	1.27	1.53
Kro_B150	0.09	0.27	0.63	0.95	1.14
Kro_B200	0.09	0.22	0.41	0.65	1.32

The curve of optimal individual scores per generation is shown in Fig. 7. The global optimal solution score change curve is shown in Fig. 8. From the figure, it can be seen that the population underwent six restarts and obtained a new optimal value in the last elite pool restart (after 700 generations).

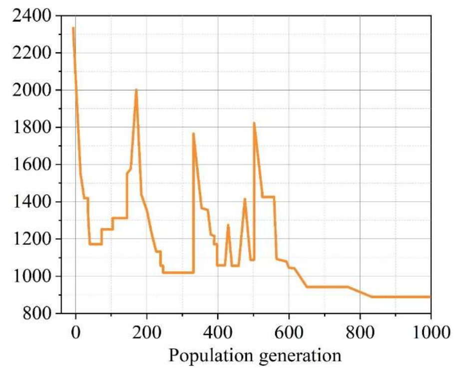

Figure 7: Each generation optimal individual score curve

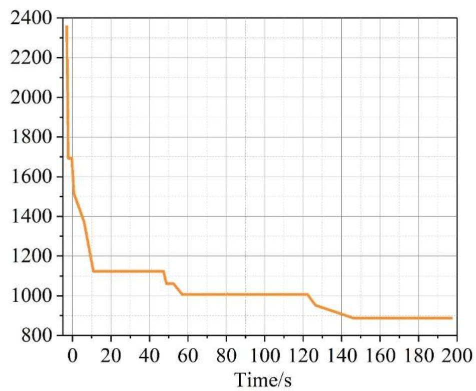

Figure 8: The global optimal solution score curve

III. B. Simulation experiment test

The simulation environment in this section is set up in a space of 4000m□2000m with 1 base station, 100 task target points and 5 mobile robots performing inspection tasks, all robots uniformly start from the base station, complete the assigned inspection tasks in turn and return to the base station. The relevant parameters in the improved NSGA-II with constraints are: the objective function is the time cost function f_1 and the power cost function f_2, crossover probability=0.9cP, variance probability=0.1mP, the threshold in the crossover operation Q=63000, the population size S=350, and the maximal number of iterations maxT=4000. The Pareto frontiers solved by NSGA-II are shown in Fig. 9. In the figure f_1 is the time cost function and f_2 is the power cost function. Each point in the graph represents an allocation scheme that satisfies the constraints, and due to the overabundance of generated schemes, the one that takes the shortest time, consumes the least amount of electricity (shortest total distance traveled), and the one that is randomly selected is taken.

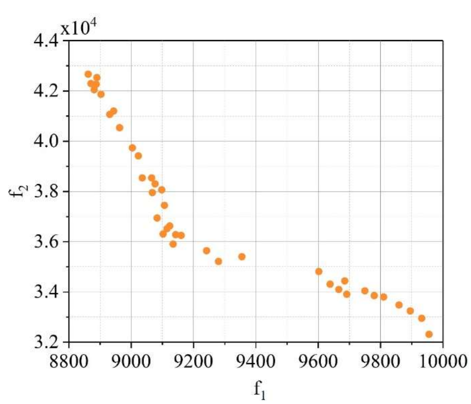

Figure 9: NSGA-II solves the front edge of the pareto

The results of the total least time consuming task allocation are shown in Table 2. The result of the least power consumption task allocation is shown in Table 3. The task assignment results for the random selection scheme are shown in Table 4. The total path length in Table 2 is 42332.1 m, the total path length in Table 3 is 32924.5 m, and the total path length in Table 4 is 38342.3 m. From Table 2, it can be seen that when the desired goal is to complete all the tasks with the shortest total elapsed time, the distance traveled by all robots is similar, and the robots have the highest efficiency, but the total distance traveled by all the robots is relatively longer, and the power consumption is higher. From Table 3, it can be seen that when the desired goal is to complete all the tasks the total distance traveled by the robot is the shortest, the distance traveled by each robot is more different, the utilization efficiency of the robot is low, and it takes longer to complete all the tasks. Table 4 shows a set of randomly selected allocation schemes whose total time to complete the tasks and total distance traveled by the robots are in between the previous two. The improved NSGA-II with constraints proposed in this chapter provides a set of compliant allocation schemes for the MRTA problem, which can be selected from the generated Pareto frontiers according to the actual needs and suitable for the actual situation.

Table 2: Total time-consuming shortest task distribution results

Robot number	Task sequence number	Path length/m
Robot1	0 85 41 62 40 45 66 11 6 22 55 83 48 26 77 53 67 25 88 91 0	8485.8
Robot2	0 58 8 72 90 80 15 54 79 32 28 45 3 35 24 79 36 59 55 23 93 87 42 59 8 44 14 66 0	8839.2
Robot3	0 56 82 26 48 92 97 76 14 63 10 0	8865.6
Robot4	0 97 74 30 47 -2 81 91 12 100 34 38 3 46 75 92 34 71 64 39 0	8801.7
Robot5	0 70 24 14 83 92 16 21 36 99 35 67 17 72 51 19 15 14 30 43 4 95 13 84 76 49 0	7545.1

Table 3: Minimum power consumption

Robot number	Task sequence number	Path length/m
Robot1	0 49 73 74 9 90 96 1 59 0	3478.4
Robot2	6 33 58 48 71 21 92 90 43 29 11 14 20 55 73 5 1 70 34 101 38 25 18 89 83 21 21 69 0	9846.5
Robot3	0 60 25 78 66 79 36 32 41 99 102 72 13 -3 40 43 25 37 81 53 23 44 60 8 47 19 65 0	9955.6
Robot4	0 95 0	338.8
Robot5	0 80 36 87 52 40 58 -3 75 89 14 98 35 38 -1 51 80 95 45 46 72 72 40 56 0	9287.8

Table 4: Task distribution of random selection schemes

Robot number	Task sequence number	Path length/m
Robot1	0 69 68 71 77 32 52 91 99 71 25 69 27 72 0	9045.4
Robot2	0 59 0 65 88 2 84 55 17 73 45 23 12 7 12 63 72 26 62 33 100 37 27 78 7 92 76 0	8634.5
Robot3	0 57 81 13 93 83 19 15 66 0	2827.9
Robot4	0 95 69 26 92 53 -2 84 90 24 101 34 42 7 56 78 88 44 45 39 67 45 41 59 0	8959.6
Robot5	0 93 14 23 51 69 49 84 11 3 85 37 33 38 0 37 8 4 12 19 78 32 63 55 24 94 87 50 62 14 46 23 0	8928.6

IV. Conclusion

In this study, the improved NSGA-II algorithm is used to solve the multi-robot task allocation problem in orbital bolting operations, and a more significant optimization effect is achieved.

The experimental results show that the improved algorithm performs well in reducing the robot path length and optimizing the task allocation efficiency on different datasets. On the Kro_A100 dataset, the maximum path deviation is only 0.14%, which is much lower than that of the traditional method, showing the advantage of the algorithm in the equalization of task allocation. Further simulation tests show that the task allocation using the improved algorithm has a path length of 42332.1m for the shortest total time-consuming scenario and 32924.5m for the least total power-consuming scenario, which is a significant reduction compared to the traditional method.

In addition, the improved NSGA-II algorithm demonstrates high stability and adaptability when dealing with large-scale tasks, and is able to effectively cope with changes in the number of different robots and the amount of tasks. Experiments show that as the number of robots increases, the algorithm is able to reasonably allocate tasks and reduce the load difference of robots, thus optimizing the overall task execution time and energy consumption. Taken together, the algorithm proposed in this paper can not only provide an effective solution for multi-robot task allocation in rail bolt operations, but also has strong generality and can be applied to other multi-robot collaborative tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (grant number 62063013); and the National Natural Science Foundation of China (grant number 62363021); the Construction of Gansu Province Logistics and Transportation Equipment Information Technology Engineering Research Center Platform (Lanzhou Science and Technology Plan Project Project, Project Number 2023-1-16).

References

[1] Wang, Y., Wang, Z., Ma, T., Li, G., & Tie, H. (2022). Research on the realization path of railway intelligent construction based on system engineering. Sustainability, 14(11), 6945.
[2] Qin, Y., Cao, Z., Sun, Y., Kou, L., Zhao, X., Wu, Y., ... & Jia, L. (2023). Research on active safety methodologies for intelligent railway systems. Engineering, 27, 266-279.
[3] Lu, C., Liu, J., Liu, Y., & Liu, Y. (2019). Intelligent construction technology of railway engineering in China. Frontiers of Engineering Management, 6(4), 503-516.
[4] Wu, Y., & Duan, Y. (2018). “Made in China”: building Chinese smart manufacturing image. Journal of Service Science and Management, 11(6), 590-608.
[5] Oregui, M., Li, Z., & Dollevoet, R. J. I. J. (2015). An investigation into the modeling of railway fastening. International Journal of Mechanical Sciences, 92, 1-11.
[6] Bai, T., Yang, J., Xu, G., & Yao, D. (2021). An optimized railway fastener detection method based on modified Faster R-CNN. Measurement, 182, 109742.
[7] Chandran, P., Asber, J., Thiery, F., Odelius, J., & Rantatalo, M. (2021). An investigation of railway fastener detection using image processing and augmented deep learning. Sustainability, 13(21), 12051.
[8] Hu, J., Qiao, P., Lv, H., Yang, L., Ouyang, A., He, Y., & Liu, Y. (2022). High speed railway fastener defect detection by using improved YoLoX-Nano Model. Sensors, 22(21), 8399.
[9] Liu, J., Teng, Y., Shi, B., Ni, X., Xiao, W., Wang, C., & Liu, H. (2021). A hierarchical learning approach for railway fastener detection using imbalanced samples. Measurement, 186, 110240.
[10] Liu, J., Huang, Y., Zou, Q., Tian, M., Wang, S., Zhao, X., ... & Ren, S. (2019). Learning visual similarity for inspecting defective railway fasteners. IEEE Sensors Journal, 19(16), 6844-6857.
[11] Gao, X., Feng, Q., Wang, Z., Liu, L., & Wang, A. (2023). Study on dynamic characteristics and wide temperature range modification of elastic pad of high-speed railway fastener. Engineering Failure Analysis, 151, 107376.
[12] Gao, X., Feng, Q., Wang, A., Sheng, X., & Cheng, G. (2021). Testing research on frequency-dependent characteristics of dynamic stiffness and damping for high-speed railway fastener. Engineering Failure Analysis, 129, 105689.
[13] Xu, J., Zhang, C., Liu, Z., & Pei, Y. (2019). A review on significant technologies related to the robot-guided intelligent bolt assembly under complex or uncertain working conditions. Ieee Access, 7, 136752-136776.
[14] Dharmara, K., Monfared, R. P., Ogun, P. S., & Jackson, M. R. (2018). Robotic assembly of threaded fasteners in a non-structured environment. The International Journal of Advanced Manufacturing Technology, 98, 2093-2107.
[15] Xunan Liu,Yiming Liu,Faquan Li,Yanpeng Zhang,Jianyong Wang & Lianwei Ma. (2025). Optimization of kinematic parameters of continuous miner based on LS-DYNA simulation analysis and NSGA-II algorithm. Results in Engineering,26,105235-105235.
[16] Zhaofan Wu,Yongcun Li,Wentao Zhou & Qiang Fu. (2025). Multi-objective optimization of heat pump drying process using NSGA-II and response surface methodology: a case study of sludge. Case Studies in Thermal Engineering,72,106257-106257.