IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 12, 2019, accepted August 3, 2019, date of publication August 7, 2019, date of current version August 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933661

An Adaptive Whale Optimization Algorithm Using Gaussian Distribution Strategies and Its Application in Heterogeneous UCAVs Task Allocation

YINTONG LI[1], TONG HAN[1], HUI ZHAO[1], AND HANJIE GAO[2]
[1]College of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China
[2]College of Computer Science and Technology, Jilin University, Changchun 130000, China

Corresponding author: Yintong Li (yintongli0007@163.com)

This work was supported in part by the Natural Science Foundation of Shaanxi Province under Grant 2017JM6078, and in part by the Aeronautical Science Foundation of China under Grant 20175196019.

ABSTRACT To overcome the defect of whale optimization algorithm (WOA) being easily fallen into local optimum caused by the ill-distribution of solutions, this paper explores an adaptive WOA variant using Gaussian distribution strategies (GDSs), named GDS-WOA. In GDS-WOA, by means of one GDS, named the Gaussian estimation of distribution method, the superior population information is used to evolve the distribution scope and modify the evolution direction. Moreover, an adaptive framework is adopted to integrate the Gaussian estimation of distribution method and WOA together, in which each individual can update its position using Gaussian estimation of distribution method or WOA according to an adaptive probability parameter. When the algorithm falls into stagnation, another GDS, named Gaussian random walk, is activated to enrich the population diversity and help the algorithm get rid of the local optimum. Additionally, the greedy strategy is carried out to select the offspring from the parents and the generated candidates to fully retain the promising solutions. The GDS-WOA is benchmarked on CEC 2014 test suite, and the performance of GDS-WOA is evaluated by comparing with WOA and its promising variant IWOA, as well as other five state-of-the-art evolutionary algorithms, i.e., COA, VCS, CoBiDE, HFPSO and GWO. The statistical results demonstrate that GDS-WOA outperforms other competitors in terms of convergence efficiency and accuracy. Finally, GDS-WOA is applied to solve the optimal task allocation problem of heterogeneous unmanned combat aerial vehicles (UCAVs). To address this constrained real-world optimizing problem efficiently, the mathematical model of heterogeneous UCAVs task allocation is described with the operational effectiveness value as the objective. The validity and practicauility of the model as well as the performance of GDS-WOA for solving constrained optimization problem are demonstrated by the experimental results.

INDEX TERMS Whale optimization algorithm, CEC 2014, numerical optimization, UCAV, task allocation.

I. INTRODUCTION

Optimization refers to the process of obtaining a global optimal solution for a problem under the given conditions. The real-world problems in the scientific fields, such as engineering design and economic planning, mostly are multimodal, high-dimensional, disconnected and oscillated optimization problems. These complex problems cannot be solved well within reasonable time using traditional method based on gradient. Therefore, inspired by natural phenomena and animal group behavior characteristics, researchers have proposed many efficient natural heuristic algorithms for high-dimensional complex optimization problems in real-world.

With the rapid developments of meta-heuristic optimization algorithms, they have been applied in various fields over the past years. The Whale Optimization Algorithm (WOA) [1] is a novel population-based meta-heuristic

The associate editor coordinating the review of this article and approving it for publication was Huiping Li.

algorithm proposed by Australian scholar Mirjalili and Lewis in 2016, which is inspired by the hunting behavior of humpback whales. It has the advantages of simple structure and few control parameters. The numerical experiment results in [1] show that WOA has some advantages in terms of convergence efficiency or accuracy compared with particle swarm optimization (PSO) [2], gravity search algorithm (GSA) [3], differential algorithm (DE) [4], rapid evolution programming (FTP) [5], and the adaptive covariance matrix evolution strategy (CMA-ES) [6]. Therefore, WOA has been widely applied to solve real-world problems in a wide range of disciplines [7]–[11]. However, we find that it is easy to fall into local optimum when dealing with high-dimensional complex optimization problems. And the quality of solution obtained by WOA needs to be further improved.

The scholars have found that the population diversity and evolutionary direction play an important role in the optimization performance of meta-heuristic algorithms. However, the population diversity shrinks rapidly in the later stage of optimization process leading WOA easily falling into local optimum. To improve the optimization performance of WOA, relevant theoretical researches have been carried out. According to the literatures we have learned about WOA research, there are two group variants of WOA.

The first group improves the performance of the algorithm by introducing other optimization strategies into WOA. Kaur and Arora [12] adjusted the control parameters by various chaos strategies in WOA to balance exploration and exploitation and improve the convergence accuracy; Hu *et al.* [13] coordinated the impact of the current optimal solution on the population iterative process by introducing inertia weight to increase population diversity; Trivedi *et al.* [14] introduced adaptive technology to improve the convergence efficiency of WOA; Elaziz and Mirjalili [15] introduced differential evolution algorithm and opposition-based learning strategy into WOA to improve the diversity of the population in the process of optimization. The improved algorithm in terms of the local optimal avoidance ability and the local search ability had been improved at the expense of a large amount of computational cost; Khalil *et al.* [16] proposed a distributed implementation of WOA, called MR-WOA, by using Hadoop MapReduce to improve the scalability of WOA for solving large-scale complex problems; For the defect of WOA premature convergence, Chen *et al.* [17] introduced Lévy flight and chaotic local search into WOA to promote the balance exploration and exploitation. These two strategies have been widely used to improve the performance of intelligent optimization algorithms.

Another group hybridizes WOA with other intelligent optimization algorithms. For example, Mafarja and Mirjalili [18] combined WOA with simulated annealing algorithm (SA) [19] to strengthen the exploitation performance of WOA and applied the hybridized algorithm to feature selection problems; Mostafa and Yazdani [20] hybridized WOA with DE by combining great exploitation of WOA with excellent exploration of DE, which had improved the quality of solution and convergence rate; Trivedi *et al.* [21] proposed the PSO-WOA in which PSO was used for local search phase and WOA was used for global search phase; Singh and Hachimi [22] proposed a novel hybrid algorithm by combining WOA and GWO in which the spiral function in WOA was used for exploration phase to cover a broader area in search space. In addition, there are many hybrid varieties of WOA [11], [23]–[26].

Although the previous research works have improved the convergence efficiency or accuracy in certain extent, there is not a variant of WOA that could effectively improve the convergence efficiency and accuracy at the same time, which could obtain great quality solutions when solving complex optimization problems such as the CEC 2014 benchmarks. In addition, most of the hybrid varieties of WOA have low computational efficiency, meaning that the computational time required to obtain a better solution is long. Therefore, it needs continuous improvement and novation. In order to avoid the local optimum in solving complex optimization problems and improve the convergence accuracy of WOA, we propose an adaptive WOA based on Gaussian distribution strategies. First, the Gaussian estimation of distribution method is adopted to evolve the distribution scope and modify the evolution direction. It is noted that the weighted covariance matrix is the core component of Gaussian estimation of distribution method. Moreover, WOA is coupled to the Gaussian estimation of distribution method by an adaptive framework, in which each individual can update its position using Gaussian estimation of distribution method or WOA according to an adaptive probability parameter. The probability parameter is adaptively updated according to the information gathered from the offspring. In addition, the Gaussian random walk is adopted to enrich the population diversity and help the algorithm get rid of the local optimum when the search falls into stagnation. Finally, the greedy strategy is carried out to select the offspring from the parents and the generated candidates to fully retain the domination individuals in order to improve the convergence speed. The simulation results of the CEC2014 test suit and UCAVs task allocation problem show that GDS-WOA has excellent performance in dealing with complex problems.

The rest organization of this paper is as follows: The review of WOA and the mathematical presentation of GDS-WOA are described in Section 2. In Section 3, the statistical results of numerical experiment based on the CEC 2014 test suite are discussed. The heterogeneous UCAVs optimal task allocation model is proposed in Section 4 and GDS-WOA is applied to solve it in Section 5. Finally, we conclude this work and note directions for future study in Section 6.

II. PROPOSED GDS-WOA

A. REVIEW OF THE BASIC WOA

WOA is a novel meta-heuristic algorithm inspired by the predation mechanism of humpback whales. Since the ‘prey’ position is unknown in the search space, WOA assumes that the current best solution obtained so far is the 'prey' position in the optimization process. And the other search individuals update their position toward the current best solution through the encircling prey mechanism and spiral bubble-net feeding maneuver mechanism. The global search and local search of the algorithm are balanced by the convergence factor a, so that the population can search from disorder to order in solution space. And finally the optimal solution of the problem is obtained. The mathematical model of the population locations update is briefly shown as follows:

$$a = 2 - 2t/t_{\max} \tag{1}$$

$$A = 2 \cdot a \cdot r - a \tag{2}$$

$$\boldsymbol{D} = |\boldsymbol{C} \cdot \boldsymbol{X}^*(t) - \boldsymbol{X}(t)| \tag{3}$$

$$C = 2 \cdot r \tag{4}$$

$$\boldsymbol{D}' = |C \cdot \boldsymbol{X}_{rand}(t) - \boldsymbol{X}(t)| \tag{5}$$

$$\boldsymbol{X}(t+1) = \begin{cases} \boldsymbol{X}^*(t) - A \cdot \boldsymbol{D}, \ \ p < 0.5 \ and \ |A| < 1 \ (6.1) \\ \boldsymbol{X}_{rond}(t) - A \cdot \boldsymbol{D}, \ \ p < 0.5 \ and \ |A| \geq 1 \ (6.2) \\ \boldsymbol{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \boldsymbol{X}^*(t), \ \ p \geq 0.5 \ (6.3) \end{cases} \tag{6}$$

where t is the current iteration number; $t_{\max}$ is the maximum iteration number; a is the convergence factor; r, l and p are random numbers between 0 and 1; $X(t)$ is the current individual position, and $X^*(t)$ is the position of the current best solution obtained so far; b is a constant that defines the shape of the spiral path. More details about WOA can be found in [1]. The pseudo code of the WOA is described as algorithm 1.

Algorithm 1 The Procedure of WOA

Step 1: Initialize the population; Set $t = 1$;
Step 2: Calculate the fitness of each search agent;
Step 3: If $t <= t_{max}$
    execute step 4;
  Else
    output $\boldsymbol{X}^*$;
  End the algorithm;
Step 4: Update the best solution X^* and its fitness;
Step 5: For each solution
    Update a, l, p, A, and $\boldsymbol{C}$;
    If $p < 0.5$
      If $|A| < 1$
        $\boldsymbol{X}_i$ is updated using (6.1);
      Else if
        $\boldsymbol{X}_i$ is updated using (6.2);
      End if
    Else
      $\boldsymbol{X}_i$ is updated using (6.3);
    End if
  End for
Step 6: Boundary control; calculate fitness of each agent;
Step 7: $t = t + 1$; execute step 3.

B. MATHEMATICAL PRESENTATION OF GDS-WOA

1) GAUSSIAN ESTIMATION OF DISTRIBUTION METHOD

In 2001, Larrañaga and Lozano [27] proposed the estimation of distribution algorithm (EDA). The EDA uses probabilistic model learning and sampling to estimate the distribution of dominant candidate solutions, which can identify the characteristics of promising solutions and estimate the evolution direction of population and it has promising performance in dealing with some complex problems [28]–[32]. Due to there are no selection, crossover and mutation operations that are the main evolution operations in traditional evolutionary algorithms, EDA and WOA belong to different types of evolutionary algorithms. Thus, we try to introduce the Gaussian estimation of distribution method into WOA for overcoming the defect in WOA.

In the basic WOA, the update of the population during the iterative process is mainly guided by the best solution obtained so far, so that the population diversity shrinks rapidly in the later stage of the optimization process. Thus, WOA is easily fallen into the local optimum. Researches on EDA show that the core component of the algorithm is the weighted covariance matrix and it has strong local optimum avoidance. Therefore, in this paper, we introduce the Gaussian estimation of distribution method into WOA in order to make full use of the promising population information to estimate the better evolution direction. The Gaussian estimation of distribution method based on weighted covariance matrix is as follows:

$$\boldsymbol{X}(t)_{mean} = \sum_{l=1}^{m} \omega_i \cdot \boldsymbol{X}_i \tag{7}$$

$$\omega_i = (\ln(m+1) - \ln(i)) / \sum_{i=1}^{m} (\ln(m+1) - \ln(i)) \tag{8}$$

$$Cov(t) = \frac{1}{m-1} \sum_{i=1}^{m} (\boldsymbol{X}_i(t) - \boldsymbol{X}(t)_{mean})(\boldsymbol{X}_i(t) - \boldsymbol{X}(t)_{mean})^T \tag{9}$$

$$\begin{aligned} \boldsymbol{X}(t+1) = \ & Gaussian(\boldsymbol{X}(t)_{mean}, Cov(t)) \\ & + rand \cdot (\boldsymbol{X}(t)_{mean} - \boldsymbol{X}(t)) \end{aligned} \tag{10}$$

The half of population with better fitness is selected as the promising population. In (7), where $m = SN/2$, SN is the number of individuals, and $X_1, X_2, X_3, \ldots, X_m$ is the m promising solutions with fitness values ranked from high to low. Equation (7) indicates that the m promising solutions are selected to estimate the weighted mean value. From (8), a higher rank means a greater weight. $Cov(t)$ is the weighted covariance matrix of the promising solutions. Population update their location using (10).

2) ADAPTIVE STRATEGY

How to effectively integrate the Gaussian estimation of distribution method based on weighted covariance matrix and WOA together has a crucial impact of the improved algorithm. Inspired by the adaptive framework to tune the coordinate systems in evolutionary algorithms in [33], this paper proposes an adaptive strategy to efficiently embed the Gaussian estimation of distribution method into WOA, in which each individual can update its position using Gaussian estimation of distribution method or WOA according to an adaptive probability parameter. $Pv = (pv_1, pv_2, \ldots, pv_{SN})$ is the probability vector. Since there is no prior information when the algorithm is initialized, this paper sets the equal probability that each individual chooses two strategies, i.e., $Pv = (0.5, 0.5, \ldots, 0.5)$. The probability vector Pv is adaptively updated based on information gathered from the offspring in the optimization process.

In the paper, the information gathered from the offspring includes two parts. One part is which strategy is used to generate the offspring and another part is whether the fitness value of the generated offspring is superior to the parent individual. The collected information is used to guide Pv to update in the following four cases:

1) *WOA is better*: WOA is adopted to generate the offspring and it has better fitness. pv_i is updated as follows

$$pv_i(t+1) = pv_i(t) + 0.1 \cdot (1 - pv_i(t)) \cdot (t/t_{\max}) \tag{11}$$

2) *WOA is worse*: WOA is adopted to generate the offspring and it has worse fitness. pv_i is updated as follows

$$pv_i(t+1) = pv_i(t) - 0.1 \cdot pv_i(t) \cdot (1 - t/t_{\max}) \tag{12}$$

3) *Gaussian estimation of distribution method is better*: Gaussian estimation of distribution method is adopted to generate the offspring and it has better fitness. pv_i is updated as follows

$$pv_i(t+1) = pv_i(t) - 0.1 \cdot pv_i(t) \cdot (1 - t/t_{\max}) \tag{13}$$

4) *Gaussian estimation of distribution method is worse*: Gaussian estimation of distribution method is adopted to generate the offspring and it has worse fitness. pv_i is updated as follows

$$pv_i(t+1) = pv_i(t) + 0.1 \cdot (1 - pv_i(t)) \cdot (t/t_{\max}) \tag{14}$$

3) GAUSSIAN RANDOM WALK STRATEGY

During the iterations, the average fitness of the promising solutions is used to judge whether the search is stagnant. If the average fitness does not change in two consecutive iterations, the algorithm is regarded as stagnating. For getting rid of the local optimum and overcoming the premature convergence of the algorithm, the Gaussian random walk strategy is used to generate the new candidates. The model is as follows:

$$\boldsymbol{X}(t+1) = Gaussian(\boldsymbol{X}(t), \sigma_1) \tag{15}$$

$$\sigma_1 = \cos(\pi \cdot t/(2 \cdot t_{max})) \cdot (\boldsymbol{X}(t) - \boldsymbol{X}_r^*(t)) \tag{16}$$

where X_r^* is a promising solution randomly selected from the promising population, In (16), the step size of the Gaussian random walk is coordinated by the cosine function $cos(\pi \cdot t/(2 \cdot t_{max}))$, which gradually decreased as the number of iterations increased. There is a larger disturbance in the early iteration, and the disturbance is smaller in the later iteration, so as to balance the exploration and exploitation of the algorithm.

In addition, the current best individual $\boldsymbol{X}^*(t)$ updates its location relying on the population information which has poor exploration ability. Therefore, the position update of $\boldsymbol{X}^*(t)$ is performed using Gaussian walk in each generation in order to improve the global search ability of the algorithm. The model is as follows:

$$\boldsymbol{X}^*(t+1) = Gaussian(\boldsymbol{X}^*(t), \sigma_2) \tag{17}$$

$$\sigma_2 = \cos(\pi \cdot t/(2 \cdot t_M)) \cdot (\boldsymbol{X}^*(t) - \boldsymbol{X}_r^*(t)) \tag{18}$$

Finally, the greedy strategy is used to guarantee the global convergence efficiency in our proposed GDS-WOA. The greedy strategy is carried out to select the offspring from the parents and the generated candidates according to the fitness. This mechanism can fully retain the domination individuals which can improve the convergence speed of the algorithm. The pseudo code of the proposed GDS-WOA is described as algorithm 2 and the flowchart for GDS-WOA is as Fig. 1.

III. NUMERICAL EXPERIMENT BASED ON CEC2014 TEST SUITE

With the rapid development of intelligent optimization algorithms, it is easy for the competitive algorithm to obtain the global optimum of the classic benchmarks. CEC 2014 test suite is more challenging than classic benchmarks and is widely used to evaluate novel algorithms. Therefore, CEC 2014 test suite is employed to evaluate the performance of GDS-WOA to verify its performance. The CEC 2014 test suite consists of 30 benchmarks that can be classified into four categories: F1 to F3 are unimodal functions which are usually used to estimate the convergence speed of algorithms; F4 to F16 are multimodal functions which are usually used to evaluate the local optimum evasive ability of algorithms; F17 to F22 are hybrid functions and F23 to F30 are complex composition functions, it's very difficult for most algorithms to reach the global optimum. More details about these 30 benchmarks can be found in [34].

According to the using principle of CEC2014 test suite in [34], the max evaluation number (FE_{max}) of each benchmark is set to $D \times 10000$. D is the dimensionality of the test problem. In this work, D is equal to 30, and the search range of each dimension for all functions is $[-100, \ 100]$. Each test function is independently solved 51 times to reduce the randomness. Due to the global optimal solution of different functions are different, the results obtained by the algorithm are recorded using $f(\boldsymbol{X}_{Best}) - f(\boldsymbol{X}^*)$ for convenience. It is noted that $\boldsymbol{X}_{Best}$ is the best solution obtained by the algorithm in an experiment and $\boldsymbol{X}^*$ is the global optimal solution of the test function.

Additionally, we have compared GDS-WOA with different competitive algorithms consisting of two groups to demonstrate its efficiency. In one group, WOA and its variant IWOA [13] are selected to make comparisons.

In the other group, five state-of-the-art evolutionary algorithms are utilized as competitor, i.e., COA [35], VCS [36], CoBiDE [37], HFPSO [38] and GWO [39]. To make a fair comparison, the eight algorithms are performed in the experiment under the condition of same number of search individuals (SN) and FE_{max}, set to 500 and 300000 respectively. The other input parameters of compared algorithms are set as the original researches, as shown in Table 1. All the algorithms are implemented in MATLAB R2013a and the test environment is set up on a computer with Intel(R) Core(TM)i7-4770K CPU@3.50GHz 8GB RAM, running on Windows 7.

Algorithm 2 The Procedure of GDS-WOA

Step 1: Initialize initial whales population X_i; Set t = 1;
  $\boldsymbol{Pv} = (0.5, 0.5, \ldots, 0.5)$;
Step 2: Calculate the fitness of each search agent;
Step 3: If $t <= t_{max}$
    execute step 4;
  Else
    output $\boldsymbol{X}^*$;
  End the algorithm;
Step 4: Update the best solution $\boldsymbol{X}^*$ and its fitness obtained so far;
Step 5: Calculate $\boldsymbol{X}(t)_{mean}$ and the covariance matrix $\boldsymbol{Cov}(t)$ by (7) and (9);
Step 6: If the search stagnates
    Population is updated using Gaussian random walk by (15);
  Else
    $\boldsymbol{X}^*$ is updated using Gaussian walk by (17);
    For each solution $\boldsymbol{X}_i$ except $\boldsymbol{X}^*$;
      If $rand >= pv_i$
        $\boldsymbol{X}_i$ is updated using (10);
      Else
        Update a, l, p, A and C;
        If $p < 0.5$
          If $|A| < 1$
            $\boldsymbol{X}_i$ is updated using (6.1);
          Else
            $\boldsymbol{X}_i$ is updated using (6.2);
          End if
        Else
          $\boldsymbol{X}_i$ is updated using (6.3);
        End if
      End if
    End for
  End if
Step 7: Boundary control; calculate fitness of each agent; update $\boldsymbol{Pv}$;
Step 8: Greedy strategy is adopted to select the offspring;
Step 9: t = t + 1; execute step 3.

TABLE 1. Parameters of eight algorithms.

Algorithm	Parameter settings
IWOA	b=1, w varying step 0.1 from 0 to 1
WOA	b=1;
VCS	λ=0.5, σ=0.3;
COA	$Cpop=SN/2$, $Mpop=SN/2$;
CoBiDE	pb=0.4, ps=0.5;
HFPSO	c_1=c_2=1.49445, a=0.2, B_0=2,γ=1, w_i=0.9, w_f=0.5;
GWO	$a=2\text{-}2\cdot t/t_{max}$
GDS-WOA	$m=SN/2$; $b=1$;

A. RESULTS AND DISCUSSION

The simulation results obtained by GDS-WOA and other compared algorithms containing the mean and standard deviation (SD) are provided in Table 3. The best solutions among the eight algorithms are showed at bold.

According to Table 3, GDS-WOA can obtain the best solutions of unimodal functions F1 to F3 compared to the seven comparison algorithms. In addition, GDS-WOA converges to the global optimal solutions of F1 to F3, which verifies the efficiency of GDS-WOA in solving ill-conditioned functions; For multimodal functions F4 to F16, GDS-WOA has a greater convergence accuracy than the other seven algorithms with the best scores on F4, F6, F7, F8, F9, and F13, and it can converge to the global optimal solution on F7; HFPSO outperforms among the eight algorithms on F5, F10, F11, F12, and F15; COA performs slightly surpasses the proposed GDS-WOA on F14; And GWO is the best only in F16 in CEC 2014 benchmarks. In addition, our proposed GDS-WOA ranks top on all hybrid functions F17 to F22. Finally, according to the results of composition functions F23 to F30, IWOA and VCS are the two best outperforming algorithms on F23 and F24; IWOA, COA and VCS obtain the same better solutions for F25; IWOA ranks to top on F27 and F28; our proposed GDS-WOA outperforms among all algorithms on F26, F29, and F30. What is more, WOA has poor convergence accuracy on all benchmarks of CEC 2014. We concluded that our proposed GDS-WOA achieved best convergence accuracy on 18 out of 30 benchmarks in the CEC 2014 test suit, which demonstrates the accuracy and efficiency of our proposed GDS-WOA in solving different types of problems.

In order to analyze the overall difference of the algorithms, the nonparametric Friedman test is used based on the mean values derived from the algorithms on the CEC2014 test functions. The results of mean ranks obtained by the Friedman test are shown in Table 2. A smaller mean rank value represents greater performance of the algorithm. From Table 2, we conclude that our proposed GDS-WOA ranks top, and the other algorithms are in the following order: COA, HFPSO, VCS, CoBiDE, GWO, IWOA, and WOA. Moreover, the *p-value* is 1.3287E-06 that is less than the significance level α = 0.05 and the *chi-square* with 5 degrees of freedom (DOFs) is 96.93, which mean that there are significant differences among eight algorithms. To further analyze the magnitude of significant differences, the Iman-Davenport test [40] with a post hoc test is adopted. Iman-Davenport test is a

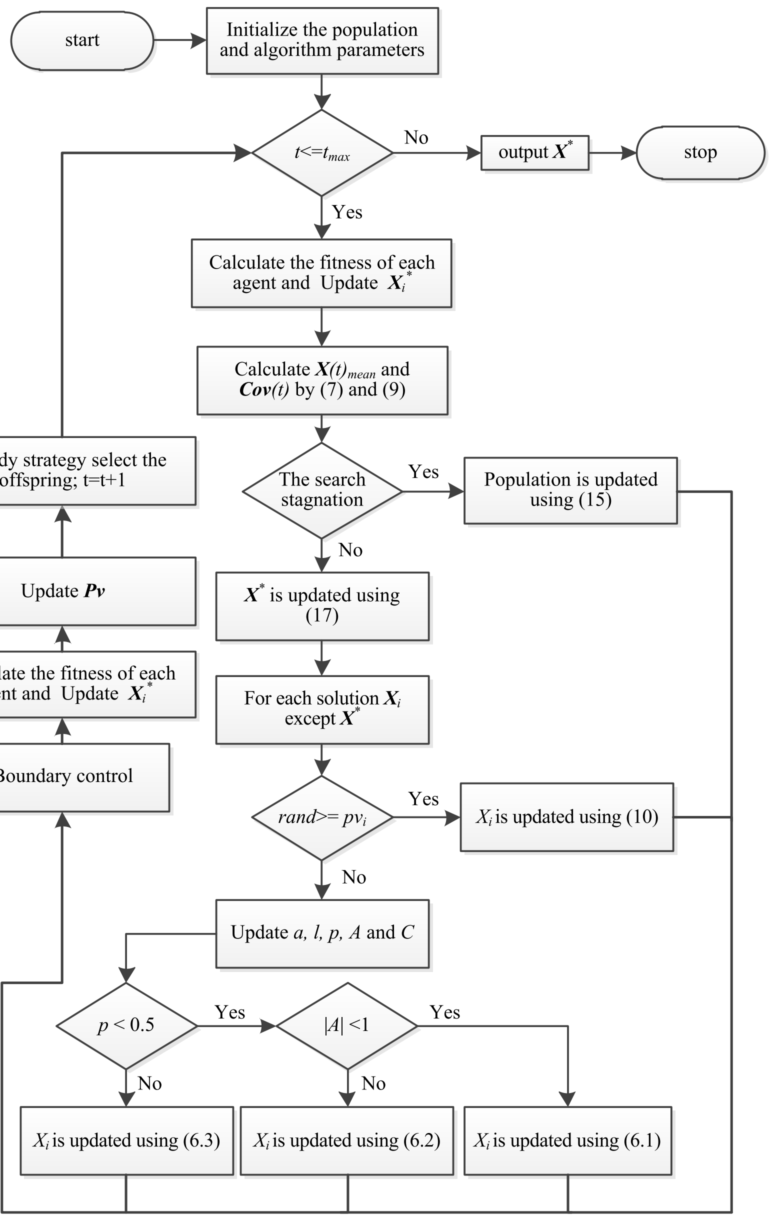

FIGURE 1. The flowchart of GDS-WOA.

TABLE 2. Mean ranks derived from the Friedman test with $\alpha = 0.05$.

Algorithm	IWOA	WOA	COA	VCS	CoBiDE	HFPSO	GWO	GDS-WOA
Mean ranks	6.6333	6.6333	3.0667	4.0333	4.6667	3.7667	5.3000	1.9000

Friedman test: *p-value* is 4.6372E-18; *Chi-square* is 96.93; Iman-Davenport test: *p-value* is 1.2425E-10;
F-distribution with 7 and 203 degrees of freedom is 24.86.

statistics distributed based on the F-distribution with (k-1) and $(k-1)(N-1)$ DOFs.

$$F_F^2 = \frac{(N-1) \cdot \chi_F^2}{N \cdot (K-1) - \chi_F^2} \tag{19}$$

In(19), K is the algorithm number and N equals to 30 being the CEC2014 benchmarks number. Therefore, the DOFs of the F-distribution in Iman-Davenport test are 7 and 203. In the paper, the Nemenyi test [41] is used as the post hoc test. The critical difference value (*CDV*) is used to assess the difference among eight algorithms based on the mean ranks obtained by the Friedman test. The *CDV* is calculated as the following:

$$CDV = q_a \cdot \sqrt{\frac{K(K+1)}{6N}} \tag{20}$$

In the paper, the critical value q_a is 2.3463 obtained from the statistical table of the F-distribution. Thus, the *CDV*

TABLE 3. Statistical results obtained by eight algorithms for CEC 2014 test set in 30D.

Mean(SD)	IWOA	WOA	COA	VCS	CoBiDE	HFPSO	GWO	GDS-WOA
F1	3.64E+08(4.62E+07)	3.55E+07(1.57E+07)	5.67E+05(3.03E+05)	5.13E+06(4.13E+06)	1.12E+06(2.33E+05)	8.48E+05(6.23E+05)	2.33E+07(1.45E+07)	**0.00E+00**(0.00E+00)
F2	5.40E+10(6.39E+09)	6.63E+06(5.15E+06)	2.17E+02(1.66E+02)	4.69E+05(7.87E+05)	5.61E+06(8.69E+05)	4.39E+03(1.01E+04)	4.57E+08(4.82E+08)	**0.00E+00(**0.00E+00)
F3	6.81E+04(4.68E+03)	2.73E+04(1.63E+04)	3.78E+01(4.14E+01)	2.22E+03(1.44E+03)	3.50E+01(5.30E+00)	1.66E+02(2.48E+02)	1.46E+04(5.44E+03)	**0.00E+00**(0.00E+00)
F4	5.24E+03(9.99E+02)	2.03E+02(6.71E+01)	7.10E+01(3.41E+01)	9.63E+01(3.41E+01)	1.26E+02(9.15E+00)	6.70E+01(3.94E+01)	1.52E+02(2.84E+01)	**0.00E+00**(0.00E+00)
F5	2.09E+01(7.34E-02)	2.03E+01(1.46E-01)	2.09E+01(6.56E-02)	2.07E+01(4.21E-02)	2.07E+01(4.17E-02)	**2.01E+01**(1.08E-01)	2.10E+01**(4.15E-02)**	2.07E+01(1.54E-01)
F6	3.41E+01(2.01E+00)	3.48E+01(3.00E+00)	1.37E+01(3.23E+00)	9.99E+00(2.99E+00)	2.61E+01(9.41E-01)	9.23E+00(4.02E+00)	1.03E+01(2.81E+00)	**1.75E-02**(1.25E-01)
F7	3.13E+02(5.26E+01)	1.09E+00(6.59E-02)	7.87E-03(8.68E-03)	6.61E-01(3.75E-01)	1.01E+00(1.80E-02)	1.14E-02(1.16E-02)	5.58E+00(4.12E+00)	**0.00E+00**(0.00E+00)
F8	2.50E+02(2.57E+01)	1.72E+02(3.47E+01)	4.04E+01(1.11E+01)	6.24E+01(7.32E+00)	8.21E+01(5.96E+00)	5.39E+01(1.32E+01)	5.29E+01(1.14E+01)	**3.95E+01**(1.32E+01)
F9	2.81E+02(2.33E+01)	2.25E+02(6.06E+01)	8.36E+01(1.99E+01)	1.14E+02(4.06E+01)	1.67E+02**(1.10E+01)**	7.04E+01(1.96E+01)	7.23E+01(2.46E+01)	**3.57E+01**(1.32E+01)
F10	5.38E+03(6.13E+02)	3.92E+03(6.56E+02)	1.63E+03(9.37E+02)	2.21E+03(6.01E+02)	2.78E+03**(2.24E+02)**	**1.13E+03**(3.23E+02)	1.83E+03(6.45E+02)	1.21E+03(4.83E+02)
F11	6.45E+03(6.91E+02)	4.59E+03(7.78E+02)	3.59E+03(8.54E+02)	4.71E+03(7.07E+02)	5.50E+03**(2.17E+02)**	**2.63E+03**(5.53E+02)	2.77E+03(1.03E+03)	2.89E+03(6.98E+02)
F12	2.28E+00(5.18E-01)	1.61E+00(5.17E-01)	1.56E+00(2.08E-01)	1.27E+00(2.45E-01)	1.31E+00**(1.70E-01)**	**2.63E-01**(1.88E-01)	2.02E+00(1.01E+00)	5.91E-01(3.97E-01)
F13	6.30E+00(5.41E-01)	5.27E-01(1.28E-01)	3.07E-01(5.48E-02)	3.94E-01(5.23E-02)	4.43E-01**(4.30E-02)**	3.46E-01(7.91E-02)	3.24E-01(7.49E-02)	**2.76E-01**(5.83E-02)
F14	8.30E+01(4.94E+00)	2.83E-01(9.22E-02)	**2.37E-01**(3.41E-02)	3.08E-01(1.24E-01)	2.55E-01**(2.96E-02)**	4.96E-01(3.21E-01)	6.98E-01(1.03E+00)	2.53E-01(3.53E-02)
F15	6.01E+03(3.71E+03)	6.86E+01(2.50E+01)	6.54E+00(2.10E+00)	1.21E+01(1.42E+00)	1.90E+01(1.38E+00)	**4.12E+00**(1.37E+00)	1.13E+01(5.19E+00)	5.31E+00(2.15E+00)
F16	1.26E+01(4.07E-01)	1.26E+01(5.40E-01)	1.21E+01(2.47E-01)	1.17E+01(2.95E-01)	1.22E+01**(1.95E-01)**	1.0687E+01(6.83E-01)	**1.0659E+01**(6.72E-01)	1.08E+01(7.87E-01)
F17	5.86E+07(1.96E+07)	3.21E+06(2.24E+06)	1.91E+04(2.19E+04)	3.88E+05(2.96E+05)	2.21E+03**(2.17E+02)**	1.57E+05(1.05E+05)	7.80E+05(6.10E+05)	**7.06E+02**(4.21E+02)
F18	1.87E+09(1.05E+09)	1.30E+04(3.54E+04)	1.65E+03(2.13E+03)	1.18E+04(1.00E+04)	1.26E+02**(1.18E+01)**	4.64E+03(6.59E+03)	2.82E+06(7.38E+06)	**3.46E+01**(1.94E+01)
F19	2.25E+02(4.94E+01)	5.85E+01(3.99E+01)	1.02E+01(1.12E+01)	9.37E+00(8.47E+00)	9.50E+00**(3.83E-01)**	8.89E+00(1.75E+00)	1.61E+01(6.26E+00)	**2.42E+00**(7.88E-01)
F20	6.86E+04(2.36E+04)	2.07E+04(1.83E+04)	1.80E+02(5.91E+01)	3.98E+03(2.31E+03)	5.97E+01**(4.61E+00)**	7.83E+02(4.94E+02)	7.66E+03(4.72E+03)	**2.96E+01**(7.55E+00)
F21	2.13E+07(1.06E+07)	9.83E+05(8.45E+05)	6.79E+03(8.41E+03)	1.39E+05(1.37E+05)	1.04E+03**(1.24E+02)**	7.49E+04(6.98E+04)	3.48E+05(3.70E+05)	**6.00E+02**(2.40E+02)
F22	1.68E+03(4.19E+02)	7.33E+02(2.21E+02)	2.13E+02(7.58E+01)	2.38E+02(1.03E+02)	2.47E+02**(6.47E+01)**	3.17E+02(1.24E+02)	2.84E+02(1.51E+02)	**1.96E+02**(1.02E+02)
F23	**2.0000E+02(0.00E+00)**	3.33E+02(7.56E+00)	2.0023E+02(4.62E-01)	**2.0000E+02(0.00E+00)**	3.15E+02(1.53E-02)	3.15E+02(2.97E-13)	3.24E+02(2.81E+00)	3.15E+02(4.02E-13)
F24	**2.0000E+02(0.00E+00)**	2.07E+02(8.53E+00)	2.0000E+02(1.46E-05)	**2.0000E+02(0.00E+00)**	2.31E+02(9.71E-01)	2.22E+02(7.24E+00)	2.00E+02(1.24E-03)	2.08E+02(1.06E+01)
F25	**2.0000E+02(0.00E+00)**	2.27E+02(1.30E+01)	**2.0000E+02**(0.00E+00)	**2.0000E+02(0.00E+00)**	2.06E+02(4.81E-01)	2.07E+02(3.73E+00)	2.09E+02(2.49E+00)	2.03E+02(3.59E-02)
F26	1.05E+02(1.85E-01)	1.01E+02(1.23E-01)	1.0031E+02(6.46E-02)	1.0036E+02(6.65E-02)	1.0044E+02(4.87E-02)	1.02E+02(1.40E+01)	1.0034E+02(8.90E-02)	**1.0026E+02**(5.51E-02)
F27	**2.0000E+02(0.00E+00)**	8.57E+02(4.26E+02)	2.06E+02(7.97E+00)	2.0000E+02(1.79E-11)	4.06E+02(1.21E+00)	4.68E+02(1.17E+02)	4.96E+02(9.15E+01)	3.61E+02(4.93E+01)
F28	**2.0000E+02(0.00E+00)**	2.07E+03(5.71E+02)	2.27E+02(6.77E+01)	2.0000E+02(4.72E-12)	1.03E+03(1.81E+01)	1.17E+03(3.15E+02)	9.18E+02(8.17E+01)	7.83E+02(1.41E+02)
F29	2.00E+02**(0.00E+00)**	3.64E+06(4.59E+06)	1.26E+03(2.90E+02)	4.62E+03(2.16E+03)	1.38E+03(9.78E+01)	1.71E+05(1.21E+06)	5.24E+04(8.01E+04)	**1.23E+02**(2.51E+01)
F30	8.29E+04(3.28E+05)	6.41E+04(4.93E+04)	2.07E+03(7.08E+02)	6.96E+03(2.45E+03)	2.15E+03**(2.15E+02)**	2.32E+03(1.01E+03)	2.23E+04(1.33E+04)	**5.15E+02**(1.49E+02)

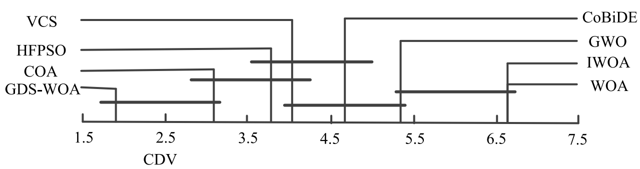

FIGURE 2. Comparison of multiple algorithms.

is 1.4839 with a significance level $\alpha = 0.05$. The differences among eight algorithms are shown in Fig.2. The algorithms with similar performance can be connected using the *CDV*. From Fig.2, GDS-WOA and COA have similar performance, and outperform the other six algorithms. The proposed GDS-WOA in this paper exhibits a superior performance on the CEC 2014 test suite compared to various types of state-of-the-art algorithms.

For much further comprehensive comparisons in a statistical manner, the Wilcoxon signed rank test, a method for testing pair data, is adopted to analyze the results obtained by the eight algorithms solving the CEC2014 test set in 30D with 51 independent runs. It can be used to test whether the performance of algorithms has obvious differences. Under the condition of significance level $\alpha = 0.05$, the results of Wilcoxon signed rank test are listed in Table 4. The meaning of symbols in Table 4 is as follows: *p-value* is the probability of observing the given results, the hypothesis is rejected at the 5% when *p-value* is no more than α, meaning that there is obvious difference between the two algorithms; 'w+' is the sum of the rank that is greater than 0 and 'w−' represents the sum of the rank that is less than 0; R indicates the results of Wilcoxon signed rank test, in which '+' represents the performance of the competitor is better than GDS-WOA, whereas '−' indicates the competitor is inferior to GDS-WOA; And the symbol '=' indicates that the competitor is similar to GDS-WOA, there is no significant difference. As we can see from the last row in Table 4, the performance of GDS-WOA is superior to the other 7 comparison algorithms in at least 22 functions. Therefore, the performance of our proposed GDS-WOA is significantly better than the other seven comparison algorithms.

To further illustrate the performance of GDS-WOA, the convergence curves of eight algorithms are shown as Fig.3. The convergence speed is compared according to the slope of convergence curve. In order to save the article layout, this paper only lists the convergence curves of eight algorithms on twelve representative functions, i.e., F1, F2, F4, F6, F7, F17, F18, F19, F20, F21, F29, F30, in which GDS-WOA has obtained the better results. These 12 representative functions

TABLE 4. Results of the Wilcoxon signed ranks test based on the solutions with 51 independent runs.

No.	IWOA vs. GDS-WOA				WOA vs. GDS-WOA				COA vs. GDS-WOA				VCS vs. GDS-WOA			
	p-value	w+	w-	R	p-value	w+	w-	R	p-value	w+	w-	R	p-value	w+	w-	R
F1	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F2	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F3	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F4	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F5	8.27E-10	8	1318	-	8.27E-10	1318	8	+	1.77E-09	21	1305	-	1.93E-01	524	802	=
F6	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F7	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F8	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.36E-01	597	729	=	1.99E-09	23	1303	-
F9	5.15E-10	0	1326	-	5.15E-10	0	1326	-	6.93E-10	5	1321	-	5.46E-10	1	1325	-
F10	5.15E-10	0	1326	-	5.15E-10	0	1326	-	1.68E-02	408	918	-	1.27E-08	56	1270	-
F11	5.15E-10	0	1326	-	1.18E-09	14	1312	-	7.94E-05	242	1084	-	1.05E-09	12	1314	-
F12	5.15E-10	0	1326	-	7.74E-09	47	1279	-	6.53E-10	4	1322	-	6.92E-09	45	1281	-
F13	5.15E-10	0	1326	-	6.15E-10	3	1323	-	1.52E-02	404	922	-	9.87E-10	11	1315	-
F14	5.15E-10	0	1326	-	6.35E-02	465	861	=	1.68E-02	918	408	+	4.89E-04	291	1035	-
F15	5.15E-10	0	1326	-	5.15E-10	0	1326	-	3.15E-03	348	978	-	5.15E-10	0	1326	-
F16	5.46E-10	1	1325	-	5.15E-10	0	1326	-	2.10E-09	24	1302	-	1.87E-07	107	1219	-
F17	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F18	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.46E-10	1	1325	-	5.15E-10	0	1326	-
F19	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F20	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F21	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F22	5.15E-10	0	1326	-	6.15E-10	3	1323	-	2.90E-01	550	776	=	3.18E-02	434	892	-
F23	9.24E-13	1326	0	+	5.15E-10	0	1326	-	5.15E-10	1326	0	+	9.24E-13	1326	0	+
F24	5.15E-10	1326	0	+	8.66E-01	681	645	=	3.39E-01	765	561	=	5.15E-10	1326	0	+
F25	5.15E-10	1326	0	+	1.77E-09	21	1305	-	5.15E-10	1326	0	+	5.15E-10	1326	0	+
F26	5.15E-10	0	1326	-	6.93E-10	5	1321	-	3.19E-04	279	1047	-	1.87E-07	107	1219	-
F27	2.04E-10	1326	0	+	5.15E-10	0	1326	-	5.15E-10	1326	0	+	2.25E-10	1326	0	+
F28	5.15E-10	1326	0	+	5.15E-10	0	1326	-	5.15E-10	1326	0	+	5.15E-10	1326	0	+
F29	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F30	1.31E-05	1128	198	+	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.46E-10	1	1325	-
+/-/=		6/24/0				1/27/2				5/22/3				5/24/1		

No.	CoBiDE vs. GDS-WOA				HFPSO vs. GDS-WOA				GWO vs. GDS-WOA			
	p-value	w+	w-	R	p-value	w+	w-	win	p-value	w+	w-	R
F1	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F2	5.15E-10	0	1326	-	5.13E-10	0	1326	-	5.15E-10	0	1326	-
F3	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F4	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F5	1.27E-01	500	826	=	5.15E-10	1326	0	+	8.27E-10	8	1318	-
F6	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F7	5.15E-10	0	1326	-	5.03E-10	0	1326	-	5.15E-10	0	1326	-
F8	5.15E-10	0	1326	-	4.37E-06	173	1153	-	1.20E-06	145	1181	-
F9	5.15E-10	0	1326	-	2.80E-09	29	1297	-	1.40E-09	17	1309	-
F10	5.15E-10	0	1326	-	3.63E-01	760	566	=	4.78E-06	175	1151	-
F11	5.15E-10	0	1326	-	6.21E-02	862	464	=	1.07E-01	835	491	=
F12	2.80E-09	29	1297	-	6.24E-06	1145	181	+	3.37E-08	74	1252	-
F13	6.93E-10	5	1321	-	4.37E-05	227	1099	-	2.97E-04	277	1049	-
F14	9.03E-01	650	676	=	2.66E-04	274	1052	-	1.07E-07	96	1230	-
F15	5.15E-10	0	1326	-	3.56E-03	974	352	+	1.02E-08	52	1274	-
F16	9.87E-10	11	1315	-	2.34E-01	790	536	=	2.61E-01	783	543	=
F17	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F18	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F19	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F20	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F21	8.77E-10	9	1317	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F22	4.64E-03	361	965	-	1.62E-05	203	1123	-	1.80E-03	330	996	-
F23	5.15E-10	0	1326	-	1.70E-10	0	1225	-	5.15E-10	0	1326	-
F24	5.15E-10	0	1326	-	1.20E-08	55	1271	-	3.39E-01	765	561	=
F25	5.15E-10	0	1326	-	5.46E-10	1	1325	-	5.15E-10	0	1326	-
F26	5.15E-10	0	1326	-	1.45E-07	102	1224	-	1.37E-05	199	1127	-
F27	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F28	5.15E-10	0	1326	-	6.15E-10	3	1323	-	6.03E-08	85	1241	-
F29	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
F30	5.15E-10	0	1326	-	5.15E-10	0	1326	-	5.15E-10	0	1326	-
+/-/=		0/28/2				3/24/3				0/27/3		

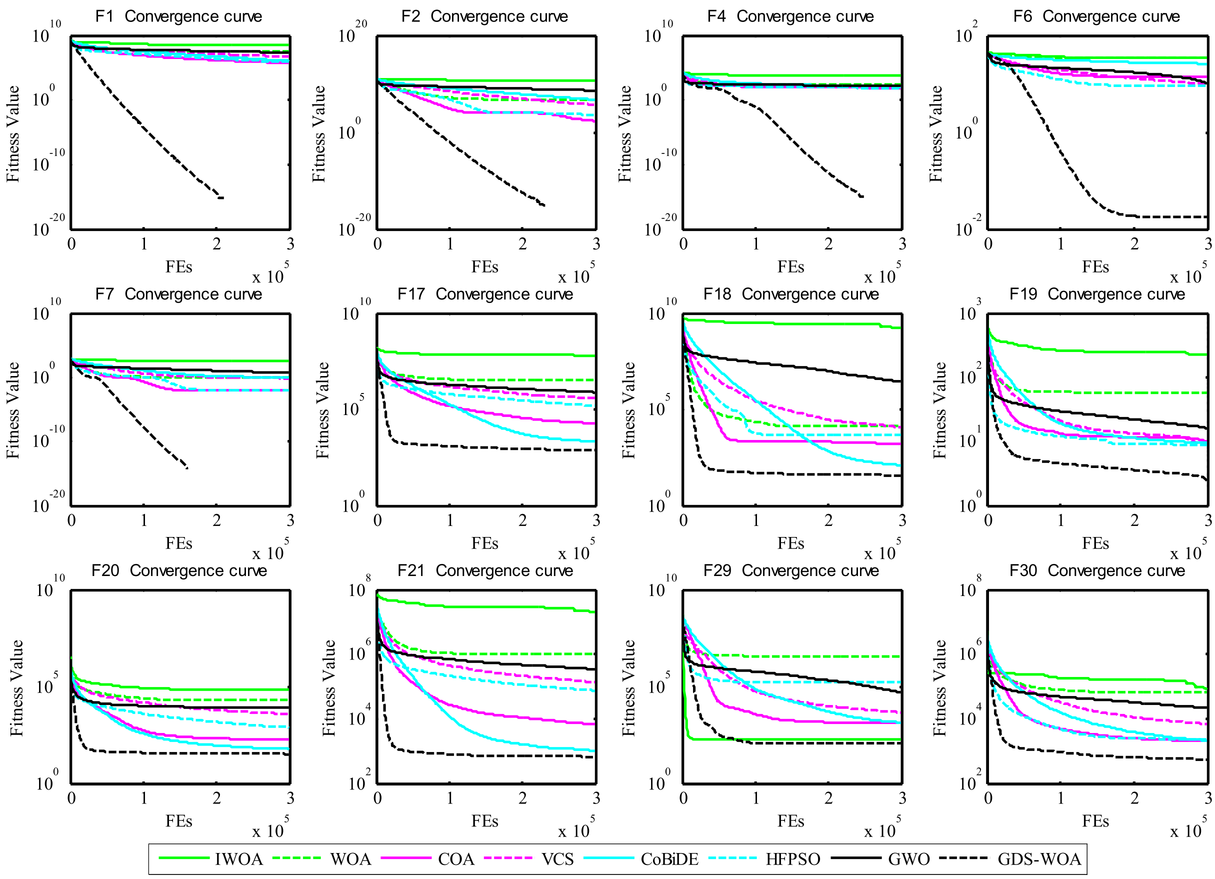

FIGURE 3. Convergence graphs of 8 algorithms on 12 representative functions.

consist of two unimodal functions, three multimodal functions, five hybrid functions, and two composition functions. The convergence curves are drawn based on the mean values of 51 independent runs. From Fig.3, our proposed GDS-WOA has a faster convergence speed and better convergence accuracy than the other seven comparison algorithms on 11 out of 12 functions except F29. For F29, the convergence rate of GDS-WOA is only inferior to IWOA. In addition, it is obvious that the search of IWOA and WOA is stagnant on 11 out of 12 functions except F29 and the local optimum evasive ability of other comparison algorithms is also worse than GDS-WOA. What is more, GDS-WOA has strong local optimum avoidance ability and more high convergence accuracy than the seven comparison algorithms. Taking the results in Table 3 into account, the convergence accuracy of GDS-WOA on 18 functions among 30 functions is better than the seven comparison algorithms. In summary, GDS-WOA is superior to the other seven comparison algorithms in convergence accuracy and efficiency.

Box diagrams of the results, obtained by eight algorithms with 51 independent runs to solve twelve representative functions, are shown as Fig.4. The box graph includes a central median value, outliers, the 1/4 and 3/4 values in 51 results. According to Fig.4, when GDS-WOA solves F1, F2, F4, F7, F17, F18, F19, F29 and F30, there are no abnormal values in 51 results. Although there are some abnormal values obtained by GDS-WOA when solving F6, F20, F21, the whole distribution of GDS-WOA is still more concentrated compared with the comparison algorithms. In summary, our proposed GDS-WOA has greater robustness and stability and generally has low standard deviation values.

B. ALGORITHM COMPUTATION COST ANALYSIS

Algorithm computational efficiency is an important issue when evaluating the performance of a novel algorithm, and it can be characterized by computational cost. The means of computational time for eight algorithms in CEC 2014 test suite with 30D are shown in Table 5. To make a more intuitive expression of the computational efficiency of algorithms, a radar graph based on the ranks of the average computational time is presented in Fig.5. The smaller the circle is, the more efficient the algorithm. Fig.5 shows that the computational cost of GDS-WOA in this paper is less than COA, VCS, and HFPSO, but more than IWOA, WOA, GWO, and CoBiDE. However, the ranks based on convergence accuracy of IWOA, WOA, GWO and CoBiDE are inferior to the other four algorithms. In addition, GDS- WOA has a faster convergence speed according to the convergence graphs, which means that it can get a better solution in solving a problem with a limitation of FE_{max} to meet real-time requirements. From what

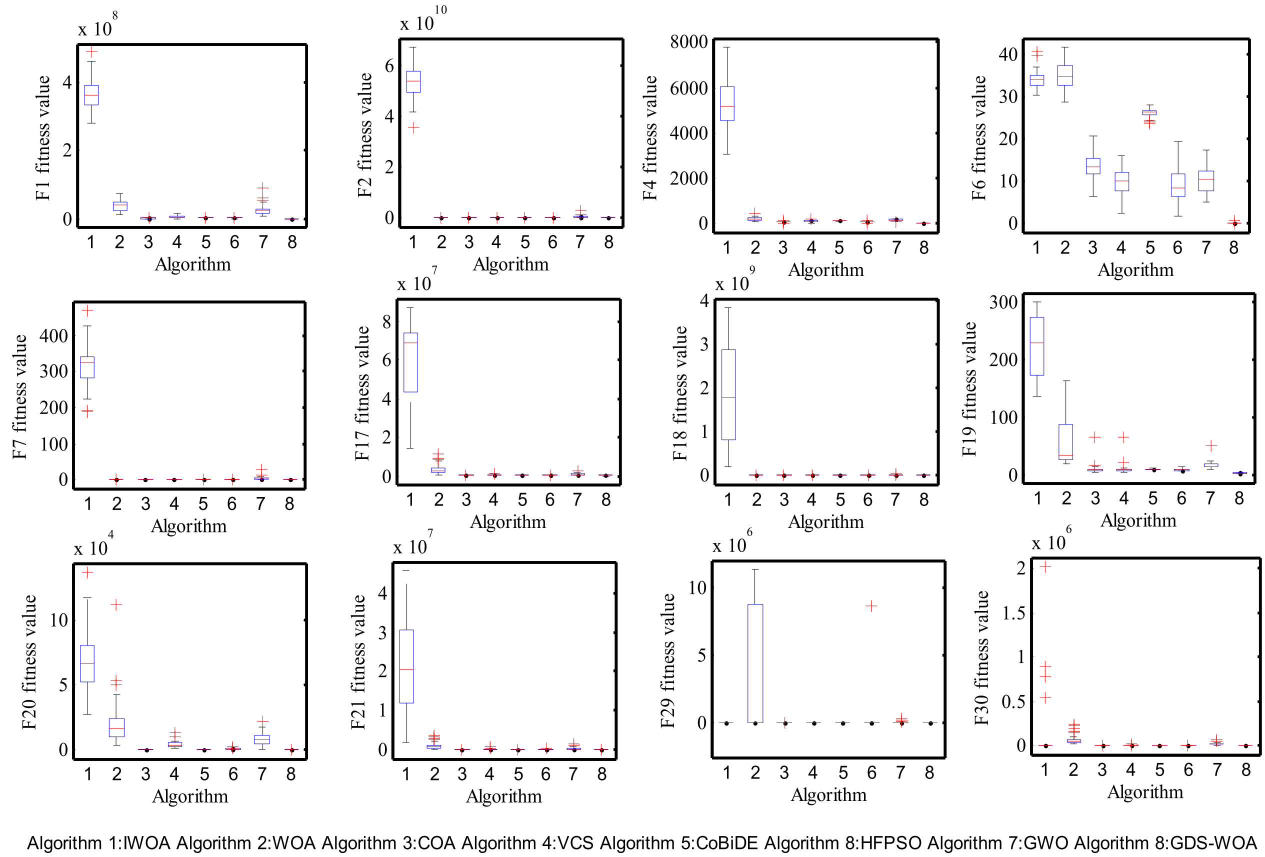

FIGURE 4. Box diagrams of solutions obtained by eight algorithms on 12 benchmark functions with 51 independent runs.

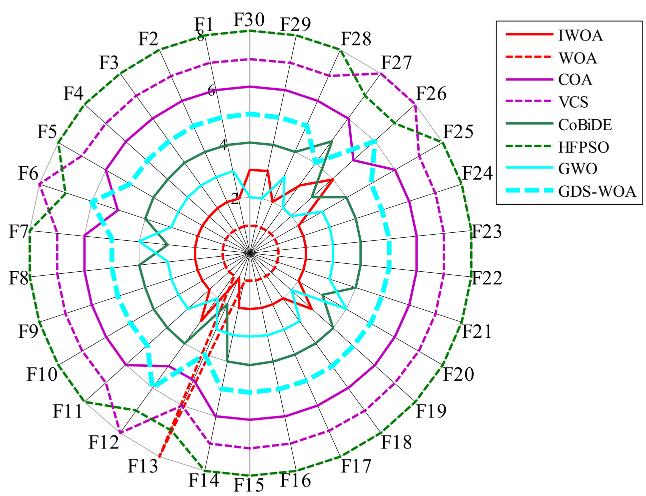

FIGURE 5. Ranks of mean computational time of algorithms.

has been analyzed above, we can conclude that GDS-WOA is superior to the comparison algorithms in both convergence accuracy and efficiency on the CEC 2014 test suite with 30D, although the computational cost of GDS-WOA is high.

IV. UCAVs TASK ALLOCATION USING GDS-WOA

The ultimate purpose of the proposed GDS-WOA is to solve optimization problems of engineering. In this section, we employ GDS-WOA to solve the heterogeneous multi-unmanned combat aircraft vehicles (UCAVs) task allocation problem. UCAV is an airborne unmanned combat system that has been developed by various military powers in the world and which can carry out various combat tasks such as air defense suppression, ground strike, air combat and intelligence reconnaissance. The UCAVs formation cooperative operation task allocation is the key technology of cooperative operation.

Recently, scholars have proposed a variety of task allocation models which are widely used, including mixed integer linear programming model (MILP) [42], multi-trip salesman problem model (MTSP) [43], vehicle routing problem model (VRP) [44] and Network Flow optimization model [45]. Based on these general models, scholars have established specific models for different operational backgrounds. Grøtli and Johansen [46] used the mixed integer linear programming model to deal with the task allocation problem of UAV, which could guarantee the global optimal allocation results, but it has poor real-time and weak applicauility in large-scale task allocation problem. Wu *et al.* [47] developed a distributed heterogeneous UAVs task allocation model based on the constraints and uncertainties in combat tasks, and proposed an algorithm based on the consistency algorithm and online collaboration strategy to solve the model. The method can obtain the feasible solution within

TABLE 5. Running time (second) of eight algorithms in CEC 2014 test with 30D.

Rank	IWOA	WOA	COA	VCS	CoBiDE	HFPSO	GWO	GDS-WOA
F1	4.86E+00	4.72E+00	9.74E+00	1.27E+01	5.65E+00	1.49E+01	5.52E+00	5.68E+00
F2	4.81E+00	4.66E+00	9.68E+00	1.25E+01	5.65E+00	1.47E+01	5.47E+00	6.11E+00
F3	4.48E+00	4.32E+00	9.29E+00	1.23E+01	5.36E+00	1.45E+01	5.13E+00	5.74E+00
F4	4.49E+00	4.34E+00	9.35E+00	1.23E+01	5.15E+00	1.46E+01	5.15E+00	5.75E+00
F5	4.76E+00	4.62E+00	9.93E+00	1.33E+01	5.87E+00	1.48E+01	5.43E+00	9.74E+00
F6	5.69E+01	5.65E+01	6.12E+01	6.71E+01	5.91E+01	6.63E+01	5.71E+01	6.18E+01
F7	4.85E+00	4.70E+00	9.72E+00	1.29E+01	5.49E+00	1.47E+01	5.50E+00	7.14E+00
F8	4.12E+00	3.97E+00	8.89E+00	1.20E+01	4.83E+00	1.40E+01	4.78E+00	6.45E+00
F9	4.75E+00	4.60E+00	9.49E+00	1.17E+01	5.56E+00	1.47E+01	5.41E+00	7.83E+00
F10	5.02E+00	4.90E+00	9.94E+00	1.34E+01	5.97E+00	1.49E+01	5.67E+00	7.67E+00
F11	5.68E+00	5.56E+00	1.07E+01	1.47E+01	6.71E+00	1.56E+01	6.35E+00	8.86E+00
F12	3.70E+01	3.59E+01	4.11E+01	4.61E+01	3.82E+01	4.61E+01	3.67E+01	4.18E+01
F13	4.72E+00	1.48E+02	9.31E+00	1.31E+01	5.11E+00	1.44E+01	5.16E+00	8.53E+00
F14	4.76E+00	4.42E+00	9.35E+00	1.30E+01	5.33E+00	1.44E+01	5.18E+00	8.87E+00
F15	5.08E+00	4.77E+00	9.56E+00	1.18E+01	5.54E+00	1.47E+01	5.50E+00	8.78E+00
F16	5.15E+00	4.83E+00	9.88E+00	1.31E+01	5.62E+00	1.48E+01	5.53E+00	8.82E+00
F17	5.40E+00	5.06E+00	9.94E+00	1.40E+01	5.87E+00	1.52E+01	5.78E+00	8.58E+00
F18	5.08E+00	4.71E+00	9.59E+00	1.36E+01	5.57E+00	1.47E+01	5.45E+00	8.27E+00
F19	1.61E+01	1.53E+01	1.99E+01	2.37E+01	1.64E+01	2.51E+01	1.58E+01	1.82E+01
F20	5.09E+00	4.76E+00	9.63E+00	1.37E+01	5.43E+00	1.48E+01	5.48E+00	8.15E+00
F21	5.41E+00	5.08E+00	9.88E+00	1.40E+01	5.87E+00	1.52E+01	5.78E+00	8.51E+00
F22	8.69E+00	8.14E+00	1.31E+01	1.75E+01	9.16E+00	1.81E+01	8.89E+00	1.10E+01
F23	1.10E+01	1.03E+01	1.52E+01	1.75E+01	1.16E+01	2.05E+01	1.11E+01	1.32E+01
F24	8.61E+00	8.17E+00	1.28E+01	1.50E+01	9.09E+00	1.81E+01	8.86E+00	1.10E+01
F25	9.80E+00	9.25E+00	1.39E+01	1.62E+01	1.04E+01	1.95E+01	9.96E+00	1.21E+01
F26	6.73E+01	6.41E+01	6.86E+01	7.53E+01	6.60E+01	7.39E+01	6.45E+01	6.96E+01
F27	6.69E+01	6.37E+01	6.85E+01	7.53E+01	6.82E+01	7.39E+01	6.44E+01	6.81E+01
F28	1.26E+01	1.23E+01	1.72E+01	2.06E+01	1.35E+01	2.34E+01	1.29E+01	1.52E+01
F29	2.02E+01	1.92E+01	2.40E+01	2.85E+01	2.10E+01	2.97E+01	1.99E+01	2.23E+01
F30	1.34E+01	1.28E+01	1.70E+01	2.17E+01	1.35E+01	2.27E+01	1.34E+01	1.56E+01

the acceptable time. Based on an analysis of key technical and tactical indicators of task allocation problems, Zhong *et al.* [48] established a task allocation model for UCAV under the constraints of manned combat aerial vehicle control, and a multi-group ant colony algorithm is adopted to solve the model. Kurdi *et al.* [49] proposed an automatic bio-inspired method for multi-UAV dynamic task allocation. The task allocation results are dynamically adjusted based on the operational status of each UAV and task parameters without direct communication between UAVs, which shows that the method has certain scalability in task allocation of different scales. In view of the low efficiency and unreasonable results of traditional contract network, Li *et al.* [50] proposed a multi-AUV (autonomous underwater vehicle) task allocation strategy based on improved contract network by introducing the token ring network and task load rate indicators. The novel method had improved the overall task allocation efficiency and obtained a reasonable allocation scheme. Huang and Zhuo [51] firstly established a multi-UCAV cooperative combat model with collaborative decision-making and control. Then, the task allocation model was established based on the flight characteristics of UCAV and the constraints of the battlefield environment. The model was solved by different algorithm according to the battlefield environment, which could satisfy the demand of the cooperative operation to a certain degree. Huang *et al.* [52] proposed a multi-type UAVs cooperative task allocation model based on cross-entropy, taking the types of UAVs and the resources constraint into account, and the author revealed that the proposed model has the advantage of solving large-scale task allocation problems. In the context of multi-UAVs performing search and rescue tasks, Miao *et al.* [53] established a dynamic task allocation model considering the conditions of detecting new targets, UAV destroyed and sudden threat source. And the model was solved by a distributed immune multi-agent algorithm. In the context of performing collaborative attack tasks, Nie-Qiang *et al.* [54] proposed a multi-UCAV cooperative task allocation algorithm based on immune evolutionary computation. However, the task allocation model used was too simple to meet actual operational needs. Xu *et al.* [55] established an extended mixed integer linear programming (MILP) task assignment model and used an improved co-evolutionary genetic algorithm to solve the model.

However, most of the researches about task allocation of UCAV discussed above do not take some factors into account, such as the UCAV platform characteristics, task types and other attribute differences. The UCAVs task allocation model is usually assumed that UCAVs and tasks have the same characteristics. Multi-UCAV collaboration operation based on the models is just a linear addition of multiple UCAV, and it is difficult to achieve the operational effectiveness values of $1 + 1 > 2$. In addition, the attributes of the targets are different which have different requirements of the UCAV. Thus, the formation of UCAV with different functions and characteristics is often required to carry out the corresponding combat tasks more efficiently. Therefore, to improve the operational effectiveness value of UCAV formation executing a variety of combat tasks, this paper proposes the heterogeneous multi-UCAV task allocation model, and the GDS-WOA is used to solve it.

A. PROPOSED UCAVs TASK ALLOCATION MODEL

To maximize the operational effectiveness value of UCAV formation, the heterogeneous UCAV formation consists of several UCAV with different kinds of task loads according to platform differences. The UCAV with different task loads focus on executing the different type combat tasks in the operation process, which could realize the overall operational effectiveness value of UCAV formation with $1 + 1 > 2$ under the constraints of limited loads in a single UCAV.

In this paper, we assume that the operational tasks need to be performed by heterogeneous UCAV formation only include three types: reconnaissance, attack, and evaluation. The formation of heterogeneous UCAVs is $U = \{U_1, U_2, U_3, U_{Nu}\}$, where Nu is the UCAV number in the formation. And the task set is $T = \{T_l, T_2, T_3, T_{Nt}\}$, where Nt is the task number. The goal of task allocation is to maximize the operational effectiveness value (OE) of the UCAV formation. Based on the task allocation model used in [48], [52], [54], [56], this paper proposes the following heterogeneous UCAV task allocation model considering the actual operational factors.

Definition 1: The operational effectiveness value of the i^{th} UCAV performing the j^{th} task is equal to the reward of performing the task divided by the cost of performing the task.

$$OE_i(T_j) = Reward_i(T_j)/Cost_i(T_j) \tag{21}$$

where $OE_i(T_j)$, $Reward_i(T_j)$, $Cost_i(T_j)$ are the operational effectiveness value, reward, and cost of the i^{th} UCAV performing the j^{th} task respectively.

1) UCAVs PERFORM TASK REWARD

Factors associated with the value of *Reward* include the value of the task, the complete capability of the UCAV performing the task, and the defensive ability of the task. The value of a task is usually determined by the chief operating officer based on the pre-acquired intelligence, and it could be dynamically adjusted in real-time according to the change of operational intent and tactics in the operational process. The defensive ability of a task is obtained by analyzing and quantifying the task information obtained by the pre-reconnaissance. The complete capability of the UCAV to perform a different type of task is quantified by the platform performance of the UCAV and the type of the task load carried, which is usually described in the form of probability according to operational statistics of related tasks in the past.

Definition 2: The *Reward* is proportional to the complete capability of the UCAV and the value of the task, and inversely proportional to the task defense capability.

$$Reward_i(T_j) = Value(T_j) \cdot P_i(T_j)/Defense(T_j) \tag{22}$$

where $Value\ (T_j)$ and $Defence\ (T_j)$ are the value and defensive ability of the j^{th} task; $P_i(T_j)$ is the complete capability of the i^{th} UCAV performing the j^{th} task.

The task reward of the entire UCAV formation is as follows:

$$Reward = \sum_{i=1}^{Nu} \sum_{j=1}^{Nt} x_{ij} \cdot Reward_i(T_j) \tag{23}$$

where $x_{ij} \in \{0, 1\}$ is the decision variable, $x_{ij} = 0$ indicates that the i^{th} UCAV does not perform the j^{th} task and $x_{ij} = 1$ represents that the i^{th} UCAV performs the j^{th} task.

2) UCAVs PERFORM TASK COST

UCAVs are at the risk of being attacked or destroyed by enemy defense systems when performing tasks in the operational area. To reduce the probability of UCAVs being destroyed, the time of UCAVs in the combat area should be reduced, that is, the track length should be minimized. At the same time, the attack ability of the task and the defensive ability of UCAV should be considered in task allocation. Therefore, the cost of UCAVs executing tasks in this model includes the track length cost and the loss cost. Since the size of track length cost and the loss cost are different, the value needs to be normalized.

The Euclidean distance between the i^{th} UCAV and the j^{th} task can be expressed as:

$$DisUT_{ij} = \sqrt{(pos_x(U_i) - pos_x(T_i))^2 + (pos_y(U_i) - pos_y(T_i))^2} \tag{24}$$

The Euclidean distance between the i^{th} task and the j^{th} task can be expressed as:

$$DisTT_{ij} = \sqrt{(pos_x(T_i) - pos_x(T_j))^2 + (pos_y(T_i) - pos_y(T_j))^2} \tag{25}$$

where $pos(U_i) = (pos_x(U_i)pos_y(U_i))$ is the coordinate of the i^{th} UCAV; $pos(T_i) = (pos_x(T_i)pos_y(T_i))$ is the coordinate of the i^{th} task; $pos(T_j) = (pos_x(T_j)pos_y(T_j))$ is the coordinate of the j^{th} task. In order to avoid repeated calculation in the optimization process, the distance matrix is constructed to store the Euclidean distance between the UCAV and the target, the task target and the task target.

$$\boldsymbol{Distance} = \begin{bmatrix} DisUT_{1,1} & \cdots & DisUT_{1,Nt} \\ \vdots & \ddots & \vdots \\ DisUT_{Nu,1} & \cdots & DisUT_{Nu,Nt} \\ DisTT_{1,1} & \cdots & DisTT_{1,Nt} \\ \vdots & \ddots & \vdots \\ DisTT_{Nt,1} & \cdots & DisTT_{Nt,Nt} \end{bmatrix} \tag{26}$$

Normalizing the distance matrix is as follow:

$$\boldsymbol{Distance} = \boldsymbol{Distance}/Distance_{\max} \tag{27}$$

where $Distance_{max}$ represents the maximum element in ***Distance***.

In summary, the track length of the i^{th} UCAV performing all assigned tasks and safely returning to the starting point can be expressed as:

$$Length_i = \sum_{n=0}^{N} Length(i)_{n,n+1} \tag{28}$$

where N is the task number that the i^{th} UCAV needs to perform; $Length(i)_{0,1}$ is the track length from the i^{th} UCAV position to the first executed task position; $Length(i)_{N,N+1}$ is the track length of the i^{th} UCAV returning to the base after performing the last task and $Length\ (i)_{\mathrm{n,n+1}}(n \neq 0, \mathrm{N})$ is the distance between the n^{th} task and the $n+1^{th}$ task.

The loss cost refers to the risk that the UCAV is attacked or destroyed by the enemy defense systems during the execution of tasks. It is related to the value of the UCAV, the defense ability of the UCAV, and the attack capability of the task.

Definition 3: The loss cost of the UCAV performing the task is equal to the attack capability of the task divided by the defense ability of the UCAV and multiplied by the value of the UCAV.

$$LossCost_{ij} = Value(U_i) \cdot Attack(T_j)/Defense(U_i) \tag{29}$$

where $LossCost_{ij}$ is the loss cost of the i^{th} UCAV performing the j^{th} task; $Value\ (U_i)$ is the value of the i^{th} UCAV; $Attack(T_j)$ is the attack capability of the j^{th} task and $Defence\ (U_i)$ is the defense ability of the i^{th} UCAV.

Considering the terrain fluctuation, take-off climb, battlefield threat avoidance, and other factors, the actual flight path cannot be a straight line. To simplify the task allocation model, the track length redundancy factor λ is introduced, and λ takes the value range of [1.3, 2] according to the battlefield complexity. In summary, the UCAV formation executing tasks cost can be expressed as follows:

$$Cost = \omega_1 \cdot \sum_{i=1}^{Nu} \lambda \cdot Length_i + \omega_2 \cdot \sum_{i=1}^{Nu}\sum_{j=1}^{Nt} x_{ij} \cdot \frac{LossCost_{ij}}{LossCost_{max}} \tag{30}$$

where $LostCost_{max}$ is the maximum loss value of a UCAV performing a task; ω_1 and ω_2 are the weight factor of *Discost* and *LostCost* respectively.

The task allocation problem of heterogeneous UCAVs is an optimization problem under multiple constraints. In order to simplify the problem and obtain higher solving efficiency, the constraint condition is introduced to the objective function in the form of punishment function Pu. Thus, the constrained optimization problem is transformed into unconstrained optimization problem for solving. In summary, the heterogeneous UCAV task allocation model is as follows:

$$max\ OE = Reward/Cost - \delta \cdot\ Pu \tag{31}$$

$$Pu = \begin{cases} 0, & results\ satisfy\ the\ constrains \\ 1, & results\ does\ not\ satisfy\ the\ constrains \end{cases} \tag{32}$$

$$s.t\ \sum_{i=1}^{Nu} x_{ij} = 1, \quad \forall j = 1, 2, \ldots, Nt \tag{33}$$

$$\sum_{i=1}^{Nu}\sum_{j=1}^{Nt} x_{ij} = Nt \tag{34}$$

$$\sum_{j=1}^{Nt} x_{ij} \leq Load_i, \quad \forall i = 1, 2, \ldots, Nu \tag{35}$$

$$\lambda \cdot Length_i \leq L(i)_{max}, \quad \forall i = 1, 2, \ldots, Nu \tag{36}$$

where δ is the penalty function scale factor, $Load_i$ is the maximum number of tasks performed by the i^{th} UCAV and $L(i)_{max}$ is the maximum flight distance of the i^{th} UCAV. (33) means that each task can only be executed by one UCAV; (34) means that all tasks are executed by UCAV formation; (35) means that the number of tasks assigned to UCAV cannot exceed its maximum executable tasks number; (36) means that the track length of all tasks performed by UCAV shall not exceed the maximum flight distance.

B. TASK ALLOCATION CODING

The whale optimization algorithm can be directly applied to continuous optimization problems, but the task allocation problem is a typical mixed integer linear programming problem, and the decision variables are discrete. Therefore, we must define the appropriate encoding to map the agent position to the task allocation result. In this paper, the mapping relationship between the agent position and the task allocation result is established based on the coding method of real vector. The mapping relationship is defined as follows: 1). The dimension of the problem is the task number, and the dimension subscript of the agent corresponds to the task subscript; 2). The search space of the solution is $(0,\ Nu)$; 3). The value, rounding the elements of the agent position to the nearest integers greater than or equal to it, correspond to the subscript of UCAV, and the first decimal part of the agent position corresponds to the order in which the UCAV performs tasks in ascending order.

For a clearer description of the mapping relationship, we take three UCAVs performing eight tasks as an example. In the example, the problem dimension is eight, the agent search range is $(0, 3)$, and the mapping relationship is as shown in Table 6.

V. TASK ALLOCATION RESULTS AND ANALYSIS

A. SIMULATION CONDITION

In order to verify the rationality of the heterogeneous UCAV task allocation model proposed in this paper, we program based on MATLAB 2013a and perform simulation experiments on a computer with the Intel(R)Core(TM) i7- 4770K CPU@3.50GHz 8GB RAM. The model is solved using IWOA, WOA, COA, VCS, CoBiDE, HFPSO, GWO and GDS-WOA, respectively. The parameters of the model are set as: the range of operational task space is 100km × 100km, $\lambda = 1.3, \omega_1 = 0.5, \omega_2 = 0.5, \delta = 5, L(i) = 600\mathrm{km}$

TABLE 6. The mapping relationship of task allocation coding.

Agent attribute	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	x_{i6}	x_{i7}	x_{i8}
Agent position	2.1	1.5	2.6	0.2	1.9	0.1	1.4	0.8
	UCAV number	Agent information		Task execution order				
Allocation results	U_1	0.1	x_{i6}	$T_6 \rightarrow T_4 \rightarrow T_8$				
		0.2	x_{i4}					
		0.8	x_{i8}					
	U_2	1.4	x_{i7}	$T_3 \rightarrow T_1 \rightarrow T_5$				
		1.5	x_{i2}					
		1.9	x_{i5}					
	U_3	2.1	x_{i1}	$T_9 \rightarrow T_6 \rightarrow T_2$				
		2.6	x_{i3}					

TABLE 7. Heterogeneous UCAVs information in case 1.

UCAV	Value	Position/km	Defense	Task type	$P(T)$
U_1	0.9	(10,20)	0.3	Reconnaissance	0.9
				Strike	0.2
				Evaluation	0.8
U_2	0.7	(30,10)	0.9	Reconnaissance	0.3
				Strike	0.9
				Evaluation	0.4
U_3	0.8	(15,9)	0.5	Reconnaissance	0.5
				Strike	0.5
				Evaluation	0.5

TABLE 8. Tasks information in case 1.

Task	Task type	Value	Position/km	Defense	Attack
T_1	Reconnaissance	0.4	(80,50)	0.2	0.8
T_2	Evaluation	0.5	(60,45)	0.4	0.6
T_3	Evaluation	0.9	(80,80)	0.9	0.1
T_4	Strike	0.8	(90,20)	0.7	0.3
T_5	Strike	0.9	(50,90)	0.8	0.2
T_6	Evaluation	0.5	(60,65)	0.4	0.6
T_7	Strike	0.7	(40,90)	0.7	0.3
T_8	Reconnaissance	0.6	(70,45)	0.2	0.8

$(i = 1, 2, 3, 4)$, $Load_i = 3(i = 1, 2, 3, 4)$. To simplify the task allocation model, the heterogeneity between UCAVs is reflected in the complete ability to perform different types of tasks. There are two cases with different number of UCAVs and tasks as the operational background.

In case 1, there is a UCAV formation with three UCAVs which needs to perform eight tasks. The information of UCAVs and tasks are shown as Table 7 and Table 8. The parameters of algorithms are set as follows: $dim = 8$, the agent search space is $(0, 3)$, $SN = 10$, $FE_{max} = 1000$. In case 2, there is a UCAV formation with 10 UCAVs which needs to perform 25 tasks. The information of UCAVs and tasks added to case 2 based on case 1 are shown as Table 9 and Table 10. The parameters of algorithms are set as follows: $dim = 25$, the agent search space is $(0, 8)$, $SN = 10$, $FE_{max} = 2000$. The relative positions of UCAVs and tasks in case 1 and case 2 are shown in Fig.6 and Fig.7 respectively. Compared to case 1, case 2 has a higher dimension and wider search range which can test the scalability of the algorithm when dealing with different dimensional problems.

TABLE 9. UCAVs information added to case 2 based on case 1.

UCAV	Value	Position/km	Defense	Task type	$P(T)$
U_4	0.8	(35,20)	0.7	Reconnaissance	0.8
				Strike	0.5
				Evaluation	0.7
U_5	0.7	(5,10)	0.8	Reconnaissance	0.7
				Strike	0.6
				Evaluation	0.8
U_6	0.6	(20,30)	0.9	Reconnaissance	0.9
				Strike	0.3
				Evaluation	0.8
U_7	0.9	(10,40)	0.6	Reconnaissance	0.7
				Strike	0.6
				Evaluation	0.7
U_8	0.8	(40,10)	0.7	Reconnaissance	0.5
				Strike	0.6
				Evaluation	0.6
U_9	0.7	(25,15)	0.6	Reconnaissance	0.8
				Strike	0.6
				Evaluation	0.7
U_{10}	0.7	(5,30)	0.8	Reconnaissance	0.9
				Strike	0.3
				Evaluation	0.7

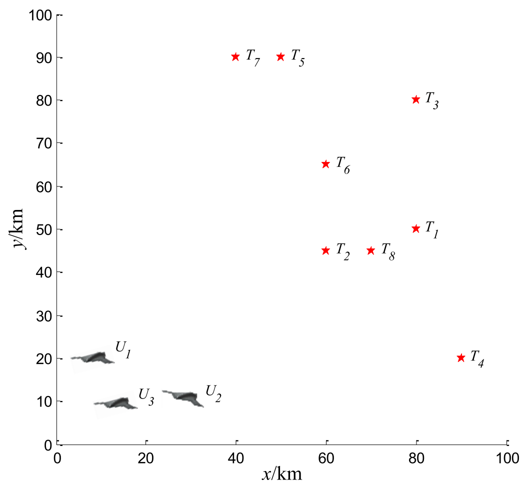

FIGURE 6. Relative positions of UCAVs and tasks in case 1.

B. SIMULATION RESULTS AND ANALYSIS

To avoid the contingency of experimental results, eight algorithms are executed 10 times independently in solving two cases based on the UCAV task allocation model. The results

TABLE 10. Tasks information added to case 2 based on case 1.

Task	Task type	Value	Position/km	Defense	Attack
T_9	Strike	0.8	(10,80)	0.8	0.2
T_{10}	Reconnaissance	0.7	(13,90)	0.3	0.7
T_{11}	Strike	0.9	(20,85)	0.9	0.1
T_{12}	Strike	0.7	(50,60)	0.8	0.2
T_{13}	Evaluation	0.5	(30,85)	0.4	0.6
T_{14}	Reconnaissance	0.6	(55,30)	0.4	0.6
T_{15}	Evaluation	0.5	(65,60)	0.6	0.4
T_{16}	Reconnaissance	0.7	(75,90)	0.5	0.5
T_{17}	Reconnaissance	0.6	(45,80)	0.4	0.6
T_{18}	Evaluation	0.6	(85,75)	0.5	0.5
T_{19}	Strike	0.9	(85,35)	0.7	0.3
T_{20}	Evaluation	0.6	(65,85)	0.6	0.4
T_{21}	Reconnaissance	0.6	(93,55)	0.5	0.5
T_{22}	Strike	0.8	(25,75)	0.9	0.1
T_{23}	Reconnaissance	0.6	(35,65)	0.4	0.6
T_{24}	Evaluation	0.5	(94,78)	0.5	0.5
T_{25}	Strike	0.8	(50,82)	0.8	0.2

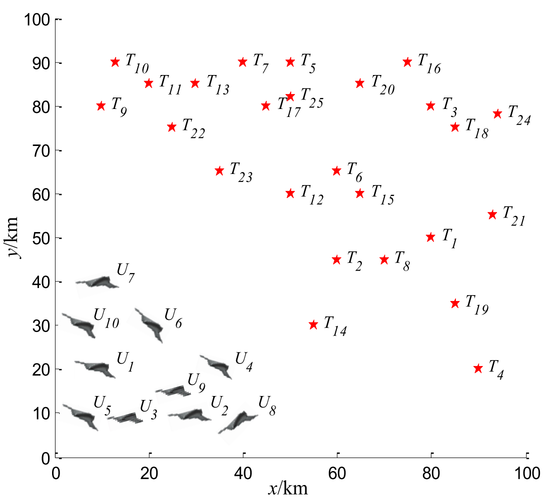

FIGURE 7. Relative positions of UCAVs and tasks in case 2.

TABLE 11. Task allocation results in case 1.

Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.3252	0.0666	0			1.4829	0.0620	0	
	1.4574	0.0624	0			1.4805	0.0627	0	
	1.4491	0.0721	0			1.4755	0.0635	0	
	1.4402	0.0642	0			1.4733	0.0593	0	
IWOA	1.3852	0.0641	0	100%	WOA	1.4733	0.0604	0	100%
	1.3762	0.0710	0			1.4723	0.0603	0	
	1.3752	0.0615	0			1.4486	0.0623	0	
	1.3668	0.0619	0			1.4435	0.0616	0	
	1.3497	0.0621	0			1.4413	0.0625	0	
	1.3299	0.0667	0			1.4381	0.0609	0	
MFS	1.3855	0.0653	Nan		MFS	1.4629	0.0616	Nan	
Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.5491	0.0790	0			1.5491	0.0864	0	
	1.5449	0.0798	0			1.5491	0.0893	0	
	1.5238	0.0794	0			1.4831	0.0857	0	
	1.5238	0.0782	0			1.5449	0.0890	0	
COA	1.5102	0.0769	0	100%	VCS	1.5280	0.0873	0	100%
	1.5028	0.0789	0			1.5263	0.0855	0	
	1.4942	0.0828	0			1.5028	0.0913	0	
	1.4926	0.0790	0			1.4942	0.0885	0	
	1.4904	0.0895	0			1.4926	0.0885	0	
	1.4904	0.0817	0			1.4834	0.0883	0	
MFS	1.5122	0.0805	Nan		MFS	1.5154	0.0880	Nan	
Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.5491	0.0843	0			1.5449	0.0786	0	
	1.5449	0.0834	0			1.5102	0.0782	0	
	1.5449	0.0832	0			1.4875	0.0805	0	
	1.5263	0.0849	0			1.4831	0.0843	0	
CoBiDE	1.5231	0.0854	0	100%	HFPSO	1.4755	0.0798	0	100%
	1.5231	0.0859	0			1.4755	0.0775	0	
	1.5102	0.0860	0			1.4682	0.0774	0	
	1.4904	0.0839	0			1.4682	0.0766	0	
	1.4875	0.0854	0			1.4527	0.0789	0	
	1.4875	0.0845	0			1.4397	0.0825	0	
MFS	1.5187	0.0847	Nan		MFS	1.4806	0.0794	Nan	
Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.5491	0.0610	0			1.5491	0.0825	0	
	1.5491	0.0589	0			1.5491	0.0799	0	
	1.5449	0.0620	0			1.5491	0.0824	0	
	1.4942	0.0586	0			1.5491	0.0902	0	
GWO	1.4942	0.0623	0	100%	GDS-WOA	1.5491	0.0782	0	100%
	1.4942	0.0649	0			1.5238	0.0810	0	
	1.4875	0.0592	0			1.5231	0.0808	0	
	1.4577	0.0621	0			1.5449	0.0809	0	
	1.4539	0.0612	0			1.5268	0.0817	0	
	1.4539	0.0675	0			1.5238	0.0817	0	
MFS	1.4979	0.0618	Nan		MFS	1.5388	0.0819	Nan	

of case 1 and case 2, consisting of operational effectiveness values (*OE*), running time/s, the value of *Pu* and the probability of obtaining feasible solutions (*P*) of algorithms, are shown in Table 11 and Table 13 respectively. It is noted that the "MFS" in Table 11 and Table 13 means the mean value of the solutions.

According to Table 11, all eight algorithms can obtain feasible solutions 100% that satisfy the constraints in solving case 1, which proves the validity and rationality of the heterogeneous UCAV task allocation model proposed in this paper. The average operational effectiveness values obtained by IWOA, WOA, COA, VCS, CoBiDE, HFPSO, GWO, and GDS-WOA are 1.3855, 1.4629, 1.5122, 1.5144, 1.5187, 1.4806, 1.4979 and 1.5388 respectively. It is easy to see that the performance of GDS-WOA is better than the other seven comparison algorithms when dealing with case 1 based on our proposed task allocation model.

Fig.8 shows the convergence curves of mean OE values of 10 task allocation results. The results indicate that GDS-WOA has a faster convergence speed. Fig.9 shows the running time of eight algorithms for solving the task allocation model 10 times in case 1 and Fig.10 is a box diagram of the operational effectiveness values obtained by the eight algorithms in Table 11. We have found that the results of GDS-WOA are more concentrated, so GDS-WOA is more stable when solving the problem. Overall, the operational effectiveness value of the task allocation results obtained by GDS-WOA is better than the comparison algorithms, but

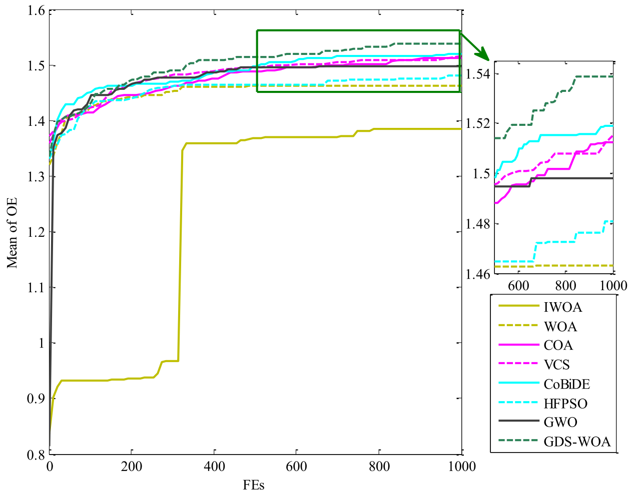

FIGURE 8. Convergence curves of mean OE values in case 1.

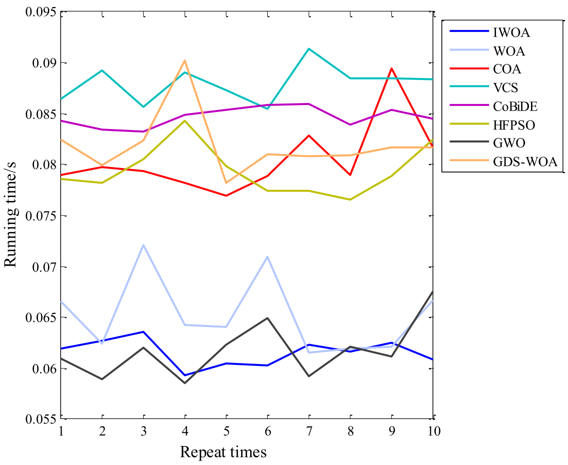

FIGURE 9. Running time in case 1.

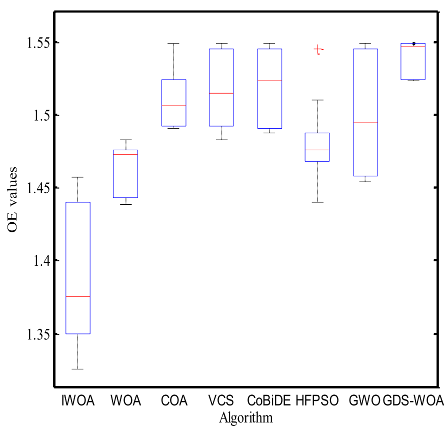

FIGURE 10. Box diagram of OE values in case 1.

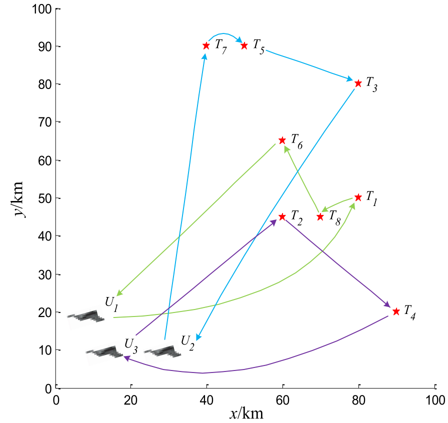

FIGURE 11. Optimal results obtained by IWOA.

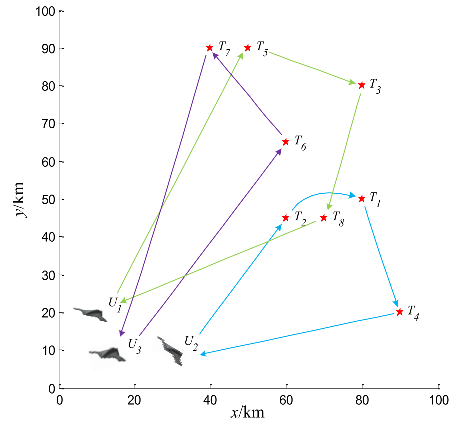

FIGURE 12. Optimal results obtained by WOA.

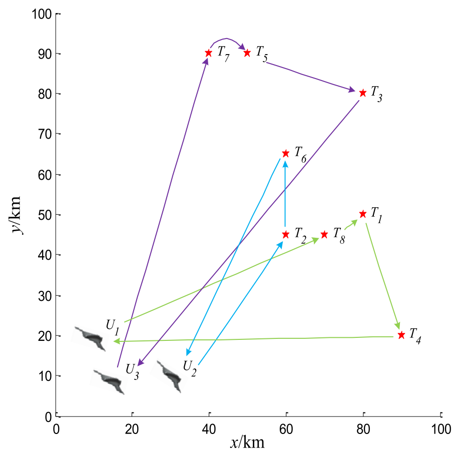

FIGURE 13. Optimal results obtained by COA.

the computational cost is worse than IWOA, WOA, HFPSO and GWO.

In order to compress the length of the paper, this paper only lists the detailed allocation results and task execution order of 3 better results in 10 times in case 1, which are shown in Table 12. For a more intuitive representation of task allocation results, Fig. 11 to Fig. 18 show the task execution track maps with the highest operational effectiveness value corresponding to IWOA, WOA, COA, VCS, CoBiDE, HFPSO, GWO and GDS-WOA in Table 12 respectively. In summary, the task allocation results obtained by GDS-WOA are better

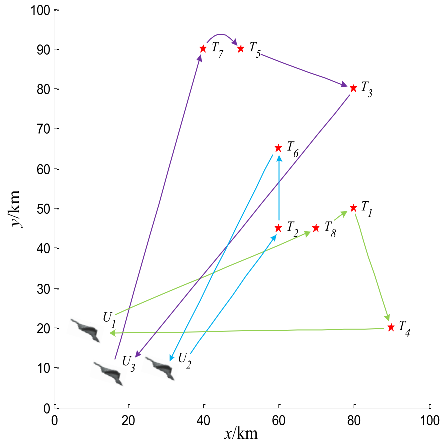

FIGURE 14. Optimal results obtained by VCS.

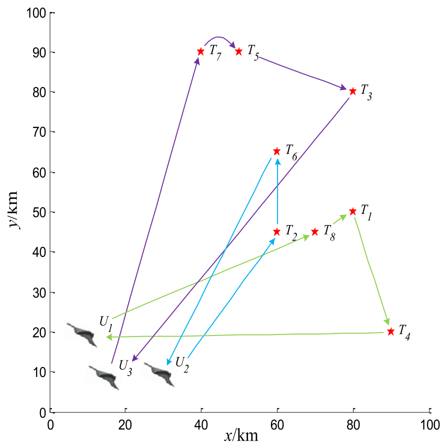

FIGURE 15. Optimal results obtained by CoBiDE.

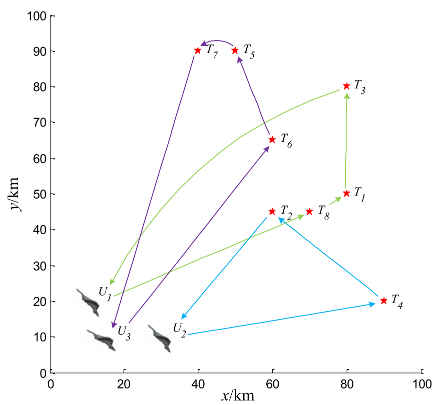

FIGURE 16. Optimal results obtained by HFPSO.

than the other seven comparison algorithms, which is consistent with the actual operational principles.

According to Table 13, when the algorithms solve the case 2 with 10 independent runs, only the solutions obtained by COA, CoBiDE and GDS-WOA are all feasible solutions. The success rates of VCS and GWO for feasible solutions are 80% and 90% respectively. What is worse is that the success rate of IWOA, WOA and HFPSO obtaining feasible solutions are only 10%, 10%, and 30% respectively. These results prove that GDS-WOA is also more robust when dealing

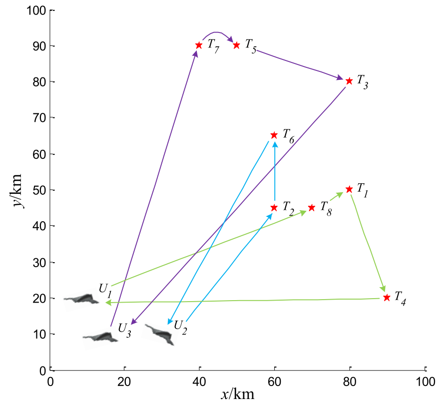

FIGURE 17. Optimal results obtained by GWO.

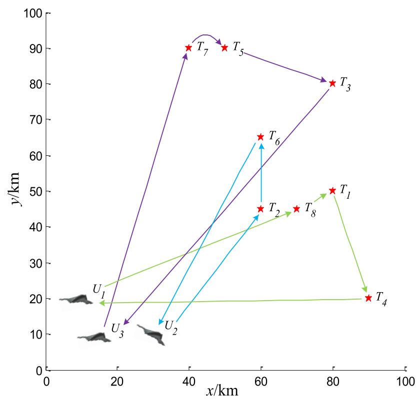

FIGURE 18. Optimal results obtained by GDS-WOA.

TABLE 12. Detail results of task allocation in case 1.

Algorithm	*OE*	Task allocation results and execution orders		
IWOA	1.4574	U_1: $T_1 \rightarrow T_8 \rightarrow T_6$	U_2: $T_7 \rightarrow T_5 \rightarrow T_3$	U_3: $T_2 \rightarrow T_4$
	1.4491	U_1: $T_1 \rightarrow T_8 \rightarrow T_7$	U_2: $T_2 \rightarrow T_4$	U_3: $T_6 \rightarrow T_3 \rightarrow T_5$
	1.4402	U_1: $T_4 \rightarrow T_8 \rightarrow T_6$	U_2: $T_1 \rightarrow T_3 \rightarrow T_2$	U_3: $T_5 \rightarrow T_7$
WOA	1.4829	U_1: $T_5 \rightarrow T_3 \rightarrow T_8$	U_2: $T_2 \rightarrow T_1 \rightarrow T_4$	U_3: $T_7 \rightarrow T_6$
	1.4805	U_1: $T_8 \rightarrow T_3 \rightarrow T_2$	U_2: $T_4 \rightarrow T_1$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
	1.4755	U_1: $T_8 \rightarrow T_5 \rightarrow T_7$	U_2: $T_2 \rightarrow T_6$	U_3: $T_3 \rightarrow T_1 \rightarrow T_4$
COA	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5449	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_4 \rightarrow T_2$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
	1.5238	U_1: $T_4 \rightarrow T_1 \rightarrow T_8$	U_2: $T_6 \rightarrow T_3 \rightarrow T_2$	U_3: $T_7 \rightarrow T_5$
VCS	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5449	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_4 \rightarrow T_2$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
CoBiDE	1..5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5449	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_4 \rightarrow T_2$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
	1.5449	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_4 \rightarrow T_2$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
HFPSO	1.5449	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_4 \rightarrow T_2$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
	1.5102	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_7 \rightarrow T_5 \rightarrow T_6$	U_3: $T_2 \rightarrow T_4$
	1.4875	U_1: $T_8 \rightarrow T_5 \rightarrow T_7$	U_2: $T_4 \rightarrow T_2$	U_3: $T_1 \rightarrow T_3 \rightarrow T_6$
GWO	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5449	U_1: $T_8 \rightarrow T_1 \rightarrow T_3$	U_2: $T_4 \rightarrow T_2$	U_3: $T_6 \rightarrow T_5 \rightarrow T_7$
GDS-WOA	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$
	1.5491	U_1: $T_8 \rightarrow T_1 \rightarrow T_4$	U_2: $T_2 \rightarrow T_6$	U_3: $T_7 \rightarrow T_5 \rightarrow T_3$

TABLE 13. Results of task allocation in case 2.

Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.5041	0.1966	0			1.5002	0.2049	0	
	-1.3818	0.1775	1			-1.2049	0.1854	1	
	-1.3754	0.1796	1			-1.1967	0.1769	1	
	-1.4567	0.1781	1			-0.8713	0.1793	1	
IWOA	-1.3706	0.1767	1	10%	WOA	-0.7554	0.1853	1	10%
	-1.3375	0.1803	1			-0.8506	0.1833	1	
	-1.3214	0.1795	1			-0.6711	0.1764	1	
	-1.7369	0.1759	1			-0.6508	0.1837	1	
	-1.2597	0.1866	1			-0.4537	0.1754	1	
	-1.2816	0.1775	1			-0.4473	0.1781	1	
MFS	-1.1018	0.1808	Nan		MFS	-0.5602	0.1829	Nan	
Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.6350	0.2233	0			1.6489	0.3189	0	
	1.5958	0.2172	0			1.6185	0.3290	0	
	1.5566	0.2182	0			1.5910	0.3086	0	
	1.5496	0.2195	0			1.5473	0.3524	0	
COA	1.5301	0.2204	0	100%	VCS	1.5324	0.3456	0	80%
	1.5252	0.2219	0			1.4911	0.3137	0	
	1.5234	0.2143	0			1.4804	0.2914	0	
	1.5232	0.2211	0			1.4750	0.2892	0	
	1.5108	0.2122	0			-2.0468	0.3078	1	
	1.5071	0.2245	0			-0.8471	0.2966	1	
MFS	1.5457	0.2193	Nan		MFS	0.9491	0.3153	Nan	
Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.6387	0.2401	0			1.5582	0.2690	0	
	1.6367	0.2378	0			1.5355	0.2719	0	
	1.6103	0.2381	0			1.4196	0.2727	0	
	1.6076	0.2465	0			-1.8670	0.2550	1	
CoBiDE	1.5842	0.2365	0	100%	HFPSO	-1.8230	0.2471	1	30%
	1.5691	0.2388	0			-1.6282	0.2508	1	
	1.5682	0.2437	0			-1.6529	0.2533	1	
	1.5448	0.2363	0			-1.4484	0.2534	1	
	1.5308	0.2475	0			-1.7231	0.2525	1	
	1.5074	0.2469	0			-2.4621	0.2609	1	
MFS	1.5798	0.2412	Nan		MFS	-0.8091	0.2587	Nan	
Algorithm	*OE*	Time/s	*Pu*	*P*	Algorithm	*OE*	time/s	*Pu*	*P*
	1.6675	0.1886	0			1.6230	0.2469	0	
	1.5843	0.2001	0			1.6350	0.2583	0	
	1.5691	0.1872	0			1.6520	0.2454	0	
	1.5447	0.1909	0			1.6983	0.2449	0	
GWO	1.5446	0.1871	0	90%	GDS-WOA	1.6853	0.2446	0	100%
	1.5194	0.1878	0			1.6844	0.2444	0	
	1.5016	0.2017	0			1.6736	0.2546	0	
	1.4859	0.1885	0			1.6732	0.2515	0	
	1.4619	0.1867	0			1.6694	0.2485	0	
	-2.2689	0.1837	1			1.6612	0.2428	0	
MFS	1.1610	0.1902	Nan		MFS	1.6655	0.2482	Nan	

with higher-dimensional task allocation problems. The average operational effectiveness value obtained by GDS-WOA is better than the compared algorithms, which proves the excellent optimization performance of GDS-WOA in solving complex constrained optimization problems. The detailed allocation results and task execution orders of 3 better results in 10 times in case 2 are shown in Table 14.

Fig.19 and Fig.20 show the convergence curves of mean OE values of 10 task allocation results and the running time of the eight algorithms for solving the task allocation model in case 2. The task allocation results obtained by GDS-WOA has faster convergence speed and higher operational effectiveness value than the comparison algorithms, but the computational cost is worse than IWOA, WOA, CoBiDE,

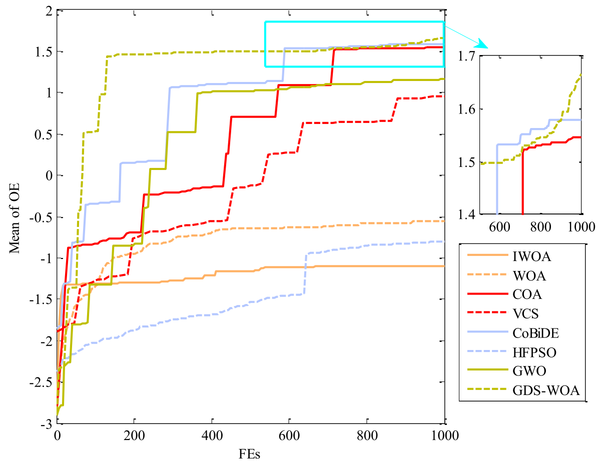

FIGURE 19. Convergence curves of mean OE values in case 2.

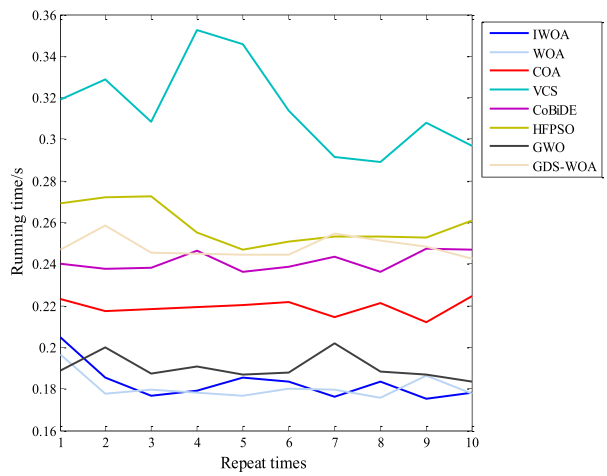

FIGURE 20. Computational costs in case 2.

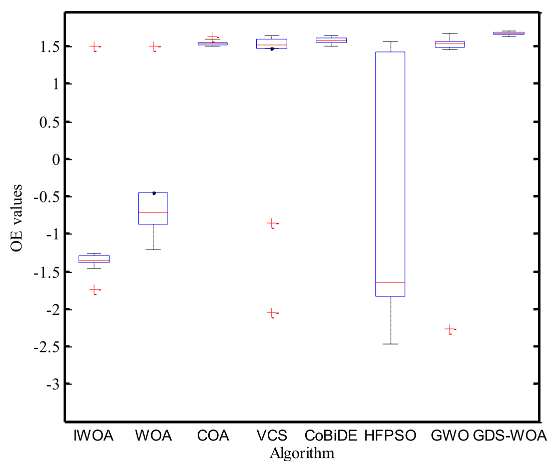

FIGURE 21. Box diagrams of OE values in case 2.

HFPSO, and GWO. Fig.21 is a box diagram of the operational effectiveness values obtained by the eight algorithms in Table 12. According to Fig.21, we have found that the

TABLE 14. Detail results of task allocation in case 2.

Algorithm	*OE*	Task allocation results and execution orders									
IWOA	1.5041	$U_1{:}T_7\rightarrow T_5\rightarrow T_{20}$	$U_2{:}T_{21}\rightarrow T_4\rightarrow T_{14}$	$U_3{:}T_3\rightarrow T_{13}$	$U_4{:}T_{10}\rightarrow T_{24}$	$U_5{:}T_6$	$U_6{:}T_{15}\rightarrow T_1\rightarrow T_{23}$	$U_7{:}T_{25}\rightarrow T_{18}\rightarrow T_{12}$	$U_8{:}T_8\rightarrow T_{17}\rightarrow T_{19}$	$U_9{:}T_2\rightarrow T_{22}$	$U_{10}{:}T_9\rightarrow T_{11}\rightarrow T_{16}$
	-1.2597	*Not a feasible solution*									
	-1.2816	*Not a feasible solution*									
WOA	1.5002	$U_1{:}T_{10}\rightarrow T_{25}\rightarrow T_{16}$	U_2:	$U_3{:}T_7\rightarrow T_{23}\rightarrow T_2$	$U_4{:}T_{22}\rightarrow T_6\rightarrow T_{14}$	$U_5{:}T_{11}$	$U_6{:}T_{21}\rightarrow T_9\rightarrow T_{19}$	$U_7{:}T_{13}\rightarrow T_8\rightarrow T_{15}$	$U_8{:}T_{17}\rightarrow T_4\rightarrow T_{20}$	$U_9{:}T_{18}\rightarrow T_3\rightarrow T_5$	$U_{10}{:}T_{24}\rightarrow T_1\rightarrow T_{12}$
	-0.4473	*Not a feasible solution*									
	-0.4537	*Not a feasible solution*									
COA	1.6350	$U_1{:}T_8\rightarrow T_{16}\rightarrow T_{11}$	$U_2{:}T_{15}\rightarrow T_{24}$	$U_3{:}T_1\rightarrow T_{23}\rightarrow T_7$	$U_4{:}T_{14}$	$U_5{:}T_{25}\rightarrow T_3\rightarrow T_{22}$	$U_6{:}T_9\rightarrow T_6\rightarrow T_{17}$	$U_7{:}T_{20}\rightarrow T_{18}$	$U_8{:}T_5\rightarrow T_{12}$	$U_9{:}T_4\rightarrow T_{19}\rightarrow T_{21}$	$U_{10}{:}T_2\rightarrow T_{13}\rightarrow T_{10}$
	1.5958	$U_1{:}T_{12}\rightarrow T_{10}\rightarrow T_3$	$U_2{:}T_4\rightarrow T_{19}\rightarrow T_{15}$	$U_3{:}T_7\rightarrow T_5\rightarrow T_{25}$	$U_4{:}T_{16}\rightarrow T_{13}\rightarrow T_{22}$	$U_5{:}T_1$	$U_6{:}T_{17}\rightarrow T_{21}\rightarrow T_{24}$	$U_7{:}T_{23}\rightarrow T_{11}\rightarrow T_9$	$U_8{:}T_6\rightarrow T_{18}\rightarrow T_8$	$U_9{:}T_{14}$	$U_{10}{:}T_2\rightarrow T_{20}$
	1.5566	$U_1{:}T_{14}\rightarrow T_{18}\rightarrow T_{13}$	$U_2{:}T_2\rightarrow T_3\rightarrow T_{19}$	$U_3{:}T_{12}\rightarrow T_{11}$	$U_4{:}T_{21}\rightarrow T_{16}\rightarrow T_{24}$	$U_5{:}T_1\rightarrow T_9\rightarrow T_{20}$	$U_6{:}T_4\rightarrow T_{10}$	U_7: *No task*	$U_8{:}T_{15}\rightarrow T_6\rightarrow T_5$	$U_9{:}T_{22}\rightarrow T_{25}\rightarrow T_{23}$	$U_{10}{:}T_{17}\rightarrow T_7\rightarrow T_8$
VCS	1.6489	$U_1{:}T_{12}\rightarrow T_{11}\rightarrow T_{16}$	$U_2{:}T_{15}\rightarrow T_9$	U_3: *No task*	$U_4{:}T_4\rightarrow T_{24}\rightarrow T_{18}$	$U_5{:}T_{20}\rightarrow T_1\rightarrow T_2$	$U_6{:}T_{23}\rightarrow T_{13}\rightarrow T_8$	$U_7{:}T_{10}\rightarrow T_{22}\rightarrow T_5$	$U_8{:}T_3\rightarrow T_7\rightarrow T_6$	$U_9{:}T_{21}\rightarrow T_{25}\rightarrow T_{19}$	$U_{10}{:}T_{14}\rightarrow T_{17}$
	1.6189	$U_1{:}T_{18}\rightarrow T_{19}\rightarrow T_5$	U_2: *No task*	$U_3{:}T_{22}\rightarrow T_{23}\rightarrow T_{20}$	$U_4{:}T_{16}\rightarrow T_{15}\rightarrow T_{10}$	$U_5{:}T_{11}\rightarrow T_{25}\rightarrow T_{17}$	$U_6{:}T_8\rightarrow T_3\rightarrow T_{13}$	$U_7{:}T_1\rightarrow T_{12}\rightarrow T_7$	$U_8{:}T_{14}$	$U_9{:}T_4\rightarrow T_{24}\rightarrow T_6$	$U_{10}{:}T_{21}\rightarrow T_2\rightarrow T_9$
	1.5910	$U_1{:}T_{14}\rightarrow T_{17}\rightarrow T_{13}$	$U_2{:}T_{22}\rightarrow T_5\rightarrow T_{15}$	$U_3{:}T_1\rightarrow T_{16}\rightarrow T_3$	$U_4{:}T_{23}\rightarrow T_{11}\rightarrow T_{25}$	$U_5{:}T_4\rightarrow T_{21}\rightarrow T_{19}$	$U_6{:}T_{20}\rightarrow T_{10}\rightarrow T_7$	$U_7{:}\ T_9$	$U_8{:}T_{18}\rightarrow T_{24}$	$U_9{:}T_8\rightarrow T_{12}$	$U_{10}{:}T_2\rightarrow T_6$
CoBiDE	1.6387	$U_1{:}T_3\rightarrow T_{21}\rightarrow T_{24}$	U_2: *No task*	$U_3{:}T_{14}\rightarrow T_{19}$	$U_4{:}T_5\rightarrow T_8\rightarrow T_7$	$U_5{:}T_9\rightarrow T_{11}$	$U_6{:}T_{10}\rightarrow T_1\rightarrow T_6$	$U_7{:}T_{12}\rightarrow T_{20}\rightarrow T_{18}$	$U_8{:}T_4\rightarrow T_{25}\rightarrow T_{15}$	$U_9{:}T_{17}\rightarrow T_{16}\rightarrow T_{13}$	$U_{10}{:}T_2\rightarrow T_{22}\rightarrow T_{23}$
	1.6367	$U_1{:}T_{22}\rightarrow T_{17}\rightarrow T_{16}$	U_2: *No task*	$U_3{:}T_{19}\rightarrow T_1$	$U_4{:}T_{15}\rightarrow T_{24}$	$U_5{:}T_4\rightarrow T_{21}\rightarrow T_6$	$U_6{:}T_{23}\rightarrow T_2\rightarrow T_{13}$	$U_7{:}T_{14}\rightarrow T_8\rightarrow T_9$	$U_8{:}T_{10}\rightarrow T_{11}\rightarrow T_7$	$U_9{:}T_{25}\rightarrow T_5\rightarrow T_{20}$	$U_{10}{:}T_3\rightarrow T_{12}\rightarrow T_{18}$
	1.6103	$U_1{:}T_9\rightarrow T_{25}\rightarrow T_{21}$	$U_2{:}T_{20}$	$U_3{:}T_7\rightarrow T_{11}$	$U_4{:}T_{18}\rightarrow T_{24}$	$U_5{:}T_{16}\rightarrow T_{17}\rightarrow T_1$	$U_6{:}T_5\rightarrow T_2\rightarrow T_{14}$	$U_7{:}T_{12}\rightarrow T_{15}\rightarrow T_{19}$	$U_8{:}T_8\rightarrow T_4$	$U_9{:}T_{22}\rightarrow T_{13}\rightarrow T_3$	$U_{10}{:}T_{23}\rightarrow T_6\rightarrow T_{10}$
HFPSO	1.5582	$U_1{:}T_{22}\rightarrow T_{10}\rightarrow T_1$	$U_2{:}T_{17}\rightarrow T_{12}\rightarrow T_{14}$	$U_3{:}T_3\rightarrow T_4\rightarrow T_2$	$U_4{:}T_5\rightarrow T_{13}\rightarrow T_{11}$	$U_5{:}T_{23}\rightarrow T_7\rightarrow T_9$	$U_6{:}T_{15}\rightarrow T_{19}$	$U_7{:}T_{25}\rightarrow T_{20}\rightarrow T_{16}$	$U_8{:}T_{18}\rightarrow T_{24}\rightarrow T_{21}$	$U_9{:}T_6$	$U_{10}{:}T_8$
	1.5355	$U_1{:}T_{20}\rightarrow T_{16}\rightarrow T_{24}$	$U_2{:}T_{23}\rightarrow T_2$	$U_3{:}T_4\rightarrow T_{25}$	$U_4{:}T_1\rightarrow T_{22}$	$U_5{:}T_8\rightarrow T_7\rightarrow T_9$	$U_6{:}T_{12}\rightarrow T_{15}\rightarrow T_{18}$	$U_7{:}T_5\rightarrow T_{11}\rightarrow T_{10}$	$U_8{:}T_6\rightarrow T_{17}$	$U_9{:}T_{19}\rightarrow T_{21}\rightarrow T_3$	$U_{10}{:}T_{14}\rightarrow T_{13}$
	1.4196	$U_1{:}T_{23}\rightarrow T_{22}$	$U_2{:}T_4\rightarrow T_{17}\rightarrow T_{13}$	$U_3{:}T_8\rightarrow T_1\rightarrow T_{20}$	$U_4{:}T_{18}\rightarrow T_6$	$U_5{:}T_9\rightarrow T_5\rightarrow T_{12}$	$U_6{:}T_7\rightarrow T_{10}\rightarrow T_{19}$	$U_7{:}T_{14}\rightarrow T_{24}\rightarrow T_{15}$	$U_8{:}T_{11}\rightarrow T_3$	$U_9{:}T_2\rightarrow T_{25}$	$U_{10}{:}T_{16}\rightarrow T_{21}$
GWO	1.6675	$U_1{:}T_{10}\rightarrow T_2\rightarrow T_{19}$	$U_2{:}T_{21}\rightarrow T_{24}$	$U_3{:}T_9\rightarrow T_4\rightarrow T_3$	$U_4{:}T_6\rightarrow T_{15}\rightarrow T_{20}$	$U_5{:}T_{16}\rightarrow T_{12}\rightarrow T_{11}$	$U_6{:}T_{13}\rightarrow T_{25}\rightarrow T_{23}$	$U_7{:}T_7\rightarrow T_5\rightarrow T_{17}$	$U_8{:}T_{18}\rightarrow T_{22}$	U_9: *No task*	$U_{10}{:}T_{14}\rightarrow T_8\rightarrow T_1$
	1.5843	$U_1{:}T_{19}\rightarrow T_{24}\rightarrow T_{16}$	$U_2{:}T_4\rightarrow T_{18}$	$U_3{:}T_{10}\rightarrow T_{25}\rightarrow T_{14}$	$U_4{:}T_7\rightarrow T_{15}\rightarrow T_8$	$U_5{:}T_2\rightarrow T_3\rightarrow T_{21}$	$U_6{:}T_1\rightarrow T_{23}\rightarrow T_{21}$	$U_7{:}T_5\rightarrow T_6\rightarrow T_{12}$	$U_8{:}T_{13}\rightarrow T_9\rightarrow T_{17}$	U_9: *No task*	$U_{10}{:}T_{11}\rightarrow T_{22}$
	1.5691	$U_1{:}T_3\rightarrow T_{14}$	$U_2{:}T_{24}\rightarrow T_{21}\rightarrow T_{15}$	$U_3{:}T_2\rightarrow T_4\rightarrow T_6$	$U_4{:}T_5\rightarrow T_{11}$	$U_5{:}T_9\rightarrow T_7\rightarrow T_{22}$	$U_6{:}T_{10}\rightarrow T_8\rightarrow T_{17}$	$U_7{:}T_{23}\rightarrow T_{13}\rightarrow T_{20}$	$U_8{:}T_{25}\rightarrow T_{18}$	$U_9{:}T_{12}$	$U_{10}{:}T_1\rightarrow T_{16}\rightarrow T_{19}$
GDS-WOA	1.6983	$U_1{:}T_{12}\rightarrow T_3\rightarrow T_{16}$	$U_2{:}T_{14}$	$U_3{:}T_{11}$	$U_4{:}T_{13}\rightarrow T_{17}\rightarrow T_{10}$	$U_5{:}T_{18}\rightarrow T_{20}\rightarrow T_7$	$U_6{:}T_{24}\rightarrow T_2\rightarrow T_8$	$U_7{:}T_{23}\rightarrow T_6\rightarrow T_{25}$	$U_8{:}T_4\rightarrow T_1\rightarrow T_{19}$	$U_9{:}T_9\rightarrow T_{22}$	$U_{10}{:}T_5\rightarrow T_{21}\rightarrow T_{15}$
	1.6853	$U_1{:}T_9\rightarrow T_{21}\rightarrow T_{12}$	$U_2{:}T_{18}\rightarrow T_8$	$U_3{:}T_{11}\rightarrow T_{25}\rightarrow T_{17}$	$U_4{:}T_{14}\rightarrow T_4\rightarrow T_{15}$	$U_5{:}T_{13}\rightarrow T_{20}\rightarrow T_{16}$	$U_6{:}T_2\rightarrow T_1\rightarrow T_{23}$	$U_7{:}T_{19}\rightarrow T_7\rightarrow T_{22}$	U_8: *No task*	$U_9{:}T_3\rightarrow T_{24}$	$U_{10}{:}T_{10}\rightarrow T_5\rightarrow T_6$
	1.6844	$U_1{:}T_{25}\rightarrow T_{13}\rightarrow T_6$	$U_2{:}T_{20}\rightarrow T_{14}$	U_3: *No task*	$U_4{:}T_{12}\rightarrow T_{15}\rightarrow T_5$	$U_5{:}T_{22}\rightarrow T_{11}\rightarrow T_3$	$U_6{:}T_{10}\rightarrow T_8\rightarrow T_2$	$U_7{:}T_{21}\rightarrow T_7$	$U_8{:}T_{18}\rightarrow T_{16}\rightarrow T_{19}$	$U_9{:}T_1\rightarrow T_{24}\rightarrow T_4$	$U_{10}{:}T_{17}\rightarrow T_9\rightarrow T_{23}$

results of GDS-WOA are more concentrated and no outlier, so GDS-WOA is still more stable when solving case 2.

It is noted that although GDS-WOA is superior to the comparison algorithms in obtaining the operational effectiveness values of the solution, the time cost is sacrificed. Nowadays, the real-time task allocation usually adopts rolling time-domain method, and the time window is set to one second which could meet the actual demand. In contrast, the running time of the algorithm is on the microsecond level. Therefore, GDS-WOA sacrifices a certain computational cost to obtain greater operational effectiveness value, which has practical significance for improving the operational effectiveness value of heterogeneous UCAV formation.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an adaptive WOA variant based on Gaussian distribution strategies. Our proposed GDS-WOA is evaluated based on CEC 2014 test suite compared with its variant and other five state-of-the-art competitive algorithms from different communities. The results show the superior performance of GDS-WOA in terms of the solution accuracy and stability.

Taking the factors affecting task allocation in actual combat into consideration, this paper constructs the heterogeneous UCAV collaborative operation task allocation model based on previous research, and GDS-WOA is applied to solve the problem. The simulation results demonstrate GDS-WOA is better than the other compared algorithms in terms of stability and operational effectiveness value. For the small-scale task allocation problem, all eight algorithms can obtain feasible solutions 100%, which verifies the validity of the established model. However, for a large-scale task allocation problem, only COA, CoBiDE, and GDS-WOA can obtain feasible solutions 100%. Moreover, GDS-WOA is able to obtain higher operational effectiveness values than the comparison algorithms when solving both large-scale and small-scale task allocation problems, showing that GDS-WOA outperforms to the compared algorithms.

Our proposed GDS-WOA has shown the superior performance when dealing with constrained and unconstrained problems. However, while improving the performance of the GDS-WOA, it sacrifices computational cost to some extent. In the next study, we will work to reduce the computational cost by optimizing the algorithm architecture and apply the improved strategies in this paper to other optimization algorithms.

REFERENCES

[1] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, Oct. 2002, pp. 39–43.

[3] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[4] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Apr. 1997.

[5] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[6] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003.

[7] P. D. P. Reddy, V. C. V. Reddy, and T. G. Manohar, "Whale optimization algorithm for optimal sizing of renewable resources for loss reduction in distribution systems," *Renewables, Wind, Water, Solar*, vol. 4, no. 1, p. 3, Dec. 2017.

[8] W. Sun and C. Zhang, "Analysis and forecasting of the carbon price using multi—Resolution singular value decomposition and extreme learning machine optimized by adaptive whale optimization algorithm," *Appl. Energy*, vol. 231, pp. 1354–1371, Dec. 2018.

[9] M. Tubishat and N. Idris, "Explicit and implicit aspect extraction using whale optimization algorithm and hybrid approach," in *Proc. Int. Conf. Ind. Enterprise Syst. Eng. (IcoIESE)*, 2019, pp. 208–213.

[10] J. Nasiri and F. M. Khiyabani, "A whale optimization algorithm (WOA) approach for clustering," *Cogent Math. Stat.*, vol. 5, no. 1, Jun. 2018, Art. no. 1483565.

[11] Y. Zheng, Y. Li, G. Wang, Y. Chen, Q. Xu, J. Fan, and X. Cui, "A novel hybrid algorithm for feature selection based on whale optimization algorithm," *IEEE Access*, vol. 7, pp. 14908–14923, 2018.
[12] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *J. Comput. Des. Eng.*, vol. 5, no. 3, pp. 275–284, Apr. 2018.
[13] H. Hu, Y. Bai, and T. Xu, "A whale optimization algorithm with inertia weight," *WSEAS Trans. Comput.*, vol. 15, pp. 319–326, 2016.
[14] I. N. Trivedi, J. Pradeep, J. Narottam, K. Arvind, and L. Dilip, "Novel adaptive whale optimization algorithm for global optimization," *Indian J. Sci. Technol.*, vol. 9, no. 38, pp. 1–8, Oct. 2016.
[15] M. A. Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowl.-Based Syst.*, vol. 172, pp. 42–63, May 2019.
[16] Y. Khalil, M. Alshayeji, and I. Ahmad, "Distributed whale optimization algorithm based on MapReduce," *Concurr. Comput. Pract. Exp.*, vol. 31, no. 1, p. e4872, May 2019.
[17] H. Chen, Y. Xu, M. Wang, and X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," *Appl. Math. Model.*, vol. 71, pp. 45–59, Jul. 2019.
[18] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neuro Comput.*, vol. 260, pp. 302–312, Oct. 2017.
[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
[20] S. M. Bozorgi and S. Yazdani, "IWOA: An improved whale optimization algorithm for optimization problems," *J. Comput. Des. Eng.*, vol. 6, no. 3, pp. 243–259, Jul. 2019.
[21] I. N. Trivedi, P. Jangir, A. Kumar, N. Jangir, and R. Totlani, "A novel hybrid PSO–WOA algorithm for global numerical functions optimization," in *Advances in Intelligent Systems and Computing*. Singapore: Springer, 2018, pp. 53–60.
[22] N. Singh and H. Hachimi, "A new hybrid whale optimizer algorithm with mean strategy of grey wolf optimizer for global optimization," *Math. Comput. Appl.*, vol. 23, no. 1, p. 14, Mar. 2018.
[23] F. Mohamed, M. AbdelNasser, K. Mahmoud, and S. Kamel, "Accurate economic dispatch solution using hybrid whale-wolf optimization method," in *Proc. 19th Int. Middle East Power Syst. Conf. (MEPCON)*, 2017, pp. 922–927.
[24] B. Bentouati, L. Chaib, and S. Chettih, "A hybrid whale algorithm and pattern search technique for optimal power flow problem," in *Proc. 8th Int. Conf. Modelling, Identificat. Control (ICMIC)*, 2016, pp. 1048–1053.
[25] S. Khalilpourazari and S. Khalilpourazary, "SCWOA: An efficient hybrid algorithm for parameter optimization of multi-pass milling process," *J. Ind. Prod. Eng.*, vol. 35, no. 3, pp. 135–147, Apr. 2018.
[26] S. T. Revathi, N. Ramaraj, and S. Chithra, "Brain storm-based whale optimization algorithm for privacy-protected data publishing in cloud computing," in *Cluster Computing*. New York, NY, USA: Springer, Feb. 2018, pp. 1–10.
[27] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, vol. 2. Boston, MA, USA: Springer, 2001.
[28] P. A. N. Bosman, J. Grahl, and D. Thierens, "Enhancing the performance of maximum–likelihood Gaussian EDAs using anticipated mean shift," in *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2008, pp. 133–143.
[29] Z. Ren, C. He, D. Zhong, S. Huang, and Y. Liang, "Enhance continuous estimation of distribution algorithm by variance enlargement and reflecting sampling," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3441–3447.
[30] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1769–1776.
[31] X. Wang, H. Zhao, T. Han, Z. Wei, Y. Liang, and Y. Li, "A Gaussian estimation of distribution algorithm with random walk strategies and its application in optimal missile guidance handover for multi-UCAV in over-the-horizon air combat," *IEEE Access*, vol. 7, pp. 43298–43317, 2019.
[32] X. Wang, H. Zhao, T. Han, H. Zhou, and C. Li, "A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem," *Appl. Soft Comput.*, vol. 78, pp. 240–260, May 2019.
[33] Z.-Z. Liu, Y. Wang, S. Yang, and K. Tang, "An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1403–1416, Apr. 2019.
[34] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," in *Proc. KanGAL*, 2013, pp. 1–32.
[35] M. Li, H. Zhao, X. Weng, and T. Han, "Cognitive behavior optimization algorithm for solving optimization problems," *Appl. Soft Comput.*, vol. 39, pp. 199–222, Feb. 2016.
[36] M. D. Li, H. Zhao, X. W. Weng, and T. Han, "A novel nature-inspired algorithm for optimization: Virus colony search," *Adv. Eng. Softw.*, vol. 92, pp. 65–88, Feb. 2016.
[37] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Appl. Soft. Comput.*, vol. 18, pp. 232–247, May 2014.
[38] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft Comput.*, vol. 66, pp. 232–249, May 2018.
[39] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[40] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the fbietkan statistic," *Commun. Stat.-Theory Methods*, vol. 9, no. 6, pp. 571–595, Jan. 1980.
[41] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Dept. Math. Statist., Princeton Univ., Princeton, NJ, USA, 1963.
[42] J. Omer and J.-L. Farges, "Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1218–1230, Sep. 2013.
[43] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Trans. Syst., Man, Cybern C, Appl. Rev.*, vol. 42, no. 5, pp. 682–691, Sep. 2012.
[44] J.-T. Li, S. Zhang, Z. Zheng, L.-N. Xing, and R.-J. He, "Research on multi-UAV loading multi-type sensors cooperative reconnaissance task planning based on genetic algorithm," in *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Cham, Switzerland: Springer, 2017, pp. 485–500.
[45] G. Quartarone and N. Anglani, "Mixed-integer linear programming for the optimal design and management of air compressor rooms," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Oct. 2014, pp. 1–9.
[46] E. I. Grøtli and T. A. Johansen, "Path planning for UAVs under communication constraints using SPLAT! And MILP," *J. Intell. Robot. Syst.*, vol. 65, nos. 1–4, pp. 265–282, Jan. 2012.
[47] W. Wu, N. Cui, W. Shan, and X. Wang, "Distributed task allocation for multiple heterogeneous UAVs based on consensus algorithm and online cooperative strategy," *Aircr. Eng. Aerosp. Technol.*, vol. 90, no. 9, pp. 1464–1473, Nov. 2018.
[48] Y. Zhong, P. Yao, Y. Sun, and J. Yang, "Cooperative task allocation method of MCAV/UCAV formation," *Math. Problems Eng.*, vol. 2016, Nov. 2016, Art. no. 6051046.
[49] H. A. Kurdi, E. Aloboud, M. Alalwan, S. Alhassan, E. Alotaibi, G. Bautista, and J. P. How, "Autonomous task allocation for multi-UAV systems based on the locust elastic behavior," *Appl. Soft Comput.*, vol. 71, pp. 110–126, Oct. 2018.
[50] J. Li, K. Zhang, and G. Xia, "Multi-AUV cooperative task allocation based on improved contract network," in *Proc. IEEE Int. Conf. Mechatron. Automat. (ICMA)*, Aug. 2017, pp. 608–613.
[51] H. Huang and T. Zhuo, "Multi-model cooperative task assignment and path planning of multiple UCAV formation," *Multimedia Tools Appl.*, vol. 78, no. 1, pp. 415–436, Jan. 2019.
[52] L. Huang, H. Qu, and L. Zuo, "Multi-type UAVs cooperative task allocation under resource constraints," *IEEE Access*, vol. 6, pp. 17841–17850, 2018.
[53] Y. Miao, L. Zhong, Y. Yin, C. Zou, and Z. Luo, "Research on dynamic task allocation for multiple unmanned aerial vehicles," *Trans. Inst. Meas. Control*, vol. 39, no. 4, pp. 466–474, Apr. 2017.
[54] N. Zhi-Qiang, Z. Rong-gang, and H. Peng, "UCAVs cooperative task allocation method based on immune evolutionary computation," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2017, pp. 55–60.
[55] M. Xu, S. Wang, J. Tao, and G. Liang, "Research on cooperative task allocation for multiple UCAVs based on modified co-evolutionary genetic algorithm," in *Proc. Int. Conf. Comput. Inf. Sci.*, Jun. 2013, pp. 125–128.
[56] S. Wang, J. He, Y. Huang, and A. Wang, "Cooperative task assignment of uninhabited combat air vehicles based on improved MOSFLA algorithm," in *Proc. 3rd Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2016, pp. 427–432.

YINTONG LI received the B.S. degree in weapon engineering from Air Force Engineering University, Xi'an, China, in 2017, where he is currently pursuing the M.S. degree. His current research interests include artificial intelligence optimization algorithms, and UCAV combat systems and technologies.

TONG HAN received the Ph.D. degree in weapon science and technology from Air Force Engineering University, Xi'an, China, in 2008, where he is currently an Associate Professor. He has engaged in teaching and researching on weapon systems and application engineering, and multi-objective optimization.

HUI ZHAO received the Ph.D. degree in weapon science and technology from Air Force Engineering University, Xi'an, China, in 2011, where he is currently a Professor. He has engaged in teaching and researching on weapon systems and application engineering, and optimal control.

HANJIE GAO received the B.S. degree in computer science and technology from Jilin University, Changchun, China, in 2018, where she is currently pursuing the M.S. degree. Her research interests include artificial intelligence optimization algorithms and knowledge graph construction domain.