An Improved Grey Wolf Optimization with Multi-Strategy Ensemble for Robot Path Planning

Lin Dong, Xianfeng Yuan *, Bingshuo Yan, Yong Song, Qingyang Xu and Xiongyan Yang

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China
* Correspondence: yuanxianfeng@sdu.edu.cn

Abstract: Grey wolf optimization (GWO) is a meta-heuristic algorithm inspired by the hierarchy and hunting behavior of grey wolves. GWO has the superiorities of simpler concept and fewer adjustment parameters, and has been widely used in different fields. However, there are some disadvantages in avoiding premature and falling into local optimum. This paper presents an improved grey wolf optimization (IGWO) to ameliorate these drawbacks. Firstly, a modified position update mechanism for pursuing high quality solutions is developed. By designing an ameliorative position update formula, a proper balance between the exploration and exploitation is achieved. Moreover, the leadership hierarchy is strengthened by proposing adaptive weights of $\alpha$, $\beta$ and $\delta$. Then, a dynamic local optimum escape strategy is proposed to reinforce the ability of the algorithm to escape from the local stagnations. Finally, some individuals are repositioned with the aid of the positions of the leaders. These individuals are pulled to new positions near the leaders, helping to accelerate the convergence of the algorithm. To verify the effectiveness of IGWO, a series of contrast experiments are conducted. On the one hand, IGWO is compared with some state-of-the-art GWO variants and several promising meta-heuristic algorithms on 20 benchmark functions. Experimental results indicate that IGWO performs better than other competitors. On the other hand, the applicability of IGWO is verified by a robot global path planning problem, and simulation results demonstrate that IGWO can plan shorter and safer paths. Therefore, IGWO is successfully applied to the path planning as a new method.

Keywords: grey wolf optimization; multi-strategy ensemble; exploitation and exploration; path planning

1. Introduction

In recent years, with the development of robot technology, more and more work began to rely on robots to finish. Mobile robots gradually become a part of human industrial manufacture and mankind daily life [1–4]. Robots can execute operations such as perception and decision-making, assisting or even replacing human beings to complete heavy, repetitive or dangerous tasks. At present, the topics about robots covers robot navigation, robot mechanism design, human-robot interaction and so on [5–8]. One of the key technologies in robot navigation is global path planning [9,10], which refers to generating a safe and effective trajectory on the premise of knowing all environmental information. Path planning algorithms can be divided into traditional algorithms and meta-heuristic algorithms [11]. Traditional path planning algorithms include A* algorithm, rapidly-exploring random tree (RRT), Dijkstra algorithm, etc. Meta-heuristic algorithms include particle swarm optimization (PSO) [12], ant colony algorithm (ACO) [13], cuckoo algorithm (CS) [14], bat algorithm (BA) [15], grey wolf algorithm (GWO) [16], etc. On account of the advantages of fewer parameters and easy implementation, the investigation about meta-heuristic algorithms gains rapid expansion [17].

At present, heuristic algorithms have been widely applied. Wu et al. [18] successfully solved the graph partitioning problem using deterministic annealing neural network. In
Wu proposed a deterministic annealing neural network approach and verified the effectiveness of the algorithm through test problems. The meta-heuristic algorithm is regarded as an improved heuristic algorithm with the addition of random elements. Meta-heuristic algorithms, also known as intelligent optimization algorithms, are a class of methods that solve optimization problems based on computational intelligence mechanisms. They constantly imitate the rules, behaviors or mechanisms in physical, biological and social fields [20]. Due to the stochastic characters and the ability of saving the information in the optimization routine, meta-heuristic algorithms are feasible in settling complex and challenging optimization problems. In addition, meta-heuristic algorithms can be adapted to different applications by small adjustments [21]. Extensive researches and experiments in recent years indicate that meta-heuristic algorithms have unique advantages in solving robot path planning problems [22,23]. In [24], an improved ant colony algorithm was proposed and applied to the path planning of mobile robots. By extending the number of search directions of ants to 16, individual can query 24 neighborhoods of the current node each time, thus increasing the optional directions and search range in the search process and improving the search efficiency and accuracy of the algorithm. To enhance the ability of jumping out of the local minima, an improved PSO was proposed by employing an adaptive velocity formula into PSO [25]. Simultaneously, a robot path planning method with smoothing strategy was proposed by combining high-order Bezier curve. To improve the performance of PSO and solve the unmanned aerial vehicle (UAV) path planning problem in complex environment, spherical vector was introduced to PSO in [26]. Song et al. [27] designed a parallel cuckoo search algorithm (CS) to help UAV avoid obstacles and reach destination in a three-dimensional complex environment. In [28], a PSO variant based on evolutionary operator was proposed by introducing crossover operator and bee colony operator to solve multi-robot path planning problems. In [29], a hybrid PSO-GWO algorithm was proposed to optimize the robot trajectory, combining the characteristics of GWO and PSO.

GWO [16] was proposed by Seyedali Mirjalili et al. in 2014. This algorithm simulates the hunting process of grey wolves and has become one of the most popular algorithms in recent years [30,31]. GWO not only has the advantages of simple structure and few parameters like other meta-heuristic algorithms, but also possesses adaptive regulatory factor, which could better balance the intensification and diversification of GWO compared with other algorithms [32,33]. Therefore, GWO has attracted extensive attention of scholars and has been successfully applied in multifarious fields, including parameter extraction [34], job shop scheduling [35], feature selection [36], and disease classification prediction [37]. However, GWO also has the common problems of meta-heuristic algorithms, such as prematurity and slow convergence [38,39]. In order to improve the population diversity, reinforcement learning (RL) was introduced to GWO in [40]. Then the improved GWO was successfully applied to UAV path planning. In [41], aiming at the weakness that GWO is easy to fall into local optima in high-dimensional tasks, random leaders and random spiral-form motions were introduced into GWO. Heidari et al. [42] proposed an improved GWO based on Levy flight in order to cope with the disadvantages of prematurity and enhance the ability to solve global optimization problems. Luo [43] proposed an enhanced GWO variant to strengthen the search guidance of the head wolves. Experimental results proved that the convergence speed and accuracy of the algorithm were significantly improved. In [44], a new GWO variant was proposed for robot path planning, where the concept of loin swarm optimization (LSO) and dynamic weights were introduced into the original GWO to augment the searching ability of $\alpha$, $\beta$ and $\delta$ on the premise of ensuring population diversity. The above studies have manifested that GWO is useful in solving optimization problems, and the performance of the algorithm can be improved from different aspects.

In order to improve the performance of the algorithm, an improved grey wolf optimization (IGWO) is proposed in this paper, and its feasibility in the field of robot path
planning is demonstrated by simulation experiments. It mainly possesses the three following improvements:

1. A modified position update mechanism is proposed to improve the accuracy of the solution by providing the algorithm with a better tradeoff of exploration and exploitation.
2. A dynamic local optimum escape strategy is designed, which can help the algorithm jump out of local optima traps when the algorithm is considered to be trapped in local stagnation.
3. An individual repositioning method is proposed. This strategy can pull back some individuals of the solutions to the vicinity of the current leaders compulsorily to accelerate the convergence of the algorithm in the anaphasis of the iterations.

To verify the performance of IGWO and the feasibility of IGWO to solve the path planning problem, this paper carries out extensive comparative experiments, including numerical optimization experiments and robot path planning examples. In the numerical optimization experiments, 20 benchmark functions are selected for two groups of statistical experiments. In the first set of experiments, IGWO is compared with five excellent GWO variants proposed recently. In the second group, seven well-established meta-heuristic algorithms including GWO are picked out in contrast with IGWO. Experimental results illustrate that IGWO has obvious advantages both in convergence accuracy and speed on the majority of benchmark functions. In robot path planning examples, this paper designs two obstacle environment models and fully verifies the applicability of IGWO with several comparison tests. In addition, cubic spline interpolation is used to smooth the path and obtain the path more aligned with the kinematic characteristics of robot. Experimental results exhibit that compared with the contrast algorithms, the routes planned by IGWO is not only shorter, but also more stable, indicating that IGWO has higher application value.

The remaining parts of the paper are structured as follows: Section 2 reviews the original GWO briefly. Section 3 provides a detailed description of the proposed approach. Section 4 illustrates the information of the benchmark functions used in experiments and analyzes the statistical results in detail, which proves that the proposed IGWO has better performance. The application of IGWO to settle robot path planning problems is highlighted in Section 5. The comparative test results of path planning are presented to verify the effectiveness and feasibility of the improved algorithm applied to path planning. Finally, Section 6 summarizes the conclusion and identifies future works.

2. Review of GWO
2.1. Leadership Hierarchy

Grey wolves live in a pack with strict social hierarchy, which is also one of the inspirations for GWO as shown in Figure 1. A grey wolf pack could be divided into 4 levels: alpha (\(\alpha\)), beta (\(\beta\)), delta (\(\delta\)) and omega (\(\omega\)), from high to low, corresponding to the structural pyramid in Figure 1 from top to bottom. Each wolf plays specific role in the pack according to the level the wolf appertains to. \(\alpha\), also called the dominant wolf, is responsible for planning and deciding the hunting behavior of the group, representing the optimal solution obtained by the algorithm so far. \(\beta\) follows \(\alpha\) in the rankings, whose duty is to assist \(\alpha\) in decision-making. \(\delta\), ranking the third level in the social hierarchy, takes charge of reconnoitering, surveilling and guarding the pack as scouts, sentinels and guardians. Although \(\omega\) wolves are in the lowest status of the tribe, their presence is essential to maintain the pack peace [16]. In the mathematical model of GWO, the first three best-fit solutions in the population are regarded as \(\alpha\), \(\beta\) and \(\delta\) wolves, and the remaining wolves are regarded as \(\omega\) wolves.
2.2. Hunting Mechanism

GWO not only mimics the leadership of grey wolves in nature, but also mimics the hunting mechanism [45]. The mathematical model is conducted with respect to the behavior of encircling prey, and the action mode can be described by Equations (1)–(5).

\[
D = |C \cdot X_p - X| 
\]

\[
X^{t+1} = X_p - A \cdot D 
\]

\[
A = 2ar_1 - a 
\]

\[
C = 2r_2 
\]

\[
a = 2 - \frac{2t}{Max_{iteration}} 
\]

where \(a\) is the convergence factor, decreasing linearly from 2 to 0 with the number of iterations. \(r_1\) and \(r_2\) represent random vectors within \([0, 1]\). Both \(A\) and \(C\) are coefficient vectors. \(D\) denotes the approximate distance between the current wolf position and prey, \(X\) and \(X^{t+1}\) stand for the positions of the wolf at \(t\)th and \(t+1\)th iteration respectively. \(X_p\) represents the position of the prey at \(t\)th iteration.

To mathematically describe the hunting mechanism, GWO assumes that the higher-level wolves are closer to the prey in the pack, that is, \(\alpha\), \(\beta\) and \(\delta\) occupy the most powerful positions in the whole pack. Thus, the positions of remaining wolves are determined by the three wolves, which can be calculated as follows:

\[
\begin{align*}
D_\alpha &= C_1 \cdot X^\prime_\alpha - X^\prime \\
D_\beta &= C_2 \cdot X^\prime_\beta - X^\prime \\
D_\delta &= C_3 \cdot X^\prime_\delta - X^\prime \\
X^\prime_\alpha &= X^\prime_\alpha - A_1 \cdot D_\alpha \\
X^\prime_\beta &= X^\prime_\beta - A_2 \cdot D_\beta \\
X^\prime_\delta &= X^\prime_\delta - A_3 \cdot D_\delta \\
X^{t+1}_\cdot &= \frac{X^\prime_\alpha + X^\prime_\beta + X^\prime_\delta}{3}
\end{align*}
\]

where \(C_1\), \(C_2\) and \(C_3\) are random coefficient vectors defined by Equation (4). \(A_1\), \(A_2\) and \(A_3\) are coefficient vectors defined by Equation (3). \(X^\prime_\alpha\), \(X^\prime_\beta\) and \(X^\prime_\delta\) represent the positions of \(\alpha\), \(\beta\) and \(\delta\). GWO pseudocode are shown in Algorithm 1, and the location update of wolves are exhibited in Figure 2.
Figure 2. Update of position in GWO.

Algorithm 1. The pseudocode of conventional GWO

1. Generate a population $X_i (i = 1, 2, ..., n)$ randomly
2. Initialize the parameters of GWO ($\text{max\_iteration}, a, A$ and $C$)
3. Calculate the fitness values and assign $\alpha$, $\beta$ and $\delta$
4. While ($t < \text{max\_iteration}$)
   5. For each grey wolf
   6. Update the position of the current grey wolf using Equations (6)–(8)
   7. End for
8. Update $a$, $A$ and $C$
9. Amend the grey wolves’ positions beyond boundary limits
10. Calculate the fitness values of the new positions
11. Update the $\alpha$, $\beta$ and $\delta$
12. $t = t + 1$
13. End while
14. Return the position of $\alpha$

3. Development of IGWO

Three modifications are proposed in this section to overcome the defects of GWO. Firstly, a modified position update mechanism is designed, where an improved updating formula is constructed, and an enhanced diversity method including opposition-based learning strategy and Cauchy mutation is imported. In the proposed updating formula, the leaders ($a$, $\beta$ and $\delta$) compose the leader-determined term. The leaders will match different leadership authorities according to their fitness values, which is more suited to the hierarchy in nature. Furthermore, a random term is added on the basis of the leader-determined term to increase the exploration capability of the algorithm. Secondly, a dynamic local optimum escape strategy is proposed to escape from the local optimal value. In this strategy, a tuning parameter is employed to dynamically adjust the degree of variation. Finally, an individual repositioning method is imported to speed up the convergence rate of the algorithm by using the positions of $\alpha$, $\beta$ and $\delta$. The framework of IGWO is presented in Figure 3 and the details of IGWO are introduced and discussed as follows.
3.1. Modified Position Update Mechanism

Wolves in nature have a strict hierarchy. The higher the rank of the wolf, the more dominant it is over the whole population and the stronger it is in the hunting process. A rigid hierarchy is crucial in the hunting process. However, in the hunting mechanism of GWO, as the position updating mechanism described by Equation (8), the leaders have the same power (1/3) over the population. It’s apparent that the distribution of power is incongruent with the real hierarchy.

Due to the fitness values of the individuals can reflect the individuals’ merits and determine their hierarchy, this paper takes the fitness value into consideration and
designs an adaptive weight coefficient based on the fitness value to better simulate the real hierarchy of the wolf pack. The mathematical expression is given as Equations (9)–(11).

\[
\theta_i = \begin{cases} 
\frac{1}{f(X_i)} + 0.0001, & i \in \{\alpha, \beta, \delta\}, \text{ when objective function is minimized} \\
\frac{1}{f(X_i)}, & i \in \{\alpha, \beta, \delta\}, \text{ when objective function is maximized} 
\end{cases}
\]  

(9)

where \( \theta_i, i \in \{\alpha, \beta, \delta\} \) is the intermediate variable for calculating the adaptive weight coefficient. Moreover, the larger \( \theta_i \) is, the closer the corresponding position is to the prey. In addition, the constant 0.0001 is added to prevent infinity when \( f(X_i) \) equals 0 in the minimization problem. Then, the weight coefficients of \( \alpha, \beta \) and \( \delta \) can be calculated using Equation (10), and the position updating formula is rewritten as Equation (11).

\[
w_i = \theta_i / \left( \theta_\alpha + \theta_\beta + \theta_\delta \right), \quad i \in \{\alpha, \beta, \delta\} \tag{10}
\]

\[
X' = \text{rand} \cdot \left( w_\alpha X'_\alpha + w_\beta X'_\beta + w_\delta X'_\delta \right) \tag{11}
\]

where \( \text{rand} \) is a random number between 0 and 1. Taking optimizing Sphere function as an example, Figure 4 shows the changing of the leaders’ weight coefficients \( (w_\alpha, w_\beta, \text{ and } w_\delta) \) calculated by Equations (9) and (10). It is obvious that during the optimization, the corresponding weights of the \( \alpha, \beta, \) and \( \delta \) are descend sequentially, which is more consistent with the hierarchical system of a wolf pack.

![Figure 4. Adaptive weight coefficients of the leaders. (a) 2-D version of Sphere function; (b) Values of \( w_\alpha, w_\beta, \text{ and } w_\delta. \)](image)

Because the individuals are randomly initialized and distributed in the search space, \( \alpha, \beta, \) and \( \delta \) have similar probability of approaching prey compared to \( \omega \) in preliminary phase. Apparently, the leader wolves are incompetent to guide the population search at the beginning. Hence using the leaders to guide the search would increase the risk of the premature convergence. To settle this deficiency, this paper adds a random term to the position update formula, which could randomly select individuals from the community to take charge of next search. Combining with Equation (11), the improved position updating formula is constructed ultimately as Equation (12).

\[
X' = n_1 \cdot \text{rand}_1 \cdot \left( w_\alpha X'_\alpha + w_\beta X'_\beta + w_\delta X'_\delta \right) / 3 + n_2 \cdot \left( X' + \text{rand}_2 \cdot \left( X'_{\text{rand}} - X' \right) \right) \tag{12}
\]

\[
n_1 = \frac{t}{\text{Max}_\text{iteration}} \tag{13}
\]

\[
n_2 = 1 - n_1 \tag{14}
\]

where \( X_{t+1} \) and \( X_t \) denote the current wolf’s position at \( t+1 \)th and at \( t \)th iteration respectively. \( X'_{\text{rand}} \) represents the randomly selected wolf’s position. \( \text{rand}_1 \) and \( \text{rand}_2 \) are
random numbers in [0, 1]. \( n_1 \) and \( n_2 \) are adjustment parameters that can bring about a tradeoff of the leader-determined term and the random term. Moreover, \( n_1 \) and \( n_2 \) are set to add up to 1, ensuring the convergence of solution. \( n_1 \) will increase linearly to 1 with the number of iterations, and the corresponding search behavior could be described as follows: While \( n_1 \) is smaller compared with \( n_2 \) in the early stage, the random term provides the main guidance of movement for the pack, conducting a full exploration in search space. In the later stage, \( n_1 \) increases to 1, and the dominating rights returns to the three leaders, enhancing the exploitation capacity of the approach. Compared with Equation (8), the new formula could better simulate the hierarchy of the wolf society, and balance the global and local searches.

In addition, this paper introduces the population opposition-based learning \([46–48]\) and Cauchy mutation to increase population diversity. The mathematical model is shown in Equation (15), where \( \text{gamma} \) is the scale parameter of mutation. A threshold \( \tau \) is set to decide how to renew the solutions. The enhanced diversity method will be executed with the probability of \( \tau \) and Equation (12) will be used with the probability of \( (1 - \tau) \). In this paper, \( \tau \) is set to 0.3. Finally, the greedy strategy is employed in this paper to reserve the high-quality solutions. The mathematical expression is defined as Equation (16).

\[
X_{\text{new}} = \begin{cases} 
X_i + \text{Cauchy}(\alpha, \text{gamma}), & i \in \{\alpha, \beta, \delta\} \\
\text{rand}(1, D) \cdot (\alpha \mathbf{b} + \beta \mathbf{b}) - X_i, & \text{else}
\end{cases}
\]  

(15)

\[
X^t = \begin{cases} 
X^t, & \text{if } f(X^t) < f(X^{t-1}) \\
X^{t-1}, & \text{otherwise}
\end{cases}
\]  

(16)

3.2. Dynamic Local Optimum Escape Strategy

The ultimate goal of this subsection is to address the concern of GWO being prone to fall into local minima. Inspired by random walk strategy in the improved symbiotic search algorithm (ISOS) \([49]\), we design a dynamic local optimum escape strategy. The phenomenon of ISOS can be classified into three fundamental categories: mutualism phase, commensalism phase and parasitism phase. Previous researches point out that in the commensalism phase in nature, it is very likely that there are no suitable individuals nearby to establish commensalism relationship. It is crucial to explore new individuals and build commensalism. Miao \([49]\) implemented a random walk strategy for the individual \( i \) to achieve a disturbance and jump out of the current solution. Then, the individual can be empowered to build commensalism relationship with the new result. More precisely, it’s the second component of Equation (17) that is the disturbance term to help the algorithm generate the new solution. And the specific model of random walk can be defined as follows:

\[
X_{\text{new}} = X + \text{rand}(1, D) \cdot (X^t - \hat{X}^t) \cdot R_i
\]  

(17)

\[
R_i = \begin{cases} 
1, & r > K \\
0, & \text{otherwise}
\end{cases}
\]  

(18)

where \( \text{rand}(1, D) \) is a \( D \)-dimensional random matrix in the range of \([0, 1]\), \( (X_m^t - \hat{X}_i^t) \) represents the distance of random walk. \( X_m^t, \hat{X}_i^t \) are random candidates in the search space, satisfying \( m \neq n \neq i \). \( R_i \) is a logical value that determines whether to execute the random walk strategy. \( K \) is a fixed threshold. \( r \) is a random number. When \( r \) is greater than \( K \), the random walk strategy will be executed. Otherwise, the original solution will be maintained.

In view of Cauchy mutation having large variation step, we propose dynamic local optimum escape strategy by importing the concept of Cauchy mutation. Its mathematical model can be described as follows:
\[ X'^{t+1} = \text{Cauchy}(X', \sigma) + \text{rand}_1 \cdot X' - \text{rand}_2 \cdot X'_{\text{rand}} \]  

(19)

\[ \sigma = \frac{a}{2} |X'_u - X'_i| \]  

(20)

where \( \text{rand}_1 \) and \( \text{rand}_2 \) are random numbers inside \([0, 1]\). \( \sigma \) is a tuning parameter standing for the extent of mutation, which changes dynamically during the search process. \( a \) changes according to Equation (5). \( \text{rand}_1 \cdot X' - \text{rand}_2 \cdot X'_{\text{rand}} \) is a small random disturbance that can further increase the ability of fleeing local optimum.

In addition, we select the top half of the superior agents as subgroup GS. The average fitness value of GS serves as the strategy’s enable switch. The average fitness value of GS can be calculated by Equation (21). The search state will be considered fallen into stagnation area and immediately activate dynamic local optimum escape strategy if Equation (22) is satisfied.

\[ \text{mean}_{\text{GSvalue}}' = \frac{2}{n} \sum_{i=1}^{n} f\left( X'_i \right) \]  

(21)

\[ \text{mean}_{\text{GSvalue}}^{t+1} = \text{mean}_{\text{GSvalue}}' \]  

(22)

In terms of the convergence of the algorithm, since the tuning parameter \( \sigma \) can be regulated according to the number of iterations and the distance from the predicted prey, it’s trustworthy that the proposed strategy does not affect the algorithm convergence. It is assumed that at the end of the evolutionary process, the average fitness of GS remains unchanged and \( a \) is close to 0. The whole pack is located near the prey in this period, causing \( |X'_u - X'_i| \) and \( \text{rand}_1 \cdot X' - \text{rand}_2 \cdot X'_{\text{rand}} \) almost equaling to 0. It can be proved that the disturbance to the current position caused by the dynamic local optimum escape strategy is quite puny, guaranteeing the algorithm convergence.

3.3. Individual Repositioning Method

Miao et al. [50] repositioned the three wolves with the worst quality by using \( \alpha, \beta, \delta \) and the original positions in each iteration to remove the poor candidates and prevent search bias. Inspired by this idea, we propose a new individual repositioning method, which ignores the original position information and only uses the position of the head wolves, to speed up the convergence of the algorithm. The corrected position of the agents can be calculated by Equation (23), and the relocation process is shown in Figure 5.

\[ X_{\text{new, worst}} = r_1 \cdot X_u + r_2 \cdot X_\beta + r_3 \cdot X_\delta \]  

(23)

\[ r_1 = \text{rand}(\frac{1-r_2}{2}, 1-r_1) \]  

(24)

\[ r_2 = 1 - r_1 - r_3 \]  

(25)

where \( X_{\text{new, worst}} \) is the wolf’s position after repositioning, and \( r_1 \) is a random number within \([0.5, 1]\). The sum of \( r_2, r_2 \) and \( r_3 \) is set to 1 to ensure the convergence of the solutions.
Sensors 2022, 22, 6843

The individuals that need to be relocated are those with low fitness. The purpose of the individual repositioning method is to make the search rapidly converge to the optimal value by correcting poorly behaved agents artificially. As the three leader wolves possess the best experience in the whole pack, it’s hoped that these individuals can be repositioned to the three leaders’ neighborhood, which is considered more likely to close to the prey. And the concept is embodied in Equation (23), where the positions of $\alpha$, $\beta$ and $\delta$ are used to generate a new position. Furthermore, as mentioned in Section 3.1, the authority ranking of the leaders is $\alpha > \beta > \delta$. To carry out the concept of the wolf hierarchy, the weights of the leaders are set to $r_1 > r_2 > r_3$ in Equation (23).

It can be observed in Figure 5 that as the proposed individual repositioning method forces some individuals to relocate near the leaders, it can reduce the population diversity partly. However, the search prophase requires a high population diversity to find more potential optimal solutions. In other words, the individual repositioning method proposed in this subsection is supposed to be dormant in the early stage of search. Therefore, the position revising method won’t be activated until anaphase to maintain population diversity.

4. Numerical Optimization Experiments

In this section, the proposed algorithm is examined on 20 benchmark functions. These functions are all minimization problems, including 6 unimodal functions, 4 multimodal functions, 3 shifted and rotated multimodal functions, 4 fixed-dimension functions and 3 composite functions. The parameter settings of the functions are presented in Tables 1 and 2, where Dim and Range represent the dimension and the boundary of the solution domain respectively, and $F^*$ symbolizes the optimal value for each benchmark function. To verify the superiority of IGWO, we conduct two suites of comparison experiments. In the first suite, IGWO is compared with five well-established GWO variants namely MixedGWO [51], GWOCS [34], LearnGWO [33], mGWO [52], and RW_GWO [53]. In the second suite, some mature meta-heuristic algorithms are chosen for the comparison of IGWO. These algorithms include GWO [16], particle swarm optimization (PSO) [12], artificial bee colony (ABC) [54], sine cosine algorithm (SCA) [55], whale optimization algorithm (WOA) [56], multi-verse optimizer (MVO) [57] and tunicate swarm algorithm (TSA) [58]. Additionally, to ensure the fairness and objectivity of the comparison experiments, all experiments are run on CPU Core i5-9400F and 8 GB RAM, and programmed on MATLAB R2018b.
Table 1. Unimodal, multimodal and shifted and rotated multimodal benchmark functions. * symbolizes the optimal value for each benchmark function.

| Function | Test Function                                                                                                                                                                                                 | Dim | Range          | F * |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----------------|-----|
| F1       | \( f_1 = \sum_{i=1}^{n} x_i^2 \)                                                                                                                                                                          | 30  | \([-100, 100]\) | 0   |
| F2       | \( f_2 = \sum_{i=1}^{n} |x_i| \)                                                                                                                                                                                                       | 30  | \([-10, 10]\)  | 0   |
| F3       | \( f_3 = \sum_{i=1}^{n} \left( x_i^2 + \sum_{j=i+1}^{n} x_j \right) \)                                                                                                                                       | 30  | \([-100, 100]\) | 0   |
| F4       | \( f_4 = \max \left( \left| x_i \right|, 1 \leq i \leq n \right) \)                                                                                                                                       | 30  | \([-100, 100]\) | 0   |
| F5       | \( f_5 = \sum_{i=1}^{n} \left[ 100 \left( x_i - x_i^3 \right)^2 + \left( x_i - 1 \right)^2 \right] \)                                                                                                     | 30  | \([-30, 30]\)  | 0   |
| F6       | \( f_6 = \sum_{i=1}^{n} x_i^r + \text{random}[0, 1] \)                                                                                                                                                   | 30  | \([-1.28, 1.28]\) | 0   |
| F7       | \( f_7 = \sum_{i=1}^{n} \left[ x_i^2 - 10 \cos \left( 2 \pi x_i \right) + 10 \right] \)                                                                                                       | 30  | \([-5.12, 5.12]\) | 0   |
| F8       | \( f_8 = -20 \exp \left( -0.2 \sqrt{\sum_{i=1}^{n} x_i^2} - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos \left( 2 \pi x_i \right) \right) + 20 + e \right) \) | 30  | \([-32, 32]\)  | 0   |
| F9       | \( f_9 = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{n}} \right) + 1 \)                                                                                     | 30  | \([-600, 600]\) | 0   |
|          | \( f_{9u} = \pi \sum_{i=1}^{n} \left( y_i - 1 \right)^2 \left[ 1 + 10 \sin^2 \left( \pi y_i \right) \right] + \sum_{i=1}^{n} a_i \left( x_i - 10, 100, \cdot \right) \)                       |     |                |     |
|          | \( y_i = 1 + \frac{x_i + 1}{4} \)                                                                                                                                                                         | 30  | \([-50, 50]\)  | 0   |
|          | \( u(x_i, a, k, m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases} \)                                                                               |     |                |     |
| F10      | Shifted and Rotated Katsuura Function                                                                                                                                                                      | 30  | \([-100, 100]\) | 1200|
| F11      | Shifted and Rotated HappyCat Function                                                                                                                                                                     | 30  | \([-100, 100]\) | 1300|
| F12      | Shifted and Rotated HGBat Function                                                                                                                                                                       | 30  | \([-100, 100]\) | 1400|

Table 2. Fixed-dimension multimodal and composition benchmark functions. * symbolizes the optimal value for each benchmark function.

| Function | Test Function                                                                                                                                                                                                 | Dim | Range          | F * |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----------------|-----|
| F14      | \( f_{14} = \sum_{i=1}^{n} x_i \left( b_i^2 + b_i x_i + c_i \right)^2 \)                                                                                                                               | 4   | \([-5, 5]\)   | 0.00030|
| F15      | \( f_{15} = \left( x_1 - \frac{5.1}{4 \pi} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8 \pi} \cos (x_1) \right) + 10 \)                                                    | 2   | \([-5, 5]\)   | 0.398 |
| F16      | \( f_{16} = \left( 1 + (x_1 + x_2 + 1)^{19 - 14 x_1 + 3 x_2^2 - 14 x_2 + 6 x_1 x_2 + 3 x_2^3} \right) \left[ 30 + (2 x_1 - 3 x_2)^2 \left( 18 - 32 x_1 + 12 x_2^2 + 48 x_2 - 36 x_1 x_2 + 27 x_2^3 \right) \right] \) | 2   | \([-2, 2]\)   | 3    |
| F17      | \( f_{17} = -\sum_{i=1}^{n} \left( (X-a)(X-a)^r + c_i \right) \)                                                                                                                                 | 4   | \([0, 10]\)   | -10.1532|
| F18      | Composition Function 1 (N = 5)                                                                                                                                                                          | 30  | \([-100, 100]\) | 2300 |
| F19      | Composition Function 2 (N = 3)                                                                                                                                                                          | 30  | \([-100, 100]\) | 2400 |
| F20      | Composition Function 3 (N = 3)                                                                                                                                                                          | 30  | \([-100, 100]\) | 2500 |
4.1. Comparison of IGWO with Different GWO Variants

In this subsection, IGWO is compared with five state-of-the-art GWO variants. The number of iterations is set to 1000 and the population size is fixed to 30 for all algorithms. All algorithms run independently thirty times on each benchmark function to reduce the randomness of experimental results. Moreover, Wilcoxon rank-sum test at the significance level of 5% is adopted. “+”, “−” and “≈” respectively represent that IGWO is superior to, inferior to, and similar to the corresponding contrast method. The average value (Mean) and the standard deviation (Std) of the numerical results of 30 independent experiments are listed in Tables 3 and 4, where the best Mean and Std are highlighted in bold. Meanwhile, Wilcoxon rank-sum test results (T) and algorithm ranking (R) are also recorded in Tables 3 and 4. Finally, the overall Wilcoxon rank-sum test results and the mean ranking of the algorithms are provided in Table 5.

Table 3. Results of GWO variants on unidimensional, multimodal and shifted and rotated multimodal benchmark functions. The best values are highlighted in bold.

| Variants | F1          | F2          | F3          | F4          | F5          | F6          | F7          | F8          | F9          | F10         | F11         | F12          |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|          | Mean        | Std         | R/T         | Mean        | Std         | R/T         | Mean        | Std         | R/T         | Mean        | Std         | R/T         |
| MixedGWO | 3.83 × 10^{-30} | 1.60 × 10^{-39} | 1.3498 × 10^{-132} | 7.04 × 10^{-30} | 1.46 × 10^{-22} | 0           | 3.94 × 10^{-29} | 2.27 × 10^{-58} | 2.8395 × 10^{-122} | 1.47 × 10^{-38} | 9.50 × 10^{-22} | 0           |
| GWOCS    | 5/+         | 4/+         | 2/+         | 3/+         | 6/+         | 1           | 1.09 × 10^{-18} | 4.00 × 10^{-35} | 6.09 × 10^{-49} | 3.02 × 10^{-38} | 1.04 × 10^{-11} | 0           |
| LearnGWO | 3.67 × 10^{-39} | 1.42 × 10^{-38} | 1.27 × 10^{-132} | 6.85 × 10^{-30} | 1.24 × 10^{-22} | 0           | 3.94 × 10^{-29} | 2.27 × 10^{-58} | 2.8395 × 10^{-122} | 1.47 × 10^{-38} | 9.50 × 10^{-22} | 0           |
| mGWO     | 5/+         | 4/+         | 2/+         | 3/+         | 6/+         | 1           | 1.09 × 10^{-18} | 4.00 × 10^{-35} | 6.09 × 10^{-49} | 3.02 × 10^{-38} | 1.04 × 10^{-11} | 0           |
| RW_GWO   | 3.67 × 10^{-39} | 1.42 × 10^{-38} | 1.27 × 10^{-132} | 6.85 × 10^{-30} | 1.24 × 10^{-22} | 0           | 5/−        | 6.77 × 10^{-58} | 3.57 × 10^{-49} | 2.83 × 10^{-39} | 3.35 × 10^{-33} | 1.02 × 10^{-9} | 0           |
| IGWO     | 3.67 × 10^{-39} | 1.42 × 10^{-38} | 1.27 × 10^{-132} | 6.85 × 10^{-30} | 1.24 × 10^{-22} | 0           | 5/−        | 6.77 × 10^{-58} | 3.57 × 10^{-49} | 2.83 × 10^{-39} | 3.35 × 10^{-33} | 1.02 × 10^{-9} | 0           |
Table 4. Results of GWO variants on fixed-dimension multimodal and composition benchmarks. The best values are highlighted in bold.

| R/T | Mean | Std  | Mean  | Std  | Mean  | Std  | Mean  | Std  | Mean  | Std  |
|-----|------|------|-------|------|-------|------|-------|------|-------|------|
|     | 4/+  | 3/+  | 6/+   | 5/+  | 2/+   | 1    |
| F13 | 1435.3 | 1410.8 | 1499.5 | 1472.9 | 1400.8 | **1400.4** |
| R/T | 107.6842 | 50.4025 | 124.8757 | 126.7876 | 1.7689 | **1.7390** |

| R/T | Mean | Std  | Mean  | Std  | Mean  | Std  | Mean  | Std  | Mean  | Std  |
|-----|------|------|-------|------|-------|------|-------|------|-------|------|
|     | 4/+  | 3/+  | 6/+   | 5/+  | 2/+   | 1    |
| F14 | 0.0099 | 3.78 × 10⁻⁴ | 5.70 × 10⁻³ | 8.73 × 10⁻⁴ | 0.0023 | **3.33 × 10⁻⁴** |
| R/T | 0.0073 | 0.0013 | 4.37 × 10⁻⁵ | 3.97 × 10⁻⁸ | 0.0047 | **0.0329** |
| F15 | 0 | 8.74 × 10⁻⁷ | 0.0224 | 3.97 × 10⁻⁸ | 2.14 × 10⁻⁴ | **1.03 × 10⁻⁵** |
| R/T | 6/+  | 3/+  | 5/+   | 2/+   | 5/+   | 1    |
| F16 | 3 | 3 | 3.0093 | 3 | 3 | 3 |
| R/T | 1/−  | 4/+  | 6/+   | 2/+   | 5/+   | 3    |
| F17 | 2.83 × 10⁻⁴ | 7.1031 | 3.8401 | 12.3134 | 11.3324 | 5.7099 |
| R/T | 1/−  | 3/+  | 6/+   | 5/+   | 4/+   | 2    |
| F18 | 2662.4 | 2642.7 | 2745.2 | 2673.7 | 2626.2 | **2500** |
| R/T | 5/+  | 3/+  | 4/+   | 6/+   | 2/+   | 1    |
| F19 | 2600.2 | 2600 | 2600 | 2600.4 | 2600 | **2600** |
| R/T | 5/+  | 3/+  | 4/+   | 6/+   | 2/+   | 1    |
| F20 | 2704.5 | 2700.4 | 2700 | 2700.9 | 2712.5 | **2700** |
| R/T | 5/+  | 3/+  | 4/+   | 6/+   | 1    |

Table 5. Overall Wilcoxon rank-sum test results and mean rank results of GWO variants.

| Result | MixedGWO | GWOCS | LearnGWO | mGWO | RW_GWO | IGWO |
|--------|----------|-------|----------|------|--------|------|
| +/+≈/− | 18/0/2   | 18/0/2 | 17/2/1   | 17/1/2 | 17/0/3 | ~    |
| Mean rank | 4.7  | 3.25  | 3.5  | 4.05 | 3.75 | 1.5  |
| Overall rank | 6   | 2     | 4    | 5    | 3    | 1    |

4.1.1. Analysis of Numerical Results

Referring to the data presented in Tables 3 and 4, IGWO hits the fittest results on fourteen benchmark functions out of twenty functions (14/20), then RW_GWO hits three out of twenty (3/20), then MixedGWO and LearnGWO hit two out of twenty (2/20). Finally, GWOCS and mGWO obtain zero out of twenty (0/22).

According to Table 3, with regard to unimodal functions (F1-F6), it can be proved that IGWO has higher chance of hitting the optimal value on most unimodal functions. It’s observed that IGWO ranks first on four functions (F1-F4) and obtains the theoretical optimal solutions of these benchmarks. Despite IGWO misses best result on F5 and F6, IGWO comes to the third place on F5 and the second place on F6 with the result very close to the best one. The improved updating formula can bring up an excellent tradeoff between exploitation and exploration, increasing the chance of positioning the optimum, while the greedy strategy ensures the preservation of the worthy candidates in each generation, which can further improve solution accuracy. For the multimodal functions
(F7-F10), IGWO hits the global optimal values on F7 and F9 and obtains the best result among the six variants on F8, signifying the better search performance of the proposed IGWO. As the dynamic local optimum escape strategy can conduct more random operations, it can be easier for the algorithm to explore the search space more fully and avoid falling into the multiple local optimal peaks. For the shifted and rotated multimodal functions (F11-F13), IGWO gains best results on F12 and F13. Additionally, the results gained by RW_GWO are more accurate than remaining methods.

Table 4 indicates that, with respect to the multimodal fixed-dimension functions (F14-F17), IGWO yields the best results on F14 and F15, and locates the theoretical optimal values on F15 and F16. Meanwhile, MixedGWO converges to the theoretical optimal values on F16 and F17, and shows a little more stable search performance. For the composite functions (F18-F20), IGWO is superior to other approaches. Although LearnGWO get the same mean value as IGWO on F19 and F20, its standard deviation is bigger than that of IGWO, which means that IGWO is more stable and reliable than LearnGWO. The improvement of the performance should be owed to the modified strategies presented in this paper, which have diverse impact in dealing with different characteristics of the complex functions.

Furthermore, Table 5 summarizes the results above, showing the overall ranking of the six approaches is: IGWO > GWOC > RW_GWO > LearnGWO > mGWO > MixedGWO.

4.1.2. Analysis of Convergence Curves

Figures 6–13 exhibit the convergence curves of the six GWO variants on eight test functions (F1, F3, F8, F9, F12, F14, F18, and F20). The x-axis and the y-axis represent the iterations and the corresponding best average fitness values achieved by thirty independent experiments, respectively.

![Comparison results of GWO variants on F1](image)

**Figure 6.** Comparison results of GWO variants on F1.
Figure 7. Comparison results of GWO variants on F3.

Figure 8. Comparison results of GWO variants on F8.

Figure 9. Comparison results of GWO variants on F9.
Figure 10. Comparison results of GWO variants on F12.

Figure 11. Comparison results of GWO variants on F14.

Figure 12. Comparison results of GWO variants on F18.
Figures 6 and 7 indicates that the curve of IGWO can converge to the global optimum at a gallop, confirming that IGWO has better search accuracy and speed on F1 and F3. Furthermore, LearnGWO performed relatively well on F1 and F3 in comparison to MixedGWO, GWOCs, RW_GWO and mGWO, with the accuracy up to 1E10-120. Figure 8 illustrates that after about forty iterations, IGWO has already reached the final result with higher accuracy. From Figure 9, it can be observed that both IGWO and mGWO can converge to the optimal value on F9 rapidly, but IGWO can avoid the local optimum more efficiently. Figure 10 shows that RW_GWO and IGWO own higher convergence accuracy on F12, and IGWO ranks first by virtue of slightly higher accuracy. For the function F14, Figure 11 exhibits that the convergence speed of the six algorithms is similar, but the convergence accuracy of IGWO is slightly better. As shown in Figures 12 and 13, compared with other improved versions, IGWO has noticeable virtue both in convergence speed and accuracy. In addition, Figure 13 demonstrates that the accuracy gap of the six algorithms is small, yet IGWO possesses an obvious advantage in the search speed. In summary, as presented in Figures 6–13, the proposed strategies can visually improve the search accuracy and convergence speed on most functions.

Therefore, from the above discussions of Tables 3–5 and Figures 6–13, it can be concluded that the proposed IGWO has better optimization precision and convergence speed for the majority of the twenty benchmarks, which indicates that the proposed strategies can effectively improve the performance of the algorithm.

4.2. Comparison of IGWO with Other Meta-Heuristic Algorithms

In this subsection, IGWO is compared with the original GWO and six other well-known meta-heuristic algorithms, including PSO, ABC, SCA, WOA, MVO and TSA. The benchmark functions used in this suite are enumerated in Tables 1 and 2. Each algorithm runs independently 30 times, and the average value (Mean) and the standard deviation (Std) of the results are recorded in Tables 6 and 7. The optimal Mean and standard deviation obtained by the eight algorithms are marked in bold. In addition, Wilcoxon rank-sum test is also conducted with the confidence interval of 5% to test whether there are significant differences between the proposed variant and other competitors, and the algorithms are ranked. The rank-sum test results (T) and the final ranking results (R) are shown in Tables 6 and 7. Table 8 summarizes the data above.
Table 6. Results of unimodal, multimodal and shifted and rotated multimodal benchmark functions.

| GWO  | SCA  | PSO  | WOA  | ABC  | TSA  | MVO  | IGWO  |
|------|------|------|------|------|------|------|-------|
| F1   | Mean | 2.07×10⁻⁵ | 1.25×10⁻² | 9.43×10⁻⁷ | 6.59×10⁻¹⁵⁰ | 0.7887 | 6.02×10⁻⁶ | 0.2963 | 0 |
|      | Std  | 1.33×10⁻⁵⁸ | 1.07×10⁻¹ | 1.88×10⁻⁷ | 1.77×10⁻¹⁴⁸ | 2.1909 | 1.59×10⁻⁵ | 0.3969 | 0 |
|      | R/T  | 3/+ | 6/+ | 4/+ | 2/+ | 8/+ | 5/+ | 7/+ | 1 |
| F2   | Mean | 1.83×10⁻³⁴ | 1.20×10⁻⁵ | 6.0004 | 1.27×10⁻¹⁰² | 0.0363 | 0.0208 | 0.4606 | 0 |
|      | Std  | 1.78×10⁻³³ | 1.26×10⁻⁵ | 38.9865 | 2.18×10⁻¹⁰¹ | 0.056 | 0.0305 | 1.1093 | 0 |
|      | R/T  | 3/+ | 4/+ | 8/+ | 2/+ | 6/+ | 5/+ | 7/+ | 1 |
| F3   | Mean | 1.20×10⁻¹² | 6.57×10³ | 16.3022 | 1.76×10⁴ | 3.41×10⁴ | 2.67×10⁴ | 45.3638 | 0 |
|      | Std  | 3.44×10⁻¹¹ | 4.86×10⁵ | 46.3105 | 4.31×10⁴ | 2.04×10⁴ | 1.49×10⁴ | 120.8618 | 0 |
|      | R/T  | 2/+ | 5/+ | 3/+ | 6/+ | 8/+ | 7/+ | 4/+ | 1 |
| F4   | Mean | 2.25×10⁻¹⁴ | 17.9 | 0.6072 | 36.9902 | 53.3308 | 26.7721 | 1.0175 | 0 |
|      | Std  | 1.07×10⁻¹³ | 9.85 | 0.6718 | 113.8156 | 24.6552 | 17.5272 | 1.7441 | 0 |
|      | R/T  | 2/+ | 5/+ | 3/+ | 7/+ | 8/+ | 6/+ | 4/+ | 1 |
| F5   | Mean | 26.7285 | 579.7010 | 56.6336 | 27.2767 | 1.72×10⁴ | 125.1067 | 425.0734 | 27.6068 |
|      | Std  | 3.1903 | 1.41×10⁷ | 233.9869 | 3.1856 | 5.29×10⁴ | 232.5186 | 3.53×10³ | 3.7676 |
|      | R/T  | 1/+ | 6/+ | 4/+ | 2/+ | 8/+ | 5/+ | 7/+ | 1 |
| F6   | Mean | 8.28×10⁻⁴ | 5.09×10⁻² | 4.8865 | 0.0021 | 0.2127 | 0.2946 | 0.0222 | 9.13×10⁻⁵ |
|      | Std  | 0.0016 | 7.87×10⁻² | 26.4596 | 0.0104 | 0.1851 | 0.2579 | 0.0465 | 2.49×10⁻⁴ |
|      | R/T  | 2/+ | 5/+ | 3/+ | 6/+ | 8/+ | 7/+ | 4/+ | 1 |
| F7   | Mean | 0.7531 | 10.4 | 86.7988 | 8.53×10⁻¹⁵ | 224.9901 | 107.1537 | 111.7582 | 0 |
|      | Std  | 10.908 | 16.4 | 123.4636 | 1.21×10⁻¹³ | 93.4056 | 53.0099 | 198.3001 | 0 |
|      | R/T  | 3/+ | 4/+ | 8/+ | 3/+ | 2/+ | 8/+ | 6/+ | 7/+ | 1 |
| F8   | Mean | 1.67×10⁻¹⁴ | 16.1 | 4.24×10⁻⁵ | 4.97×10⁻¹⁵ | 1.569 | 1.3307 | 0.9705 | 8.88×10⁻³⁶ |
|      | Std  | 1.55×10⁻¹⁴ | 8.48 | 2.64×10⁻¹⁴ | 1.36×10⁻¹⁴ | 2.5771 | 4.6003 | 3.2009 | 0 |
|      | R/T  | 3/+ | 4/+ | 8/+ | 3/+ | 2/+ | 7/+ | 6/+ | 5/+ | 1 |
| F9   | Mean | 0.0036 | 0.335 | 0.0094 | 0.0036 | 0.8348 | 0.2145 | 0.554 | 0 |
|      | Std  | 0.0299 | 0.335 | 0.0443 | 0.0699 | 0.4122 | 0.3981 | 0.5881 | 0 |
|      | R/T  | 2/+ | 6/+ | 4/+ | 3/+ | 8/+ | 7/+ | 5/+ | 1 |
| F10  | Mean | 0.0427 | 4.1166 | 0.0173 | 0.0069 | 6.3958×10³ | 0.6925 | 1.2145 | 0.1822 |
|      | Std  | 0.1214 | 68.7625 | 0.2116 | 0.0234 | 6.5686×10⁴ | 2.1067 | 4.9150 | 0.5448 |
|      | R/T  | 3/+ | 7/+ | 2/+ | 1/+ | 8/+ | 5/+ | 6/+ | 4 |
| F11  | Mean | 1202.3 | 1203.1 | 1200.4 | 1202.3 | 1203.3 | 1201.5 | 1200.7 | 1201.5 |
|      | Std  | 6.2334 | 1.2522 | 1.6106 | 3.9568 | 2.4321 | 2.8913 | 1.8405 | 1.1522 |
|      | R/T  | 6/+ | 7/+ | 1/+ | 5/+ | 8/+ | 4/+ | 2/+ | 3 |
| F12  | Mean | 1300.6 | 1303.8 | 1300.5 | 1300.6 | 1300.6 | 1300.6 | 1300.6 | 1300.5 |
|      | Std  | 2.0869 | 1.7531 | 0.6553 | 0.7098 | 0.4312 | 0.3043 | 0.8459 | 0.0700 |
|      | R/T  | 6/+ | 8/+ | 2/+ | 5/+ | 4/+ | 3/+ | 7/+ | 1 |
| F13  | Mean | 1405.5 | 1472.1 | 1400.7 | 1403.0 | 1400.8 | 1400.3 | 1400.7 | 1400.6 |
|      | Std  | 44.3370 | 74.1032 | 8.0383 | 26.9729 | 0.2715 | 0.2175 | 1.9733 | 1.9105 |
|      | R/T  | 7/+ | 8/+ | 4/+ | 6/+ | 5/+ | 1/+ | 3/+ | 2 |

Table 7. Results of fixed-dimension multimodal and composition benchmark functions.

| GWO  | SCA  | PSO  | WOA  | ABC  | TSA  | MVO  | IGWO  |
|------|------|------|------|------|------|------|-------|
| mean | 0.0014 | 1.00×10⁻⁵ | 0.0057 | 6.09×10⁻⁴ | 6.22×10⁻⁴ | 3.83×10⁻⁴ | 0.0034 | 3.33×10⁻⁴ |
| F14  | sd   | 0.0195 | 4.23×10⁻⁴ | 0.0335 | 0.0015 | 4.51×10⁻⁴ | 2.51×10⁻⁴ | 0.0366 | 2.58×10⁻⁴ |
| R/T  | 6/+ | 5/+ | 8/+ | 3/+ | 4/+ | 2/+ | 7/+ | 1 |
Table 8. Overall Wilcoxon rank-sum test results and mean rank results.

| Result | GWO | SCA | PSO | WOA | ABC | TSA | MVO | IGWO |
|--------|-----|-----|-----|-----|-----|-----|-----|------|
|        | +/- | 18/0/2 | 20/0/0 | 16/0/4 | 17/1/2 | 17/0/3 | 16/0/4 | 19/0/1 |
| Mean rank | 3.75 | 6.05 | 4.15 | 4.1 | 6 | 4.2 | 5.4 | 1.7 |
| Overall rank | 2 | 8 | 4 | 3 | 7 | 5 | 6 | 1 |

4.2.1. Analysis of Numerical Results

Referring to the results presented in Tables 6 and 7, IGWO hits the best results on thirteen benchmark functions out of twenty functions (13/20), then TSA both hit three out of twenty (3/20), then PSO and ABC hits two out of twenty (2/20), then GWO and WOA hit one out of twenty (1/20). Finally, SCA and MVO obtain zero out of twenty (0/22). As shown in Table 6, IGWO overtakes all the comparison algorithms on unimodal functions except F5. Particularly, IGWO reaches the theoretical optimal values on F1–F4. For the multimodal functions (F7–F10), IGWO beats other contrast meta-heuristic algorithms on F7–F9, and converges to the theoretical optimal values on F7 and F9. For F8, compared with GWO, the search accuracy of IGWO is improved by 2 orders of magnitude. With respect to the shifted and rotated multimodal functions (F11–F13), IGWO obtains the best result among other meta-heuristic algorithms on F12. PSO and TSA do best on F11 and F13 respectively, whereas IGWO ranks third on F11 and second on F13, and the results are close to the best.

For fixed-dimension functions (F14–F17), IGWO obtains the best result on F14 and reaches the theoretical optimal values on F15 and F16. TSA ranks the first on F17, but the result gained by IGWO is also in good quality. The results acquired by IGWO are closer to the optimal solution on the composite functions (F18–F20), and the search performance is more stable by referring to the corresponding Std. Besides, compared with the original GWO, IGWO shows competitive performance except for F5 and F10, which is inseparable with the strategies proposed in this paper.

Table 8 exhibits the overall Wilcoxon rank-sum test results and average ranking of the algorithms according to Tables 6 and 7. The average ranking of IGWO is 1.7, manifesting that IGWO has outstanding search ability in function optimization problems compared with the other seven algorithms. Furthermore, we can reach the conclusion that
the performance ranking of the algorithms in this part is IGWO > GWO > WOA > PSO > TSA > MVO > ABC > SCA.

4.2.2. Analysis of Convergence Curve

Figures 14–21 illustrate the convergence curves of IGWO and seven contrast algorithms on partial functions (F1, F2, F7, F9, F12, F14, F18 and F20). The number of iterations of each algorithm is set to 1000, and the value corresponding to each iteration in the curve is the mean value of thirty independent experimental results.

**Figure 14.** Comparison results on F1.

**Figure 15.** Comparison results on F2.

**Figure 16.** Comparison results on F7.
Figure 17. Comparison results on F9.

Figure 18. Comparison results on F12.

Figure 19. Comparison results on F14.

Figure 20. Comparison results on F18.
Figures 14 and 15 indicate that IGWO outperforms all other methods on F1 and F2, where IGWO can converge to the global optimal value at the fastest speed. Meanwhile, WOA reaches the second place by means of faster convergence speed and higher accuracy on F1 and F2. Figure 16 reveals that the performance of IGWO is significantly better than others on F7, and WOA achieves similar search accuracy to IGWO. It’s worth mentioning that WOA traps into the local minimums during the 200th iteration to the 500th while IGWO converges rapidly. Therefore, IGWO algorithm has better local optimum avoidance ability. The preeminence of IGWO can be noted in Figure 17 in terms of the search accuracy and speed on multimodal function F9. Figure 18 shows that all algorithms obtain similar results and IGWO obtains slightly better result on F12. As shown in Figure 19, IGWO and TSA all achieve better results on F14, while the result obtained by IGWO owns slightly higher accuracy. Figures 20 and 21 indicate that solution accuracy gained by IGWO ranks first on the composite functions F18 and F20, with the obvious speed advantage contemporarily. Figures 14–21 vividly show the search performance of the eight methods, certifying that the proposed IGWO has excellent performance on most functions.

5. Application of IGWO in Robot Path Planning

To verify the feasibility of IGWO to solve the robot path planning problem, this section designs two terrains to simulate the environment where the robot works, one is simple and the other is complex. The proposed IGWO is compared with MEGWO [59] and RMPSO [60] on the experiments. In addition, we introduce cubic spline interpolation to smooth the planned track, making it more in line with the dynamics characteristics of robots and increasing application value.

5.1. Environment Models

To simplify the problem, we make two assumptions: (a) Since the area of a circle is larger than that of a square when the perimeters are equal, the obstacle is set as a circle; (b) The size of the robot is added to the radius of the obstacle, so that the robot can be regarded as a mass point. Based on the above assumptions, the two kinds of environmental models are designed as shown in Figure 22. In the environmental models, the starting points, ending points and obstacles are represented by yellow squares, green pentagrams and black circles, respectively. The simple obstacle environment model consists of three obstacles, with the starting point set as (0, 0) and the end point set as (4, 6), while the complex environment model contains nine obstacles, with the starting point at (0, 0) and the end point at (10, 10). Additionally, the Euclidean distance between the beginning and the end is considered to be the shortest path length. The information of the two terrains is listed in Table 9. The mathematical expression of the obstacle in the coordinate system is defined as follows:
\[ r^2 = (x - x_{\text{obs}})^2 + (y - y_{\text{obs}})^2 \]  

(26)

where \( r \) is the radius of the obstacle, and \((x_{\text{obs}}, y_{\text{obs}})\) represents the coordinates of the center of the obstacle.

Figure 22. Basic environment models. (a) Simple obstacle environment model (case 1); (b) Complex obstacle environment model (case 2).

Table 9. Environmental model information.

| Simple Environment (Case 1) | Complex Environment (Case 2) |
|-----------------------------|-----------------------------|
| Obstacles                   | 3                           | 9                           |
| Starting point              | (0,0)                       | (0,0)                       |
| Ending point                | (4,6)                       | (10,10)                     |
| The shortest length         | 7.21                        | 14.14                       |

5.2. Path Smoothing

Wang et al. [61] smoothed the planned path by utilizing inner arcs to ensure the continuity of robot motion. Liu et al. [62] pointed out that using the cubic spline interpolation to smooth path has significant virtues compared with circular arcs or straight line and less constraints compared with fillet fitting. Therefore, we adopt the cubic spline interpolation to smooth the trajectory. Cubic spline interpolation is a piecewise interpolation method, which obtains a smooth curve through a series of interpolation points based on cubic polynomials. The interpolation process can be described as follows:

Take \( n + 1 \) nodes on a given interval \([a, b]\), then the interval can be divided into \( n \) subintervals \([x_i, x_{i+1}]\), namely segmentation process. And the value of the node is given in advance as follows:

\[ f(x_i) = f_i, \quad i = 0,1,2,...,n \]  

(27)

Suppose there is a function \( S(x) \), if \( S(x) \) satisfies the following conditions:

1. \( S(x), S'(x), S''(x) \) is continuous.
2. \( S(x_i) = f_i, \quad i = 0,1,2,...,n \)
3. \( S(x) = S_i(x), \quad x \in [x_i, x_{i+1}] \) is a cubic polynomial.

Then \( S(x) \) is called a cubic spline interpolation function, whose curve can be utilized to generate the required curve. Furthermore, on each subinterval \([x_i, x_{i+1}]\), \( S(x) \) is defined as follows:

\[ S(x) = a_ix^3 + b_ix^2 + c_ix + d_i, \quad i = 0,1,2,...,n - 1 \]  

(28)
where \( a, b, c \) and \( d \) are undetermined coefficients. As \( S(x) \) has \( 4n \) undetermined coefficients, there is a need for at least \( 4n \) known conditions to solve the undetermined coefficients, and the specific conditions are given as follows:

\[
\begin{align*}
&n + 1: S(x_i) = f_i, \quad i = 0, 1, 2, ..., n \\
&n - 1: S'(x_i) = S'(x_i), \quad i = 1, 2, ..., n - 1 \\
&n - 1: S''(x_i) = S''(x_i), \quad i = 1, 2, ..., n - 1
\end{align*}
\]

The other two conditions can be obtained by boundary conditions. The commonly used three boundary conditions can be described as follows:

1. Clamped Spline: \( S'(x_0) = \theta, S'(x_n) = 0 \).
2. Natural Spline: \( S''(x_0) = S''(x_n) = 0 \).
3. Not-A-Knot Spline: \( S''(x_0) = S''(x_n) = 0 \).

The proposed IGWO is applied to robot path planning problems. The common points between adjacent subintervals are considered as path nodes, and each node represents a turn along the planned path. Thus, we take the number of path nodes as the dimension of a grey wolf, that is, an individual of the wolf pack represents a candidate path.

Suppose \( D \) path nodes are given, and their coordinates are \((x_0, y_0), (x_1, y_1), ..., (x_n, y_n)\), where the starting and ending coordinates are \((x_0, y_0)\) and \((x_n, y_n)\). Firstly, split the abscissa and ordinate of the above \( D + 2 \) points into the sets of \([x_0, x_n, x_{n+1}, x_{n+2}, ..., x_D]\) and \([y_0, y_{n+1}, y_{n+2}, ..., y_D]\). Then, apply cubic spline interpolation to \([x_0, x_n, x_{n+1}, x_{n+2}, ..., x_D]\) and \([y_0, y_{n+1}, y_{n+2}, ..., y_D]\) separately and obtain the abscissa and the ordinate of \( n \) interpolation points, namely \([x_1, x_2, ..., x_n]\) and \([y_1, y_2, ..., y_n]\). Finally, \([(x_0, y_0), (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)]\) is the path of the robot after smoothing.

5.3. Construction of Fitness Function

The purpose of robot path planning is to plan a shortest path without collision with obstacles from the starting point \((x_0, y_0)\) to the end point \((x_n, y_n)\) [63]. Therefore, a fitness function is constructed to measure the performance of a robot obstacle avoidance and path length in this subsection. The mathematical model is defined as follows:

\[
f = f_i * (1 + \lambda * f_{abs})
\]

where \( \lambda \) is a penalty factor employed to exclude the candidates through the obstacle area. \( f_i \) is the sum of Euclidean distance between the interpolation points and the calculation method is shown in Equation (30). \( f_{abs} \) is a marker variable to evaluate the path obstacle avoidance level, whose initial value is 0. \( f_{abs} \) can be calculated using Equations (31)–(33).

\[
f_i = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}
\]

\[
d_i = \sqrt{(x_i - x_{obs_k})^2 + (y_i - y_{obs_k})^2}, \quad i = 1, 2, ..., n, \quad k = 1, 2, ..., n_{obs}
\]

\[
\varepsilon_k = \frac{1}{n} \sum_{i=1}^{n} \text{MAX} \left(1 - d_i / r_i, 0\right), \quad k = 1, 2, ..., n_{obs}
\]

\[
f_{abs} = \sum_{k=1}^{n_{obs}} \varepsilon_k
\]

where \( x_i \) and \( y_i \) respectively represent the \( x \)-coordinate and \( y \)-coordinate of the \( i \)-th interpolation point on a path. \((x_{obs_k}, y_{obs_k})\) is on behalf of the center coordinate of the \( k \)-th obstacle. \( n \) is the number of interpolation points of the path. \( n_{obs} \) is the total number of obstacles and \( r_i \) is the corresponding radius of the obstacle. Equation (31) is to deal with the Euclidean distance between the interpolation point on the candidate path and the center of \( k \)-th obstacle. Equation (32) could determine whether the path intersects with the \( k \)-th obstacle. If there is a path point entering the \( k \)-th obstacle, then \( \varepsilon_k > 0 \). Otherwise, \( \varepsilon_k = 0 \). Moreover, if a planned path avoids all the obstacles successfully, \( f_{abs} = 0 \).
5.4. Experimental Environment and Parameter Setting

To pursue the objective and fair experimental results, RMPSO, MEGWO and IGWO all use the same software and hardware platform. The population size, the number of path nodes (i.e., individual dimension), number of interpolation points, and the maximum of iterations of the three algorithms remain consistent, as shown in Table 10.

Table 10. Experimental parameter configuration.

| Simple Environment (Case 1) | Complex Environment (Case 2) |
|-----------------------------|------------------------------|
| Population size             | 30                           |
| Path points                 | 2                            |
| Interpolation points        | 100                          |
| Iterations                  | 100                          |

5.5. Analysis of Path Planning Results

5.5.1. Single Contrast Experiment

In this subsection, IGWO is applied to robot path planning and compared with MEGWO and RMPSO to verify the superiority of IGWO in solving robot path planning problems. The simulation results of the two cases are shown in Figures 23–26. The schematic diagrams of the routes planned by the three algorithms are exhibited in Figures 23 and 24. Figures 25 and 26 show the trend of the path length with the number of iterations.

Figure 23. Comparison of three algorithm in case 1, where the yellow square and green five-pointed star respectively represent the starting point and ending point.

Figure 24. Comparison of three algorithm in case 2, where the yellow square and green five-pointed star respectively represent the starting point and ending point.
Figures 23 and 24 indicate that as the environment gets more complex, IGWO generates the path closest to the optimal in both cases. Additionally, Figure 25 demonstrates that IGWO can generate a shorter path compared with RMPSO and MEGWO in case 1. It can be seen from Figure 26 that when the number and density of environmental obstacles are increased, IGWO keeps its strengths of the path length. Therefore, the improved algorithm can plan shorter paths in simple and complex environment and has better path optimization performance.

5.5.2. Thirty Independent Contrast Experiments

Since the results of the single experiment may be accidental, this subsection, we use three algorithms to conduct thirty independent experiments in two obstacle environments, which makes up for the deficiency of the single experiment. Simulation experiment results of path planning in two terrains are shown in Figures 27 and 28. The statistical results are listed in Tables 11 and 12, where Mean, Best, Worst represent the average, the shortest and the longest length of the thirty planned paths, respectively. Unsafe path represents the number of paths that touch obstacles in the thirty independent tests. Success rate signifies the percentage of secure paths obtained in thirty independent experiments. To visualize the differences between the three algorithms, we convert the experimental results into Figures 29 and 30, where three indexes (Mean, Best, and Worst) are subtracted from the shortest path length listed in Table 9 to observe the gaps with the shortest path more conveniently. Moreover, Figures 31 and 32 exhibit the path length obtained by three methods in thirty experiments, which make it easier to compare the performance of these methods in each experiment.
Figure 27. Results of the three algorithms in case 1, where the yellow square, green five-pointed star and lines in different colors in the figure represent the starting point, ending point and paths, respectively. (a) Results of RMPSO; (b) Results of MEGWO; (c) Results of IGWO.

Figure 28. Results of the three algorithms in case 2, where the yellow square, green five-pointed star and lines in different colors in the figure represent the starting point, ending point and paths, respectively (a) Results of RMPSO; (b) Results of MEGWO; (c) Results of IGWO.

Table 11. Results comparison in case 1, where the best values are highlighted in bold.

|        | Mean   | Best   | Worst  | Unsafe Path | Success Rate |
|--------|--------|--------|--------|-------------|--------------|
| RMPSO  | 7.8486 | 7.8308 | 7.8989 | 3           | 90%          |
| MEGWO  | 7.9557 | 7.7764 | 8.3557 | 3           | 90%          |
| IGWO   | 7.6529 | 7.5669 | 7.6981 | 0           | 100%         |

Table 12. Results comparison in case 2, where the best values are highlighted in bold.

|        | Mean   | Best   | Worst  | Unsafe Paths | Success Rate |
|--------|--------|--------|--------|--------------|--------------|
| RMPSO  | 15.5894| 14.9284| 17.5323| 4            | 86.67%       |
| MEGWO  | 16.1491| 14.5662| 17.6817| 3            | 90%          |
| IGWO   | 14.7052| 14.5691| 14.8698| 0            | 100%         |
Figure 29. The bar chart of the results in case 1.

Figure 30. The bar chart of the results in case 2.

Figure 31. The path length curves in case 1.
Figure 32. The path length curves in case 2.

For the performance of the algorithms in case 1, Figure 27 shows that the planned trajectory of RMPSO and IGWO are relatively stable, while the thirty trajectories of MEGWO fluctuate slightly. Table 11 shows that the average, shortest, and longest length of the paths planned by IGWO are all the smallest among the three algorithms. Furthermore, RMPSO and MEGWO generate three failed paths in the thirty independent experiments, and IGWO plans thirty safe paths successfully, indicating that IGWO possesses better stability. Figure 29 turns the statistical results into a bar graph, more intuitively showing the superiority of IGWO in case 1.

With respect to case 2, Figure 28 denotes that with the increase of obstacles, the gaps between the three algorithms become prominent. The behaviors of RMPSO and MEGWO appear obvious jitter, while IGWO plans the path in more efficient and safer manner. Table 12 illustrates that although MEGWO plans the shortest path, IGWO owns a better performance in the remaining indicators. It is worth mentioning that the success rate of IGWO maintains in 100% despite the increased complexity of the environment, while the success rates of RMPSO and MEGWO are 86.67% and 90% respectively. In addition, the statistical data presented in Figure 30 vividly signifies the better search stability and accuracy of IGWO.

Figures 31 and 32 exhibit the length of thirty paths obtained by each method. It can be observed that there are obvious differences in the performance of the three methods. In case 1, IGWO gets the optimal path in every attempt, and the path length is the most stable, while MEGWO has apparent oscillation. For case 2, IGWO obtains the maximum number of optimal paths compared with RMPSO and MEGWO. Even if the environment complexity is increased, the length of the paths generated by IGWO is stably between 14.5 and 15. In contrast, the path length obtained by RMPSO and MEGWO are sometimes less than 15 and sometimes more than 17. Therefore, it is concluded that the trajectories generated by IGWO are both the shortest and the safest.

5.6. Contrast Experiment in Complex Environment with Irregular Obstacles

To further test the capability of the proposed IGWO to solve the path planning problem, a 50 × 50 complex environment model with more irregular obstacles is designed in this section, as shown in Figure 33. Inspired by the polygonal obstacles designed by Dai et al. [64], 18 irregular-shaped obstacles are used in the new environment model. The path starts at (10, 30) and ends at (25, 12).
Beiravand et al. [65] provides the standards and guidelines for researchers to test the performance of optimization algorithms. In this section, comparison experiments are designed referring to [65]. To examine the performance of the proposed IGWO, RMPSO, MEGWO and mGWO are selected as the comparative algorithms. To ensure the fairness and objectivity of the comparison experiments, all experiments are run on CPU Core i5-9400F and 8GB RAM, and programmed on MATLAB R2018b. To avoid randomness of results, all algorithms are run independently for 30 times. Beiravand et al. indicated that algorithm performance metrics can be classified into three categories: efficiency, solution quality and reliability [65]. Therefore, Iteration, Path length and Success Rate are used as performance metrics in this section. The final results are shown in Table 13, with the optimal values among the four algorithms marked in bold. Iteration is the number of iterations when the algorithm obtains the final result, and Path length represents the length of the final path obtained by the algorithm. The values of Iteration and Path length take the average of the results of 30 independent experiments. Success Rate is equal to the ratio of the number of safe paths planned by the algorithm to the total number of independent experiments.

Figure 33. Paths planned by four algorithms in complex obstacle environment. (a) Results of RMPSO; (b) Results of MEGWO; (c) Results of mGWO; (d) Results of IGWO.
Table 13. Experimental results in complex obstacle environment, and the best values are highlighted in bold.

| Algorithm | Iteration | Path Length | Success Rate |
|-----------|-----------|-------------|--------------|
| RMPSO     | 96        | 36.289      | 86.67%       |
| MEGWO     | 93        | 40.1963     | 83.33%       |
| mGWO      | 101       | 39.3321     | 80%          |
| IGWO      | 72        | 31.8779     | 90%          |

Figure 33 shows the trajectories planned by the four algorithms, visually demonstrating that the proposed IGWO is capable of planning the shorter safe path. Table 13 shows that IGWO achieves the best results in all three metrics compared to the comparative algorithms. In terms of the metric Iteration, the average number of iterations taken by IGWO is 72, which is 24, 21 and 29 generations earlier than RMPSO, MEGWO and mGWO, respectively, indicating that IGWO can search for safe paths more efficiently. In terms of metric Path length, IGWO plans shorter paths among the four algorithms. Since a shorter path length means lower energy consumption, the paths obtained by IGWO are of higher quality. In addition, the planning success rate of IGWO is 90%, ahead of the other algorithms, implying that the IGWO algorithm has higher safety and stability under complex environment model.

6. Conclusions and Future Work

In this paper, an improved grey wolf optimizer, namely IGWO, is proposed and applied to robot path planning. The proposed IGWO uses the three well-designed enhancement strategies to balance the global and local search abilities and settle the defects of prematurity and slow convergence to some extent. More exactly, a modified position update mechanism with an improved updating formula is firstly proposed to strengthen the leadership hierarchy of the pack and coordinate the global and local searches more reasonably in the meantime. Simultaneously, the population opposition-based learning and Cauchy mutation are introduced to ensure the diversity of the algorithm population. Then, inspired by the random walk strategy, a dynamic local optimum escape strategy is designed to help the algorithm jump out of the local stagnation. Finally, the bad individual repositioning method is proposed to effectively accelerate the convergence.

IGWO is testified on twenty benchmark functions, and first compared with five state-of-the-art GWO variants, and then seven other well-known meta-heuristic algorithms. Furthermore, Wilcoxon rank-sum test is utilized to judge the significance difference between IGWO and the comparison algorithm. Empirical results show that IGWO outperforms the competitors on most functions. Although IGWO misses the optimal solutions on some functions, by reference to No Free Lunch (NFL), no algorithm can always obtain optimal results in all fields. Afterwards, IGWO is used to deal with robot path planning problems, and cubic spline interpolation is imported to smooth the trajectory of robot. Experimental results reveal that IGWO is a reliable method to solve robot path planning problems, whether in the simple environment or complex environment.

For future work, we will conduct further research in terms of the following issues: First, IGWO will be applied to more challenging terrains, taking the dynamic obstacles into consideration. Second, it’s worth a try to carry out practical experiments to testify the application value and practical significance of the algorithm. Finally, combining GWO with other excellent meta-heuristic algorithms, and the hybrid algorithm may have a more promising performance.
Author Contributions: Conceptualization, L.D.; formal analysis, X.Y. (Xianfeng Yuan) and B.Y.; investigation, L.D. and X.Y. (Xianfeng Yuan); methodology, L.D. and X.Y. (Xianfeng Yuan); resources, X.Y. (Xianfeng Yuan), Y.S. and Q.X.; software, X.Y. (Xianfeng Yuan), B.Y., Q.X. and X.Y. (Xiongyan Yang); supervision, Y.S.; validation, L.D.; visualization, L.D.; writing—original draft, L.D.; writing—review & editing, X.Y. (Xianfeng Yuan). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Natural Science Foundation of China (Grant No. 61803227, 61773242, 61973184), Independent Innovation Foundation of Shandong University (Grant No. 2018ZQXM005), Young Scholars Program of Shandong University, Weihai (Grant No. 2018ZQXM005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict 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.

References

1. Cui, Z.; Sun, H.M.; Yu, J.T.; Yin, R.; Jia, R. Fast detection method of green peach for application of picking robot. Appl. Intel. 2022, 52, 1718–1739.
2. Shin, H.G.; Park, I.; Kim, K.; Kim, H.K.; Chung, W.K. Corneal suturing robot capable of producing sutures with desired shape for corneal transplantation surgery. IEEE Trans. Robot. 2021, 37, 304–312.
3. Zhang, M.; Tian, G.; Zhang, Y.; Duan, P. Service skill improvement for home robots: Autonomous generation of action sequence based on reinforcement learning. Knowl. Based Syst. 2021, 212, 106605.
4. Lv, Z.; Qiao, L. Deep belief network and linear perceptron based cognitive computing for collaborative robots. Appl. Soft Comput. 2020, 92, 106300.
5. Yao, W.; de Marina, H.G.; Lin, B.; Cao, M. Singularity-free guiding vector field for robot navigation. IEEE Trans. Robot 2021, 37, 1206–1221.
6. Li, C.; Tian, G. Transferring the semantic constraints in human manipulation behaviors to robots. Appl. Intel. 2020, 50, 1711–1724.
7. Gao, Y.; Wei, W.; Wang, X.; Yu, Q. Feasibility, planning and control of ground-wall transition for a sectorial hexapod robot. Appl. Intel. 2021, 51, 5506–5524.
8. Tognon, M.; Alami, R.; Siciliano, B. Physical human-robot interaction with a tethered aerial vehicle: Application to a force-based human guiding problem. IEEE Trans. Robot 2021, 37, 723–734.
9. Szczepanski, R.; Tarczewski, T. Global path planning for mobile robot based on artificial bee colony and dijkstra’s algorithms. In Proceedings of the IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021; pp. 725–730.
10. Deng, X.; Li, R.; Zhao, L.; Wang, K.; Gui, X. Multi-obstacle path planning and optimization for mobile robot. Expert. Syst. Appl. 2021, 183, 115445.
11. Liu, J.; Yang, J.; Liu, H.; Tian, X.; Gao, M. An improved ant colony algorithm for robot path planning. Soft Comput. 2021, 25, 3829–3839.
12. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
13. Dorigo, M.; Maniezzo, V.; Colomi, A. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Evol. Comput. 1997, 1, 53–66.
14. Yang, X.S.; Suash, D. Cuckoo search via Levy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, 9–11 December 2009; pp. 210–214.
15. Yang, X.S. A new metaheuristic bat-inspired algorithm. Comput. Knoval. Technol. 2009, 284, 65–74.
16. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. Adv. Eng. Softw. 2014, 69, 46–61.
17. Wu, J.; Srivastava, J.; Yun, U.; Tayeb, S.; Lin, J.C. An evolutionary computation-based privacy-preserving data mining model under a multithreshold constraint. Trans. Emerg. Telecommun. Technol. 2021, 32, e4209.
18. Wu, Z.; Karimi, H.R.; Dang, C. An approximation algorithm for graph partitioning via deterministic annealing neural network. Neural Netw. 2019, 117, 191–200.
19. Wu, Z.; Gao, Q.; Jiang, B.; Karimi, H.R. Solving the production transportation problem via a deterministic annealing neural network method. Appl. Math. Comput. 2021, 411, 126518.
20. Zivkovic, M.; Stoean, C.; Chhabra, A.; Budimirovic, N.; Petrovic, A.; Bacanin, N. Novel improved salp swarm algorithm: An application for feature selection. Sensors 2022, 22, 1711.

21. Singh, N.; Son, L.H.; Chichlana, F.; Magnet, J.P. A new fusion of salp swarm with sine cosine for optimization of non-linear functions. Eng. Comput. 2020, 36, 185–212.

22. Qiang, N.; Gao, J.; Gao, F. Multi-robots global path planning based on PSO algorithm and cubic spline. J. Syst. Simul. 2017, 29, 1397–1404.

23. You, D.; Kang, Y.; Liu, P.; Hu, Y. A path planning method for mobile robot based on improved grey wolf optimizer. Mach. Tool Hydraul. 2021, 49, 1–6.

24. Xu, L.; Fu, W.; Jiang, W.; Tao, Z. Mobile robots path planning based on 16-directions 24-neighbourhoods improved ant colony algorithm. Control. Decis. 2021, 36, 1137–1146.

25. Song, B.; Wang, Z.; Zou, L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. Appl. Soft Comput. 2021, 100, 106960.

26. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. Appl. Soft Comput. 2021, 107, 107376.

27. Song, P.C.; Pan, J.S.; Chu, S.C. A parallel compact cuckoo search algorithm for three-dimensional path planning. Appl. Soft Comput. 2020, 94, 106443.

28. Das, P.K.; Jena, P.K. Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. Appl. Soft Comput. 2020, 92, 106312.

29. Medjahed, S.A.; Saadi, T.A.; Benyettou, A.; Ouali, M. Gray wolf optimizer for hyperspectral band selection. Appl. Soft Comput. 2016, 40, 178–186.

30. Hou, Y.; Gao, H.; Wang, Z.; Du, C. Improved gray wolf optimization algorithm and application. Sensors 2022, 22, 3810.

31. Wang, S.; Yang, X.; Wang, X.; Qian, Z. A virtual force algorithm-lévy-embedded grey wolf optimization algorithm for wireless sensor network coverage optimization. Sensors 2019, 19, 2735.

32. Jarray, R.; Al-Dhaifallah, M.; Rezk, H.; Bouallègue, S. Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles. Sensors 2022, 22, 1826.

33. Yue, Z.; Zhang, S.; Xiao, W. A novel hybrid algorithm based on grey wolf optimizer and fireworks algorithm. Sensors 2020, 20, 2147.

34. Long, W.; Cai, S.; Jiao, J.; Xu, M.; Wu, T. A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. Energy Convers. Manag. 2020, 203, 112243.

35. Peng, T.; Zhou, B. Hybrid bi-objective gray wolf optimization algorithm for a truck scheduling problem in the automotive industry. Appl. Soft Comput. 2019, 81, 105513.

36. Zhang, C.; Wang, W.; Pan, Y. Enhancing electronic nose performance by feature selection using an improved grey wolf optimization based algorithm. Sensors 2020, 19, 1577.

37. Goel, T.; Murugan, R.; Mirjalili, S.; Chakraborty, D.K. OptCoNet: An optimized convolutional neural network for an automatic diagnosis of COVID-19. Appl. Intell. 2021, 51, 1351–1366.

38. Zhang, Q.; Wang, R.; Yang, J.; Lewis, A.; Chichlana, F.; Yang, S. Biology migration algorithm: A new nature-inspired heuristic methodology for global optimization. Soft Comput. 2019, 23, 7333–7358.

39. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. Expert Syst. Appl. 2021, 166, 113917.

40. Qu, C.Z.; Gai, W.D.; Zhong, M.Y.; Zhang, J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. Appl. Soft Comput. 2020, 89, 106099.

41. Heidari, A.A.; Abbaspour, R.A.; Chen, H.L. Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. Appl. Soft Comput. 2019, 81, 105521.

42. Heidari, A.A.; Pahlavani, P. An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Appl. Soft Comput. 2017, 60, 115–134.

43. Luo, K.P. Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey. Appl. Soft Comput. 2019, 77, 225–235.

44. Liu, J.; Wei, X.; Huang, H. An improved grey wolf optimizer and its application in path planning. IEEE Access 2021, 9, 121944–121956.

45. Saxena, A.; Kumar, R.; Das, S. β-chaotic map enabled grey wolf optimizer. Appl. Soft Comput. 2019, 75, 84–105.

46. Zhao, X.; Fang, Y.; Liu, L.; Li, J.; Xu, M. An improved moth-flame optimization algorithm with orthogonal opposition-based learning and modified position updating mechanism of moths for global optimization problems. Appl. Intell. 2020, 50, 4434–4458.

47. Wang, Z.; Ding, H.; Yang, Z.; Li, B.; Guan, Z.; Bao, L. Rank-driven salp swarm algorithm with orthogonal opposition-based learning for global optimization. Appl. Intell. 2022, 52, 7922–7964.

48. Mahdavi, S.; Rahnamayan, S.; Deb, K. Opposition based learning: A literature review. Swarm Evol. Comput. 2018, 39, 1–23.

49. Miao, F.; Yao, L.; Zhao, X. Symbiotic organisms search algorithm using random walk and adaptive Cauchy mutation on the feature selection of sleep staging. Expert Syst. Appl. 2021, 176, 114887.

50. Miao, Z.; Yuan, X.; Zhou, F.; Qiu, X.; Song, Y.; Chen, K. Grey wolf optimizer with an enhanced hierarchy and its application to the wireless sensor network coverage optimization problem. Appl. Soft Comput. 2020, 96, 106602.
51. Martin, B.; Marot, J.; Bourennane, S. Mixed grey wolf optimizer for the joint denoising and unmixing of multispectral images. *Appl. Soft Comput.* **2019**, *74*, 385–410.

52. Gupta, S.; Deep, K. A memory-based grey wolf optimizer for global optimization tasks. *Appl. Soft Comput.* **2020**, *93*, 106367.

53. Gupta, S.; Deep, K. A novel random walk grey wolf optimizer. *Swarm Evol. Comput.* **2019**, *44*, 101–112.

54. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471.

55. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133.

56. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67.

57. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2015**, *27*, 495–513.

58. Kaur, S.; Awasthi, L.K.; Sangal, A.L.; Dhiman, G. Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541.

59. Tu, Q.; Chen, X.; Liu, X. Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Appl. Soft Comput.* **2019**, *76*, 16–30.

60. Jana, B.; Mitra, S.; Acharyya, S. Repository and mutation based particle swarm optimization (RMPSO): A new PSO variant applied to reconstruction of gene regulatory network. *Appl. Soft Comput.* **2019**, *74*, 330–355.

61. Wang, H.; Yin, P.; Zheng, W. Mobile robot path planning based on improved A* algorithm and dynamic window method. *Robot* **2020**, *42*, 92–99.

62. Liu, J.; Ji, H.; Li, Y. Robot path planning based on improved bat algorithm and cubic spline interpolation. *Acta Autom. Sin.* **2021**, *47*, 1710–1719.

63. Zheng, F.; Zecchin, A.C.; Newman, J.P. An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems. *IEEE Trans. Evol. Comput.* **2017**, *21*, 773–791.

64. Dai, J.; Qiu, J.; Yu, H.; Zhang, C.; Wu, Z.; Gao, Q. Robot Static Path Planning Method Based on Deterministic Annealing. *Machines* **2022**, *10*, 600.

65. Beiranvand, V.; Hare, W.; Lucet, Y. Best practices for comparing optimization algorithms. *Optim. Eng.* **2017**, *18*, 815–848.