Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-Robot Task Allocation and Path Planning Based on HIGWO-NSGAII

Xiancheng Fan
Anhui Institute of Information Technology
Yongxia Wang
wyx89wxy@163.com

Anhui Institute of Information Technology
Xinyu Ling
Anhui Institute of Information Technology
zhuo Li
Anhui Institute of Information Technology

Article

Keywords: multi-robot, task allocation, path planning, NSGAII, grey wolf optimizer, Q-learning

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11360778/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Multi-Robot Task Allocation and Path Planning Based on HIGWO-NSGAII

Xiancheng Fan[1],Yongxia Wang[1*],Xinyu Ling[1], Zhuo Li[1]

1 School of Electrical and Electronic Engineering, Anhui Institute of Information Technology, Wuhu 241100, China;2382493123@qq.com

Correspondence: wyx89wxy@163.com

Abstract: Multi-robot collaborative task allocation and path planning are characterized by solution spaces that expand rapidly with problem scale and by increasingly conflicting objectives. Existing swarm intelligence algorithms often suffer from slow convergence and reduced diversity in later stages. To overcome these limitations, this paper proposes a hybrid algorithm, termed HIGWO-NSGAII, which integrates an improved grey wolf optimizer (GWO) with NSGAII. The algorithm first adopts NSGAII as the framework for non-dominated sorting and elitism preservation. A GWO-based gene injection mechanism is then designed for permutation encoding to accelerate convergence. In addition, Q-learning is introduced to adaptively schedule the crossover probability, mutation probability, and GWO guidance ratio, thereby dynamically balancing exploration and exploitation. Finally, a constrained 2-opt local search and an elite archive are employed to improve the quality of the Pareto front. Comparative experiments were conducted against seven algorithms in three grid-based scenarios. The results show that HIGWO-NSGAII achieved the best values in HV, IGD, Spacing, and the number of Pareto solutions. In both the small- and large-scale experiments, the IGD reached 0.0000, and the standard deviations of the key metrics were zero. In the medium-scale experiment, HV improved by 22.3% and IGD decreased by 90.3% relative to the second-best algorithm, with a Wilcoxon test p-value below 0.001. The proposed algorithm exhibits good convergence, stability, and scalability, providing an effective solution for multi-robot collaborative task planning.

Keywords: multi-robot; task allocation; path planning; NSGAII; grey wolf optimizer; Q-learning

1 Introduction

Owing to their advantages in collaborative operation and complementary heterogeneous resources, multi-robot systems have been widely applied in fields such as warehouse logistics, disaster rescue, and intelligent manufacturing[1]. Their collaborative operation revolves around task allocation and path planning. As the numbers of robots and target points increase, the number of allocation schemes grows factorially with the problem scale, exposing traditional exact algorithms to a severe curse of dimensionality. In addition, practical scenarios often require the simultaneous optimization of multiple conflicting objectives, e.g., minimizing the total path length and the maximum single-robot path length, which further complicates the solution process[2]. For path planning, A* is widely adopted to calculate the shortest path cost in grid maps due to its high search efficiency and optimality guarantee.

Nevertheless, A* is designed for single-objective, static environments; it cannot directly handle path conflicts among multiple robots, nor can it simultaneously optimize conflicting objectives such as path length and maximum single-robot path length [3],[4]. Accordingly, this paper uses A* to compute the shortest path cost from each robot to each target point, constructs an A* cost matrix, and embeds it into a multi-objective optimization model to support objective function evaluation in task allocation and subsequent local search.

In terms of task allocation, multi-objective evolutionary algorithms (MOEAs) obtain a set of Pareto-optimal solutions through non-dominated sorting and diversity maintenance, thereby providing decision makers with a rich set of alternatives. NSGAII (non-dominated sorting genetic algorithm II) is one of the most widely used MOEAs and exhibits excellent diversity preservation, yet it suffers from slow convergence and degraded diversity in later stages[5]. The grey wolf optimizer (GWO) mimics the social hierarchy and hunting behavior of grey wolves and features few parameters and fast convergence; however, its elite-guided mechanism readily leads to diversity loss under discrete permutation encoding[6]. To address the diversity loss of GWO, Rane et al.[7] proposed a hybrid algorithm combining NSGAII and GWO, in which the hierarchical leadership and hunting attack mechanisms of GWO are incorporated into the non-dominated sorting and crowding distance framework of NSGAII. This hybrid maintains population diversity via NSGAII while accelerating convergence through GWO, and it effectively improves the available transfer capability in generator participation factor control of renewable-energy-integrated power systems. Nevertheless, such hybrid algorithms are primarily designed for continuous optimization problems. When directly applied to discrete permutation-encoded problems, the continuous position update formula must be adapted through rounding and repair, which may cause information loss and reduced search efficiency[8]. This limitation is not confined to GWO-NSGAII hybrids. At present, swarm intelligence algorithms such as the multi-objective grey wolf optimizer (MOGWO) [9], multi-objective salp swarm algorithm (MSSA) [10], multi-objective particle swarm optimization (MOPSO)[11], multi-objective differential evolution (MODE)[12], and multi-objective multi-verse optimizer (MOMVO)[13] have also been extensively explored in multi-objective optimization. However, their adaptability under discrete permutation encoding remains to be verified. In addition, existing studies have combined Q-learning with evolutionary algorithms for adaptive parameter tuning, such as the Q-learning-based improved NSGA-III[14] and QLMOEA/D[15]. Nevertheless, current Q-learning schemes in hybrid algorithms mostly adopt binary switching or single-parameter tuning, lacking a systematic design for multi-parameter, multi-channel weight allocation.

To address the above issues, this paper proposes a multi-objective hybrid optimization algorithm based on an A* cost matrix and NSGAII/IGWO, termed HIGWO-NSGAII, whose framework is illustrated in Fig. 1. The algorithm devises a GWO gene injection mechanism under permutation encoding, achieving rapid convergence through differentiated injection of the three hierarchical leader wolves (α, β and δ). A constrained 2-opt local search is introduced to refine the Pareto front. An

elite archive is established to preserve historically optimal solutions. A Q-learning-based adaptive parameter scheduling mechanism is designed, which discretizes parameter configurations into six actions and simultaneously schedules the crossover probability, mutation probability, and GWO guidance ratio. To validate the algorithm performance, HIGWO-NSGAII is compared with seven algorithms NSGAII, MOGWO, MSSA, MOPSO, MODE, MOMVO, and GWO-NSGAII on three grid maps of different scales. Each algorithm was independently run 30 times on each map, and the results were evaluated using hypervolume (HV), inverted generational distance (IGD), spacing, and other metrics, with statistical significance verified by the Wilcoxon rank-sum test. Experimental results demonstrate that HIGWO-NSGAII offers advantages in Pareto front coverage, solution set stability, and large-scale scalability.

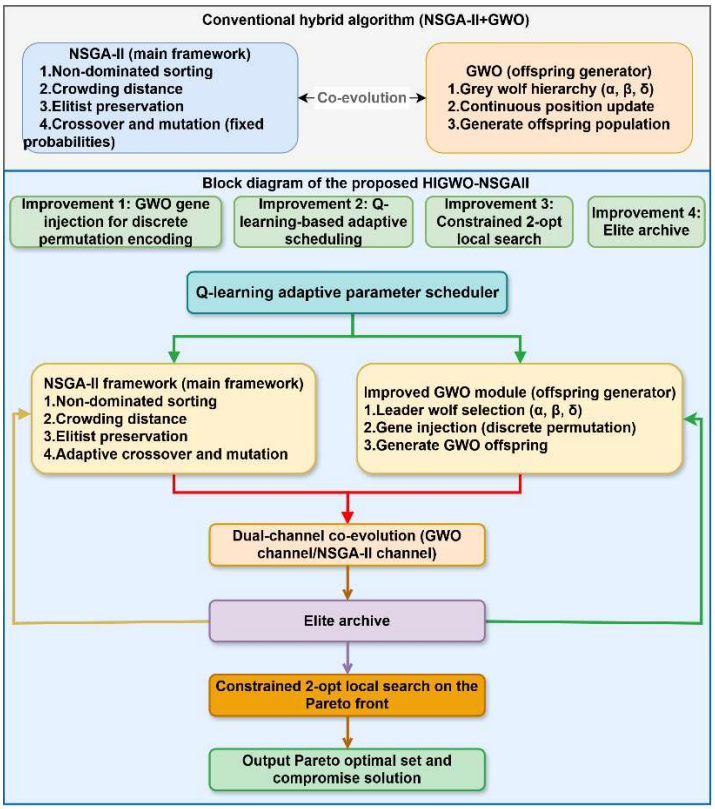


Fig. 1. Conventional vs. proposed algorithm framework

2 Mathematical Modeling

2.1 Two-Dimensional Grid Environment Modeling

The two-dimensional workspace is discretized using a grid-based method. The workspace is divided into $n \times m$ square cells, each identified by coordinates (x, y), where x=1,2,…, n denotes the row index and y=1,2,…, m denotes the column index. A binary matrix $Z(x, y) \in \{0,1\}^{n \times m}$ is used to represent the environment, where $Z(x, y) = 1$ indicates an obstacle cell and $Z(x, y) = 0$ indicates a traversable cell. The robot adopts an 8-neighborhood movement model, i.e., it can move from the current cell to any of the eight surrounding adjacent cells, with an orthogonal move cost of 1 and a diagonal move cost of $\sqrt{2}$ [16]. Obstacles are arranged in a regular block layout, with several fixed

grid obstacle blocks placed in the map. Let K be the total number of obstacle cells; then the obstacle ratio ρ is defined as in Eq. (1).

$$\rho = K / (n \times m) \tag{1}$$

where n is the number of grid rows and m is the number of grid columns.

2.2 Description of Multi-Robot Multi-Objective Task Allocation

Consider a multi-robot system comprising n_R robots and n_T target points. The robot set is denoted as $R = \{r_1, \mathrm{L}\ , r_{n_R}\}$, and the start point of the i-th robot is S_i. The target set is denoted as $T = \{t_1, \mathrm{L}\ t_{n_T}\}$, and the position of the j-th target point is $g_j = (x_j^g, y_j^g)$ [17]. This paper focuses on a one-to-one task allocation scenario, in which each robot is assigned a unique target point and each target point is assigned to only one robot. The task allocation scheme is represented as a permutation, as shown in Eq. (2):

$$\tau = [\tau_1, \tau_2, \mathrm{L}\ , \tau_{n_R}] \tag{2}$$

where $\tau_i \in \{1, \mathrm{L}\ , n_T\}$ denotes the target index assigned to the i-th robot, and satisfies $\tau_i \neq \tau_j, \forall i \neq j$.

2.3 *A* Cost Matrix Construction*

The A* algorithm[18],[19] is employed to compute the shortest path cost from each robot to each target point, and the cost matrix $D \in ¡^{n_R \times n_T}$ is constructed, as shown in Eq. (3).

$$D = \begin{bmatrix} D_{1,1} & D_{1,2} & \mathrm{L} & D_{1,n_T} \\ D_{2,1} & D_{2,2} & \mathrm{L} & D_{2,n_T} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ D_{n_R,1} & D_{n_R,2} & \mathrm{L} & D_{n_R,n_T} \end{bmatrix} \tag{3}$$

where $D_{i,j}$ denotes the shortest path length from the start point s_i of the i-th robot to the j-th target point g_j. To prevent a robot from traversing target points other than its own, the remaining $n_T - 1$ target points are treated as temporary obstacles when computing $D_{i,j}$.

2.4 Multi-Objective Optimization Model

This paper considers two conflicting optimization objectives:

Objective 1: Minimize the total path length, which reflects the overall task execution efficiency of the system, as shown in Eq. (4).

$$f_1(\tau) = \sum_{i=1}^{n_R} D_{i,\tau_i} \tag{4}$$

Objective 2: Minimize the maximum single-robot path length, which reflects the balance of task allocation, as shown in Eq. (5).

$$f_2(\tau) = \max_{i=1}^{n_R} D_{i,\tau_i} \tag{5}$$

These two objectives are inherently conflicting: reducing f_1 usually causes the path of some robots to become extremely long, thereby increasing f_2, and vice versa. Consequently, no unique optimal solution exists, and a set of Pareto-optimal solutions must be sought.

3 HIGWO-NSGAII Algorithm

3.1 Algorithm Fundamentals

NSGAII [20],[21],[22] is one of the most widely used algorithms in the field of multi-objective optimization. Its core components include fast non-dominated sorting, crowding distance, and an elitist preservation strategy, which together endow it with excellent diversity maintenance capability. Nevertheless, it suffers from drawbacks such as slow convergence and degraded diversity in later stages. GWO mimics the social hierarchy and hunting behavior of grey wolf packs. It involves few parameters and converges rapidly, but its elitist guidance is excessively strong, which readily causes diversity loss under discrete permutation encoding; moreover, its continuous position update formula cannot be directly applied to permutation encoding [23]. In addition, continuous swarm intelligence algorithms are difficult to apply directly under discrete permutation encoding. Simple discretization tends to induce diversity loss and premature convergence, thereby markedly limiting search efficiency [24],[25]. Therefore, a hybrid algorithm is required that can preserve the diversity advantage of NSGAII, exploit the fast convergence capability of GWO, and simultaneously accommodate discrete permutation encoding.

3.2 Algorithm Improvements

3.2.1 GWO Gene Injection Mechanism

To accommodate the discrete nature of permutation encoding, a GWO discretization mechanism based on leader-wolf gene injection is designed. In each generation, three leader wolves are selected from the elite archive: α is the individual with the largest crowding distance on the Pareto front in the archive, β is the suboptimal individual, and δ is randomly selected from the Pareto front with a probability of 50%. Let the current solution be x, and let the convergence factor be $a = 2(1 - t / T_{\max})$. The injection quantities are given by Eq. (6):

$$k_1 = \max(1, [\frac{2-a}{2} \mathrm{g} \frac{n_R}{2}]), k_2 = \max(1, [\frac{2-a}{2} \mathrm{g} \frac{n_R}{4}]), k_3 = \max(1, [\frac{2-a}{2} \mathrm{g} \frac{n_R}{4}]) \tag{6}$$

where n_R is the number of robots, t is the current iteration number, and $T_{\max}$ is the maximum number of iterations. For the α, β and δ wolves, k_1, k_2 and k_3 positions are randomly selected, respectively, and the corresponding gene values are injected into the same positions of x. After injection, a repair mechanism is applied to ensure that x remains a valid permutation. Specifically, the duplicated gene positions in x are identified, unused gene values borrowed from the original solution are used to fill these positions, and any remaining vacancies are randomly supplemented from the feasible gene set. The repaired x is then taken as the offspring individual generated by the GWO channel.

3.2.2 Q-learning-Based Adaptive Parameter Scheduling

在 In evolutionary algorithms, parameters such as the crossover probability p_c and the mutation probability p_m are usually set to fixed values or decreased linearly based on empirical rules. Such static strategies are difficult to adapt to changes in the population state during the search, and can easily lead to an imbalance between exploration and exploitation [26],[27],[28]. To this end, Q-learning is introduced as an adaptive parameter scheduler. Before reproduction in each generation, it dynamically selects a parameter combination according to the population state and outputs the crossover probability, the mutation probability, and the GWO guidance ratio r_{gwo}, which are then provided to GWO and NSGAII for selection and use, respectively.

In this paper, the population state is encoded into an 18-dimensional state space, which is composed of three dimensions: population diversity d_t, stagnation degree c_t, and iteration progress p_t. Specifically, the diversity d_t is divided into three levels-low, medium, and high-with the level index denoted as l_d. The stagnation degree c_t is divided into two levels-normal and stagnant-with the level index denoted as l_c. The iteration progress p_t is divided into three levels-early, middle, and late-with the level index denoted as l_p. The level combinations of the three dimensions yield a total of 3×2×3=18 states. The state index S_t is given in Eq. (7).

$$S_t = (l_d - 1) \times 6 + (l_c - 1) \times 3 + l_p \tag{7}$$

where S_t is an integer ranging from 1 to 18. Each state corresponds uniquely to a combination of diversity, stagnation degree, and iteration progress. The formulas for calculating the levels of diversity, stagnation, and progress are given in Eqs. (8), (9), and (10), respectively.

$$l_d = \begin{cases} 1, d_t < 0.3 \\ 2, 0.3 \le d_t \le 0.6 \\ 3, d_t > 0.6 \end{cases} \tag{8}$$

$$l_c = \begin{cases} 1, c_t < 15 \\ 2, c_t \ge 15 \end{cases} \tag{9}$$

$$l_p = \begin{cases} 1, p_t < 0.33 \\ 2, 0.33 \le p_t \le 0.67 \\ 3, p_t > 0.67 \end{cases} \tag{10}$$

Here, p_t denotes the iteration progress, defined as the ratio of t to $T_{\max}$, where t is the iteration count and $T_{\max}$ is the maximum number of iterations. The diversity d_t is measured by the average Hamming distance among individuals [29],[30]. The stagnation counter c_t records the number of consecutive generations in which the best f_1 on the current Pareto front has not improved. When c_t is greater than or equal to 25, the search is considered stagnant, and a strong perturbation is triggered on the offspring.

According to the state index, an action space A comprising six parameter combinations is defined, corresponding to strong exploration, weak exploration, strong exploitation, GWO acceleration, NSGAII refinement, and alert mode. Q-learning does not directly output continuous parameters; instead, it discretizes the parameter configuration into six actions. In each generation, one action is selected according to the current state, and the specific parameter values are then obtained through a mapping table. Each action is mapped to a tuple (p_c, p_m, r_{gwo}), with the specific values listed in Table 1. After the action a_t is selected, these three parameters are passed to the reproduction process to control the triggering probabilities of crossover, mutation, and GWO-guided crossover, respectively. This enables the algorithm to dynamically balance exploration and exploitation according to the search stage.

Table 1 Action space

Action a_t	p_c	p_m	r_{gwo}	Application scenario
1	0.95	0.20	0.50	Strong exploration
2	0.85	0.15	0.30	Weak exploration
3	0.70	0.08	0.10	Strong exploitation
4	0.90	0.12	0.40	GWO acceleration
5	0.75	0.10	0.20	NSGAII refinement
6	0.60	0.05	0.05	Alert mode

Action selection adopts the *ε-greedy* strategy [31]. When a random number is smaller than ε_t, an action is selected uniformly at random from the action space A; when the random number is greater than or equal to ε_t, the action with the maximum current Q value is selected according to Eq. (11). The exploration rate $\varepsilon_t = 0.15(1-0.5p_t)$ decreases linearly with the iteration progress p_t, so that the algorithm favors exploration in the early stage and exploitation in the later stage.

$$a_t = \arg\max_{a \in A} Q(S_t, a) \tag{11}$$

To evaluate the quality of actions, the reward function adopts a time-aware, bi-objective weighted form that accounts for the relative improvement of both the hypervolume (HV) and the total path length f_1. In the early stage, the algorithm is designed to prioritize expanding the coverage of the Pareto front, and thus the reward places greater emphasis on HV improvement. In the later stage, convergence to the optimal solution becomes more critical, so the reward shifts its emphasis toward f_1 improvement. The reward function is given in Eq. (12).

$$R_t = 100[w_{HV} \frac{HV_t - HV_{t-1}}{HV_{t-1}} + w_{f_1} \frac{f_{1,t-1}^{best} - f_{1,t}^{best}}{f_{1,t-1}^{best}}] \tag{12}$$

Here, HV_t denotes the hypervolume of the Pareto front at the t-th generation, and $f_{1,t}^{best}$ denotes the minimum total path length of the Pareto front at the t-th generation. The weights vary smoothly with the iteration progress, as expressed in Eq. (13).

$$w_{HV} = 1-0.5p_t, w_{f_1} = 0.5+0.5p_t \tag{13}$$

The Q-table records the cumulative expected return of taking action a_t in state S_t, and its update rule is given in Eq. (14).

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \eta_t[R_t + \gamma \max_{a\in A} Q(S_{t+1},a) - Q(s_t,a_t)] \tag{14}$$

Here, the learning rate $\eta_t = \eta_0(1-0.7p_t)$ decays with iterations, with an initial learning rate $\eta_0 = 0.15$; the discount factor $\gamma = 0.9$; and S_{t+1} is the state index of the next-generation population after executing action a_t and completing one generation of reproduction.

After executing an action in each generation, the Q-value is updated according to the reward and the next state, so that the Q-table gradually approaches the true state-action value. Through this mechanism, Q-learning can select the optimal parameter combination online according to the population state, thereby dynamically balancing exploration and exploitation during the search and improving the overall performance of the algorithm.

3.2.3 2-opt Local Search

To enhance the local refinement capability of the Pareto front, a constrained directional 2-opt [32] is introduced for individuals on the Pareto front in the later evolutionary stage. For a permutation solution τ, two positions a and b ($a<b$) are randomly selected, and the corresponding genes are swapped to obtain a new solution τ'. Incremental evaluation is adopted, so that only the cost changes corresponding to the two swapped genes are calculated.

Let the cost vector of the original solution τ be c, whose k-th component is D_{k,τ_k}. After swapping positions a and b, the new cost is given by Eq. (15).

$$c'_a = D_{a,\tau_b}, c'_b = D_{b,\tau_a} \tag{15}$$

The variations in the total path length and the maximum single-robot path length are given in Eqs. (16) and (17).

$$\Delta f_1 = c'_a + c'_b - c_a - c_b \tag{16}$$

$$\Delta f_2 = \begin{cases} \max(c') - f_2(\tau), 若 c_a = f_2(\tau) 或 c_b = f_2(\tau) \\ \max(0, \max(c'_a, c'_b) - f_2(\tau)), else \end{cases} \tag{17}$$

To balance the two objectives, a weighted scoring criterion is adopted, as shown in Eq. (18).

$$score(\tau) = wf_1(\tau) + (1-w)f_2(\tau) \tag{18}$$

Here, $w \in [0,1]$ is a weight randomly generated for each local search. If the swap satisfies $score(\tau') < score(\tau) - 10^{-9}$, the swap is accepted and the permutation and cost vector are updated; otherwise, it is rejected. Since w is randomly generated, the algorithm can achieve precise local search in different regions of the Pareto front.

3.2.4 **Elite Archive**

The elite archive is maintained independently of the evolutionary population. In each generation,

the Pareto front solutions of the current population are merged with the archive. After non-dominated sorting and deduplication, if the archive size exceeds the upper limit of 2N, individuals are retained in descending order of crowding distance. The archive serves as the source for selecting leader wolves, participates in the synthesis of the final Pareto front, and acts as a backup for the historical best solutions. The archive upper limit is given in Eq. (19).

$$|Archive| \le A_{\max} = 2N \tag{19}$$

Where $|Archive|$ denotes the archive size; $A_{\max}=2N$ is the upper limit of the archive; and N is the population size. The archive provides the source for leader wolf selection, participates in the synthesis of the final Pareto front, and serves as a backup for the historical best solutions.

3.2.5 **Overall Algorithm Framework**

HIGWO-NSGAII adopts a dual-channel cooperative evolution architecture. It uses NSGAII as the framework for non-dominated sorting and elitist preservation, incorporates the improved GWO gene injection mechanism, and is supplemented by Q-learning-based adaptive parameter scheduling, 2-opt local search, and an elite archive. The algorithm proceeds as follows. After population initialization, non-dominated sorting and crowding distance calculation are performed in each generation, and the elite archive is updated. Subsequently, Q-learning outputs the crossover probability p_c, the mutation probability p_m, and the GWO guidance ratio r_{gwo} according to the population state. The GWO channel or the NSGAII channel is then selected for offspring reproduction based on the iteration progress and diversity. The offspring and parents are merged for environmental selection to form the next generation. Finally, the Pareto front is extracted from the population and the archive, and the solution on the normalized Pareto front closest to the ideal point is taken as the compromise solution [33].

Algorithm 1: HIGWO-NSGAII Main Procedure

Input: Cost matrix D, population size N, maximum iterations $\boldsymbol{T}_{\max}$

Output: Pareto optimal solution set and compromise solution

1. Initialize population *pop*, elite archive *Archive*, and Q-learning scheduler *QL*
2. for $t = 1$ to $\boldsymbol{T}_{\max}$ do
3. Calculate objective values of *pop*, perform non-dominated sorting and crowding distance calculation
4. Update elite archive *Archive*
5. Calculate diversity d_t, stagnation counter c_t, and iteration progress p_t
6. *QL* outputs p_c, p_m, r_{gwo} according to population state
7. Select leader wolves α, β, δ from Archive
8. Select parent pool *pool* using binary tournament selection
9. if $p_t <= 0.3$ or $d_t < 0.3$ then
10. Offspring: GWO channel $(pool, \alpha, \beta, \delta, p_c, p_m)$
11. else

12. Offspring: NSGAII channel $(pool, \alpha, \beta, \delta, p_c, p_m, r_{gwo})$

13. end if

14. if c_t >= 25 then perform strong perturbation on offspring

15. Evaluate offspring objective values

16. if p_t > 0.4 then perform 2-opt on Pareto front individuals and re-evaluate

17. Merge parents and offspring, environmental selection to generate next generation *pop*

18. Calculate and update Q-table

19. end for

20. Merge final population with *Archive*, extract Pareto front and select compromise solution

21. return Pareto optimal solution set and compromise solution

4 Experimental Setup and Result Analysis

4.1 Experimental Setup

All experiments in this paper were conducted in the MATLAB R2023b programming environment. To ensure a fair comparison, the population size and maximum number of iterations of all compared algorithms were kept consistent with those of the proposed algorithm. The population size N was set to 60. The maximum number of iterations $T_{\max}$ was set to 200 for the small- and medium-scale scenarios and 300 for the large-scale scenario. Each algorithm was run independently 30 times. The compared algorithms included NSGAII, MOGWO, MSSA, MOPSO, MODE, MOMVO, GWO-NSGAII, and HIGWO-NSGAII. The evaluation metrics adopted were HV, IGD, spacing, and the number of Pareto solutions [34],[35],[36]. The configurations of the three path planning experiments are listed in Table 2, and the complete configurations are provided in Supplementary Tables S1, S5, and S9, respectively.

Table 2 Configurations of the three experimental groups

Experimental group	Grid size	Obstacle ratio ρ	Number of robots n_R	Number of targets n_T	Maximum iterations $T_{\max}$
Small-scale	30×30	4%	12	12	200
Medium-scale	**50×50**	**20%**	**20**	**20**	**200**
Large-scale	80×80	25%	30	30	300

。In all experiments, obstacles are arranged in a regular block layout, and the start points and target points use fixed coordinates rather than being generated randomly, thereby ensuring that the results are fully reproducible.

4.2 Comparison of Small and Medium-Scale Experiments

The small- and medium-scale experiments correspond to scenarios with 12 robots/12 targets and 20 robots/20 targets, respectively. Table 3 summarizes the mean values of the metrics obtained by the eight algorithms over 30 independent runs under both scales. The complete mean ± standard deviation

results are provided in Supplementary Tables S2 and S6, and the raw run data are given in Supplementary Tables S4 and S8.

Table 3 Summary of the main metrics for the two scales

Scale	Algorithm	f_1	f_2	HV	IGD	Spacing	Number of Pareto solutions
Small-scale	NSGAII	153.43	23.08	701.4	0.5089	1.2600	5.23
	MOGWO	160.22	23.63	512.4	2.8412	1.0322	3.13
	MSSA	161.65	23.24	410.6	3.9959	0.0000	1.03
	MOPSO	156.22	21.41	637.6	1.4523	1.1350	3.70
	MODE	155.29	20.52	689.1	1.2559	1.5540	3.63
	MOMVO	151.59	30.01	203.2	3.4072	0.0000	59.00
	GWO-NSGAII	153.22	23.40	714.2	0.5424	1.4137	5.40
	HIGWO-NSGAII	153.81	**21.49**	**753.1**	**0.0000**	**0.5450**	**6.00**
Medium-Scale	NSGAII	215.04	23.14	6054.8	11.1493	0.7548	2.37
	MOGWO	301.56	31.75	2564.9	61.0725	2.0942	1.83
	MSSA	305.15	30.19	2559.3	63.9884	0.0000	1.07
	MOPSO	254.89	26.37	4363.7	32.6671	3.8639	2.20
	MODE	242.45	24.96	5081.4	26.4073	2.1079	1.80
	MOMVO	244.14	31.66	3113.9	29.6822	0.0000	59.00
	GWO-NSGAII	211.55	22.91	6461.1	9.9076	1.1875	2.30
	HIGWO-NSGAII	**194.19**	**17.57**	**7904.8**	**0.9559**	**0.1747**	2.20

In the small-scale experiment, HIGWO-NSGAII performed best across all metrics. Its HV reached 753.1, about 5.4% higher than that of the second-best algorithm GWO-NSGAII (714.2), with a standard deviation of 0; the IGD was 0.0000, indicating that the obtained Pareto front completely coincided with the reference front; and the number of Pareto solutions was 6.00, exactly consistent with the reference front size, achieving 100% coverage. In contrast, although MSSA and MOMVO had a Spacing of 0, their numbers of solutions were only 1.03 and 59.00, respectively, indicating that the solutions completely overlapped or that there was almost only one effective solution, so the actual diversity was zero.

In the medium-scale experiment, the advantage of HIGWO-NSGAII was further expanded. Its HV reached 7904.8, about 22.3% higher than that of the second-best algorithm GWO-NSGAII (6461.1); the IGD was only 0.9559, far lower than the 9.9076 of GWO-NSGAII, a reduction of 90.3%; the Spacing was 0.1747, indicating the most uniform distribution. The mean value of f1 was 194.19 with a

standard deviation of only 2.13, both being optimal. The above results indicate that HIGWO-NSGAII achieves the best overall performance in both small- and medium-scale scenarios. The box plots of HV and IGD are provided in Supplementary Figures S1 and S4, respectively.

4.3 Large-Scale Experiment and Scalability Analysis

The large-scale experiment used an 80×80 grid, a 25% obstacle ratio, 30 robots, and 30 targets, with a maximum number of iterations of 300 and a reference Pareto front size of 1. Table 4 presents the mean values of the main metrics of the eight algorithms in the large-scale experiment. The complete statistics are provided in Supplementary Table S10, and the raw run data are given in Supplementary Table S12.

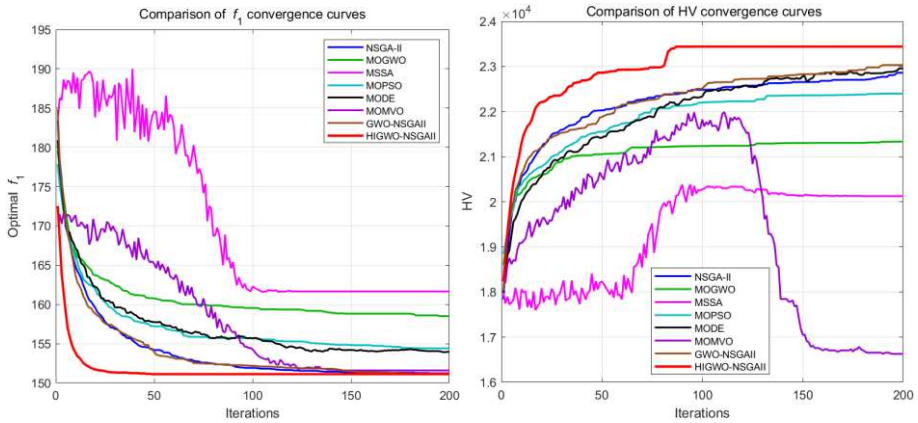
Table 4 Summary of the main metrics in the large-scale experiment

Algorithm	f_1	f_2	HV	IGD	Spacing	Number of Pareto solutions
NSGAII	358.54	26.78	53158.3	61.3311	0.0000	1.10
MOGWO	727.95	55.77	16767.1	426.4693	1.9782	1.57
MSSA	696.02	54.75	17722.5	398.6296	0.0000	1.03
MOPSO	649.44	56.05	22577.9	329.6901	12.8187	2.80
MODE	622.64	53.53	24048.4	317.3079	6.4762	2.10
MOMVO	406.97	51.97	32492.6	115.2430	0.0000	59.03
GWO-NSGAII	352.70	27.62	53817.9	56.3896	0.0000	1.23
HIGWO-NSGAII	**300.00**	**10.00**	**70038.5**	**0.0000**	**0.0000**	**1.00**

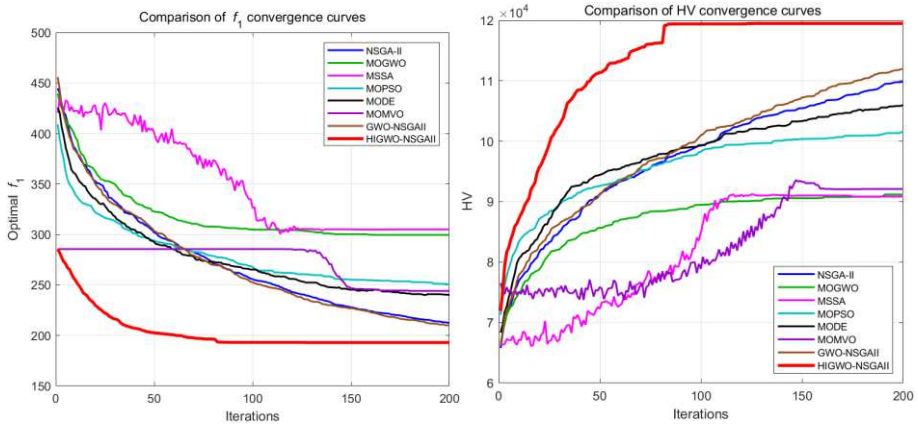
Experimental results show that HIGWO-NSGAII still maintained an overall lead. Its HV reached 70038.5, about 30.1% higher than that of the second-best algorithm GWO-NSGAII (53817.9). The IGD was 0.0000, indicating that the obtained Pareto front completely coincided with the reference front. The mean f1 was 300.00 and the mean f2 was 10.00, both uniquely optimal, with standard deviations of 0 over 30 independent runs. The reference Pareto front size was 1, indicating that the solutions found by HIGWO-NSGAII completely dominated those found by all other algorithms and constituted the only non-dominated solution. From 12 robots to 30 robots, HIGWO-NSGAII consistently maintained optimal performance, verifying its good scalability. In contrast, the continuous swarm intelligence algorithms MOMVO and MSSA exhibited significantly degraded path quality in the large-scale scenario, further demonstrating the effectiveness of the GWO gene injection mechanism under discrete permutation encoding.

4.4 Convergence and Stability Analysis

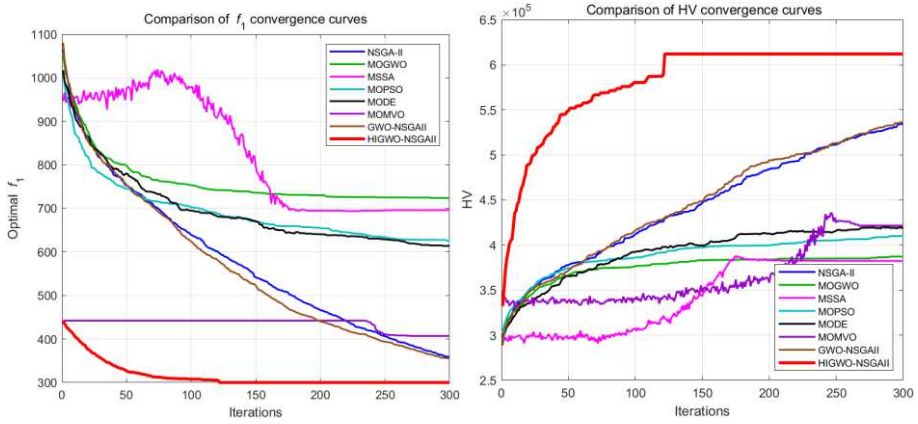
The convergence curves of f_1 and HV for the eight algorithms in the three experimental groups are shown in Fig. 2, averaged over 30 runs, corresponding to Supplementary Figures S2, S5, and S8, respectively.



(a) Small-scale convergence curves



(b) Medium-scale convergence curves



(c) Large-scale convergence curves

Fig. 2 Convergence curves of f_1 and HV for the three experimental groups

From the convergence curves, it is evident that HIGWO-NSGAII leads comprehensively in convergence speed, convergence accuracy, and stability: its f_1 and HV curves reach the optimum and level off within far fewer iterations than the compared algorithms, and the curves for 30 independent runs under the small- and large-scale scenarios completely overlap, with only negligible fluctuations under the medium-scale scenario. NSGAII and GWO-NSGAII not only converge more slowly but also yield markedly worse final values, with

varying degrees of oscillation in their curves. MOMVO and MSSA, by contrast, fall into stagnation early, exhibit large curve fluctuations, and their HV remains low throughout, reflecting the loss of diversity of these two continuous swarm intelligence algorithms under discrete permutation encoding.

In terms of stability, the standard deviations of f_1、f_2, HV, IGD, and Spacing for HIGWO-NSGAII were all 0 in the small- and large-scale experiments, and the outputs of 30 independent runs were completely identical. In the medium-scale experiment, its standard deviations were only 2.13 for f_1, 0.62 for f_2, and 181.86 for HV, all far smaller than those of the other algorithms. For example, under the medium-scale scenario, the HV standard deviation was 920.95 for NSGAII and 796.23 for GWO-NSGAII; under the large-scale scenario, the f_1 standard deviation reached 62.90 for MOGWO and 117.01 for MOPSO. These results indicate that HIGWO-NSGAII not only achieves high convergence accuracy but also possesses strong stability and engineering reliability.

4.5 Wilcoxon Rank-Sum Test

To verify the statistical significance of the differences among the algorithms, this paper employs the Wilcoxon rank-sum test to compare the HV metric of the eight algorithms, with a significance level of a=0.05. The key comparison results are shown in Table 5, and the complete pairwise comparisons are provided in Supplementary Tables S3, S7, and S11.

Table 5 Summary of Wilcoxon rank-sum test results

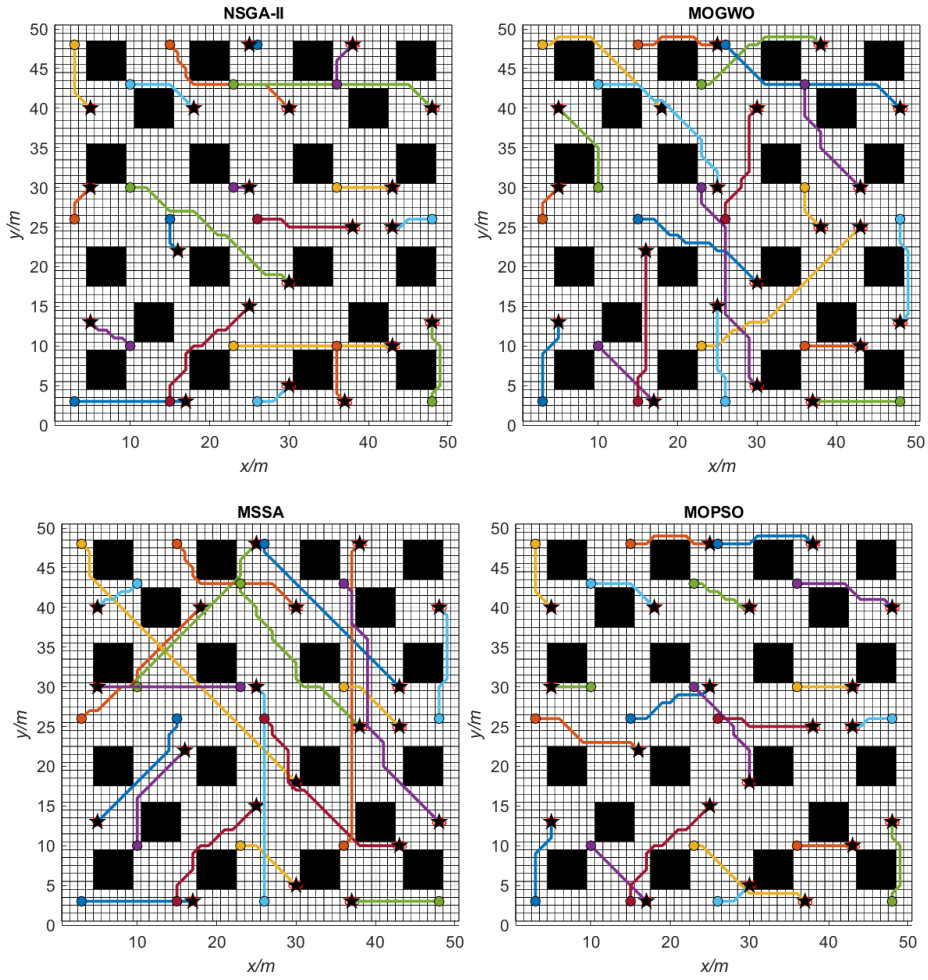
Comparison	*p*-value (small-scale)	*p*-value (medium-scale)	*p*-value (large-scale)
HIGWO-NSGAII vs NSGAII	< 0.001	< 0.001	< 0.001
HIGWO-NSGAII vs GWO-NSGAII	< 0.001	< 0.001	< 0.001
HIGWO-NSGAII vs MOGWO	< 0.001	< 0.001	< 0.001
HIGWO-NSGAII vs MSSA	< 0.001	< 0.001	< 0.001
HIGWO-NSGAII vs MOPSO	< 0.001	< 0.001	< 0.001
HIGWO-NSGAII vs MODE	< 0.001	< 0.001	< 0.001
HIGWO-NSGAII vs MOMVO	< 0.001	< 0.001	< 0.001
GWO-NSGAII vs NSGAII	0.7958	0.0963	0.4011

From the statistical test results, the *p*-values between HIGWO-NSGAII and all compared algorithms were less than 0.001, confirming the significant superiority of the proposed algorithm in terms of HV. The *p*-values between GWO-NSGAII and NSGAII in the three experimental groups were 0.7958, 0.0963, and 0.4011, respectively, none of which were significant. This indicates that the simple hybridization of classical GWO and NSGAII yields limited improvement, thereby validating the effectiveness of the proposed improvement strategies.

4.6 Path Planning Simulation Results

To intuitively demonstrate the performance of each algorithm in multi-robot task allocation and path planning, this section presents partial path planning simulation results using the medium-scale experiment as an example, as shown in Fig. 3. In the figure, black block regions represent obstacles,

white areas denote traversable regions, circular markers indicate robot start points, five-pointed star markers represent target points, and polylines in different colors depict the actual planned paths of each robot from its start point to its assigned target point. Each algorithm was run independently 30 times, and the figure shows the path planning results from the last run. The complete path planning results for the small, medium, and large-scale experiments are provided in Supplementary Figures S3, S6, and S9, respectively. The f_1 and f_2 results of the last planning run for each scale and each algorithm are listed in Table 6.



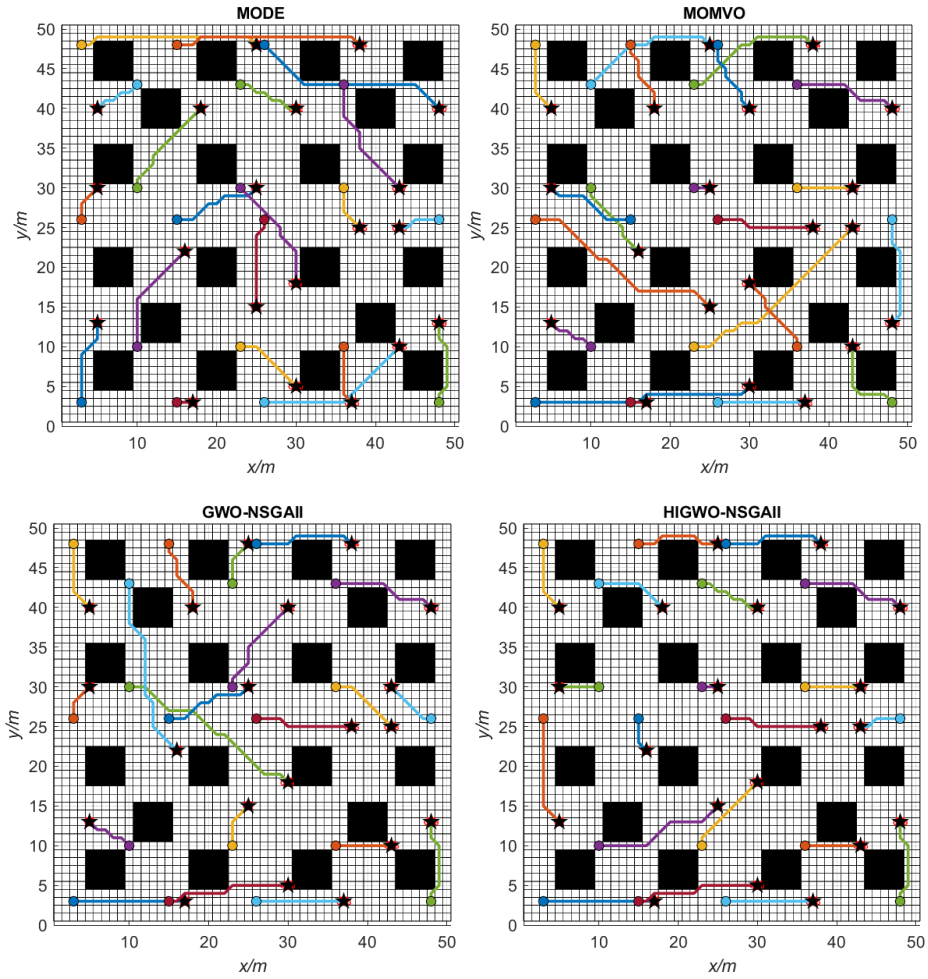


Fig. 3 Path planning simulation results of the medium-scale experiment

Table 6 Path planning results of each algorithm in the experiments of different scales

Experimental scale	Metric	NSGAII	MOGWO	MSSA	MOPSO	MODE	MOMVO	GWO-NSGAII	HIGWO-NSGAII
Small-scale	f_1	151.47	165.95	166.05	160.40	154.30	149.95	150.54	152.64
	f_2	23.31	22.31	22.07	20.66	22.31	26.63	24.97	20.90
Medium-scale	f_1	212.30	307.01	415.62	211.85	244.24	255.31	226.68	199.91
	f_2	26.24	27.90	41.18	17.31	25.31	27.83	24.97	17.07
Large-scale	f_1	370.71	785.14	643.43	580.05	601.83	411.54	328.28	300.00
	f_2	24.14	58.28	58.28	44.14	44.14	44.14	24.14	10.00

Taken together, Table 6 and Fig. 3 show that HIGWO-NSGAII achieves the best values for both the total path length and the maximum single-robot path length in the medium-scale scenario, namely 199.91 and 17.07, which are about 5.7% and 1.4% lower than those of the second-best algorithm, respectively. In contrast, algorithms such as MSSA and MOGWO produce severe detours, and the paths of some robots deviate noticeably from the shortest direction, leading to significantly increased total distance and maximum single-robot distance. In the small-scale scenario, the differences among the algorithms are relatively small. The total path length of HIGWO-NSGAII is slightly larger than those

of MOMVO and GWO-NSGAII, but its maximum single-robot path length is 20.90, markedly lower than 26.63 for MOMVO and 24.97 for GWO-NSGAII, indicating better load balancing. In the large-scale scenario, HIGWO-NSGAII yields a total path length of 300.00 and a maximum single-robot path length of 10.00, far outperforming the other algorithms, with no path conflicts or redundant detours. As the problem scale increases from 12 robots to 30 robots, HIGWO-NSGAII consistently maintains optimal path quality, which verifies its good scalability and engineering practicality. By comparison, the continuous swarm intelligence algorithms MOMVO and MSSA exhibit markedly degraded path quality in the large-scale scenario, further demonstrating the effectiveness of the GWO gene injection mechanism under discrete permutation encoding.

4.7 Discussion

Taken together, the results of the three experimental groups show that HIGWO-NSGAII performs best across all metrics. In the small- and large-scale experiments, the IGD was 0.0000, and the Pareto front completely coincided with the reference front; in the medium-scale experiment, the IGD was only 0.9559, far lower than that of the other algorithms. The HV ranked first in all three experimental groups, improving by 5.4%, 22.3%, and 30.1% over the second-best algorithm, respectively. In terms of stability, the standard deviations of all key metrics were 0 in the small- and large-scale experiments, and were also far smaller than those of the other algorithms in the medium-scale experiment. The mean values of f_1 and f_2 were likewise optimal, and HIGWO-NSGAII maintained a consistent lead from 12 robots to 30 robots, demonstrating good scalability. At the same time, the continuous swarm intelligence algorithms exposed severe deficiencies under discrete permutation encoding. For MOMVO, all 59 solutions overlapped, the Spacing was 0, and only one solution was actually effective; MSSA found only about one solution on average; and the numbers of solutions for MOPSO and MOGWO were 2.20-3.70 and 1.57-3.13, respectively, far lower than those of the hybrid algorithms. A further comparison between classical GWO and the improved GWO reveals that GWO-NSGAII showed no statistically significant difference from NSGAII, whereas HIGWO-NSGAII significantly outperformed all algorithms. Taking the medium-scale case as an example, the HV increased from 6461.1 to 7904.8, a rise of 22.3%; the IGD decreased from 9.9076 to 0.9559, a reduction of 90.3%; the Spacing decreased from 1.1875 to 0.1747, a reduction of 85.3%; f1 decreased from 211.55 to 194.19, a reduction of 8.2%; and the HV standard deviation decreased from 796.23 to 181.86, a reduction of 77.2%. These results further demonstrate the effectiveness of the improved algorithm.

5 Conclusions

To address the strongly coupled and multi-objective problem of multi-robot task allocation and path planning in two-dimensional grid environments, this paper proposes a hybrid algorithm, HIGWO-NSGAII, which integrates an improved grey wolf optimizer with NSGAII. Through multi-scenario simulation experiments and statistical tests, the following conclusions are drawn.

1.A hybrid optimization framework was constructed with NSGAII serving as the framework for non-dominated sorting and elitist preservation. A GWO gene injection mechanism was designed

specifically for the permutation encoding characteristics of task allocation and path planning, enabling the improved grey wolf optimizer to act directly on the discrete permutation space. This achieves rapid convergence while maintaining the quality of the Pareto solution set.

2.Q-learning was introduced to adaptively schedule the crossover probability, mutation probability, and GWO guidance ratio, allowing the algorithm to dynamically balance global exploration and local exploitation according to the evolutionary state. By combining a constrained 2-opt local search with an elite archive, the Pareto front is refined and historical best solutions are maintained, thereby improving both convergence and diversity.

3.Systematic experiments on multiple robots and maps of different scales show that HIGWO-NSGAII outperforms the compared algorithms in all four metrics: HV, IGD, Spacing, and the number of Pareto solutions. In the small- and large-scale experiments, the IGD reached 0.0000 and the standard deviations of the key metrics were 0, indicating a stable ability to approximate the optimal front. In the medium-scale experiment, HV improved by 22.3% and IGD decreased by 90.3% relative to the second-best algorithm, with a Wilcoxon rank-sum test p-value below 0.001, confirming that the performance improvement is statistically significant.

4.In the comparative experiments, the continuous swarm intelligence algorithms MOMVO and MSSA exhibited pronounced diversity loss under discrete permutation encoding, whereas the proposed GWO gene injection mechanism effectively adapts to discrete permutation encoding, maintains population diversity, and avoids premature convergence. These results indicate that HIGWO-NSGAII provides an effective solution for multi-robot collaborative task planning that balances convergence, diversity, and scalability, and offers a useful reference for multi-robot scheduling in fields such as warehouse logistics, disaster rescue, and intelligent manufacturing.

This study still has certain limitations: only static obstacle environments were considered, and validation on a real robot platform was not performed. Future research will be conducted in the following directions: 1) extending the problem to dynamic obstacle and dynamic task scenarios and investigating online replanning and robust optimization methods; 2) introducing deep reinforcement learning to automatically encode high-dimensional states, thereby enhancing adaptive decision-making capability in complex dynamic environments; and 3) conducting experimental tests on actual multi-robot platforms to verify the engineering applicability of the proposed method.

Author Contributions: Conceptualization, X.C.F. and Y.X.W.; methodology, X.C.F.; software, X.C.F.; validation, X.C.F., X.Y.L. and Z.L.; formal analysis, X.C.F.; investigation, X.C.F., X.Y.L. and Z.L.; resources, Y.X.W.; data curation, X.C.F.; writing-original draft preparation, X.C.F.; writing-review and editing, X.C.F. and Y.X.W.; visualization, X.C.F. and X.Y.L.; supervision, Y.X.W.; project administration, Y.X.W.; funding acquisition, Y.X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Research Project of Higher Education Institutions of Anhui Province (Grant No. 2024AH050637), the Wuhu Science and Technology Program (Grant No.

2025kj047), and the Youth Scientific Research Fund Project of Anhui Institute of Information Technology (Grant No. 24QNJJKJ012).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Data Availability Statement

The authors declare that the data supporting the findings of this study are available within the paper and its supplementary information files. This statement includes all original data generated in this study and any secondary data reused to support the results and analyses.

References

[1] ZHOU X T,HUANG B,ZHAO S Y.A survey of multi-robot task planning methods[J].*Information and Control*, 2026, 55(2):188-208.

[2] MAHMUD S A, IBRAHIM A M, KHAN M R,et al. A comprehensive review on multi robot task assignment and path planning: Operational constraints, challenges, and potentials of deep reinforcement learning [J].Applied Soft Computing, 2026,174:114498.

[3] [3] REN Z Q, CARLOS H, MAXIM L,et al.EMOA*: A framework for search-based multi-objective path planning[J]. Artificial Intelligence. 339,104260 (2025).

[4] REN Z Q,SIVAKUMAR,,HOWIE C,et al.Subdimensional Expansion for Multi-Objective Multi-Agent Path Finding[J]. IEEE Robotics and Automation Letters.6,7153-7160(2021).

[5] QUAN E Z, LIU Y J,GAO H,et al.An Enhanced NSGA-II Algorithm Combining Lévy Flight and Simulated Annealing and Its Application in Electric Winch Trajectory Planning: A Complex Multi-Objective Optimization Study [J].Machines.13,1017(2025).

[6] CHEN S L, ZHENG J G. Discrete Gray Wolf Optimizer for Solving Distributed Permutation Flowshop Scheduling Problem[J].Concurrency and Computation: Practice and Experience.37,e70090(2025).

[7] P.R.RANE, P. M. MANKAR, H. S. KULAT,et al. Hybrid NSGA-II and Grey Wolf Optimization-Based Generator Participation Factor Control for ATC Enhancement in Renewable Integrated Power Systems[J]. Advances in Nonlinear Variational Inequalities.28,534-547(2025).

[8] CHEN X C, LI C K,CHOU F,et al. Multiobjective optimization of collaborative robotic task sequence assignment problems under collision-free constraints[J].Advances in Mechanical Engineering.16,1-17(2024).

[9] GUO W Y, LI S L,Dai F,et al. A hybrid-strategy multi-objective grey wolf optimizer based on diversity ranking[J].Cluster Computing.28,754(2025).

[10] ALI E R,SEYEDALI M. Multi robot exploration using an advanced multi-objective salp swarm algorithm for efficient coverage and performance[J].Scientific Reports.15,26196(2025).

[11] YAN H,ZHANG M,ZHU Y Q,et al.Exploring multi-objective optimization scheduling strategies for power dispatch robots based on an improved particle swarm optimization algorithm framework[J].Scientific Reports.16,21119(2026).

[12] SEYDE N S, ALI B, REZA T M.Adaptive multiobjective differential evolution based on parallel cell coordinate system[J].Cluster Computing.28,606(2025).

[13] RAJA J,SOUFIENE B.A parallel cooperative coevolutionary multi-verse algorithm for large-scale multi-objective UAV path planning problems[J].Memetic Computing.17,44(2025).

[14]LI Z F,LIU L,ZHAO Z L, et al.Optimizing coal blending for metallurgical coke production using a reinforcement learning-enhanced evolutionary algorithm [J].Fuel.406,136777(2025).

[15] XUE F, WEN J B,WANG P W,et al.A decomposition-based multi-objective evolutionary algorithm with reinforcement learning for workflow scheduling in cloud computing environment[J].Cluster Computing .28,678(2025).

[16] JIANG Y Y, ZHANG Y Y. Path planning based on an improved 8-neighborhood node search strategy A* algorithm[J]. Journal of Electronic Measurement and Instrumentation.36,234-241(2022).

[17] FAN X C, LING X Y, HUANG H B, et al. Multi-robot multi-target path planning based on A* and simulated annealing algorithm[J]. Journal of Changzhou Institute of Technology.38,43-49(2025).

[18] FAN X C, LING X Y, YU Y Q, et al. Research on multi-robot multi-target path planning based on algorithm fusion[J]. Journal of Yunnan Minzu University (Natural Sciences Edition).34,342-349(2025).

[19]YANG L W, LI P, WANGT, et al. Multi-area collision-free path planning and efficient task scheduling optimization for autonomous agricultural robots[J].Scientific Reports.14,18347(2024).

[20] LIE M P,HISAO I,KALYANMOY D,et al.MaNSGA-II: Many-Objective NSGA-II[J]. IEEE Transactions on Emerging Topics in Computational Intelligence.9,63922-3937(2025).

[21] CHEN J H,OUYANG H B, LI S,et al.Altruistic NSGA-II with abandonment threshold and double selection strategy for solving multi-objective optimization problems[J]. Applied Soft Computing.170,112653(2025).

[22] MA H P,ZHANG Y J,SUN S Y, et al. A comprehensive survey on NSGA-II for multi-objective optimization and applications[J]. Artificial Intelligence Review.56,15217-15270(2023).

[23] SIMI K, SARADA J, BILAL A,et al.Adaptive Shannon Entropy-Driven Chaotic Grey Wolf Optimization (ASE-CGWO) for Combinatorial Optimization Problems[J]. biomimetics.11,689(2026).

[24] FELIPE C C, BRODERICK C, JORGE M,et al.Chaotic Regulation of Exploration and Exploitation in Bio-

Inspired Swarm Intelligence for Combinatorial Optimization[J]. biomimetics.11,540(2026).

[25] ZHANG H L, CAO H W. Research on Multi-objective Flexible Job Shop Scheduling Problem Based on Improved Salp Swarm Algorithm[J]. Journal of Anhui University of Technology（Social Sciences）. 41,17-23(2024).

[26] FUE X,CHEN Y Z,WANG P W,et al. A decomposition-based multi-objective evolutionary algorithm with Q-learning for adaptive operator selection[J] .The Journal of Supercomputing.80,21229-21283(2025).

[27]RAVI P,BRAJESH K U, RANVIJAY.ReinforceAdapt: multi-objective approach for environmental adaptation method[J]. Connection Science.37,2523960(2025).

[28] LIU J D,HE Y J ,SHEN B,et al.A Q-Learning Crested Porcupine Optimizer for Adaptive UAV Path Planning[J]. machines.13,566(2025).

[29] YANG X K, YANG H, WEN W, et al. Multi-objective flow shop scheduling optimization based on Hamming distance and neighborhood search[J]. Computer Systems & Applications.34,244-252(2025).

[30] PENG F,ZHENG L.Integrating real-time manufacturing data into a novel serial two-stage adaptive alternate genetic fireworks algorithm for solving stochastic type-II simple assembly line balancing problem[J]. Complex & Intelligent Systems.9,7075-7102(2023).

[31] YIN L F,LIU R K. A normalized deep neural network with self-attention mechanisms based multi-objective multi-verse optimization algorithm for economic dispatch[J].Applied Energy.383,125414(2025).

[32] BODO M, JESSE V R.Performance of efficient variants of the 2-Opt heuristic for the traveling salesperson problem[J].Discrete Applied Mathematics.375,7-16(2025).

[33] TANG J F,WANG HD,JIN Y C,et al.Knee-oriented expensive many-objective optimization via aggregation-dominance: A multi-task perspective[J]. Swarm and Evolutionary Computation.92,101813(2025).

[34] XUE Q,WU C J,NIE J Y,et al.An improved multi-objective animated oat optimization algorithm for resource-constrained construction project organization design[J]. Scientific Reports.16,10239(2026).

[35] MAO Y,XIAO L.Machine learning modeling and multi objective optimization of artificial detrusor[J]. Scientific Reports.15,13864(2025).

[36] YANG L W,LI P Y,WNAG T, et al.Multi-area collision-free path planning and efficient task scheduling optimization for autonomous agricultural robots[J]. Scientific Reports.14,18347(2024).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementaryinformation.docx