International Journal of Innovative Computing, Information and Control
Volume **14**, Number **3**, June **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 881–897

FACTORED GREY WOLF OPTIMIZER WITH APPLICATION TO RESOURCE-CONSTRAINED PROJECT SCHEDULING

Weimin Xiao[1,2], Haojiang Deng[1], Yiqiang Sheng[1,*] and Linlin Hu[1]

[1]National Network New Media Engineering Research Center
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ xiaowm; denghj; hull }@dsp.ac.cn; *Corresponding author: shengyq@dsp.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

Received October 2017; revised February 2018

Abstract. *In this paper, we propose a Factored Grey Wolf Optimizer (FGWO) algorithm by adapting the convergence factor and adding weighted position factor to improve the existing Grey Wolf Optimizer (GWO). The value of convergence factor is nonlinearly changed during the iterations over the search process to better adapt the exploration and exploitation process, and the weighted position factor is linearly changed as the iteration increases to diverse the hierarchy of grey wolves. The FGWO algorithm is benchmarked on well-known test functions, and the results are verified by comparing FGWO against GWO, Particle Swarm Optimization (PSO), and Ant Lion Optimizer (ALO). This paper also presents an application of the proposed method to solve Resource-Constrained Project Scheduling Problem (RCPSP). The results show that the FGWO algorithm provides very competitive performances compared to the meta-heuristics. FGWO obtains 71.4% best solutions and 14.3% second best solutions of unimodal benchmark functions, 15.6% best solutions and 21.9% second best solutions of multimodal benchmark functions, as well as 50% best solutions of composite benchmark functions. The results of solving RCPSP also prove that FGWO can show high performance in solving RCPSP.*

Keywords: Factored grey wolf optimizer, Nonlinear convergence factor, Linear-weighted position factor, RCPSP

1. Introduction.

Over the last two decades, meta-heuristic optimization algorithms are very popular and have been used for obtaining optimal solutions in many scientific or engineering fields. Among these algorithms, the Bee Algorithm (BA) [1], bat algorithm [2, 3], Particle Swarm Optimization (PSO) [4, 5], and Genetic Algorithm (GA) [6] are fairly well-known. Meta-heuristic algorithms search in a search space for a global optimum by creating random solutions for a given problem. The random solutions, also called the set of candidate solutions, are improved during the iteration until satisfying the terminating condition. The iterative improvement process is considered to find a more accurate approximate value of the global optimum than the original random solutions. The mechanism makes meta-heuristics become prominent common and intrinsic advantages: simplicity, flexibility, derivation independency, and escaping from local minima. The search process of meta-heuristic optimization algorithm can be divided into two phases: exploration and exploitation [7, 8]. Exploration refers to searching and investigating the search space as widely as possible while exploitation refers to the local search ability to find the optimum solution around the obtained promising regions during exploration. It is challenging to strike a proper balance between exploration and exploitation.

The Grey Wolf Optimizer (GWO) [9] is a state-of-the-art Swarm Intelligence (SI) algorithm inspired by the social hierarchy and hunting for the prey behavior of grey wolf packs. However, the linear convergence factor to control the exploration and exploitation limits the performance of GWO as an algorithm is considered to search as broadly as possible during the exploration and converge as fast as possible in the exploitation. Besides, GWO algorithm does not consider the difference between wolves in the social hierarchy.

This work proposes a Factored Grey Wolf Optimizer (FGWO) algorithm to improve the performance of search process for the optimum solution. The "Factored" of FGWO means the convergence factor and weighted position factor. The convergence factor is adapted to achieve better balance between exploration and exploitation, and the weighted position factor can strengthen the social hierarchy of grey wolves. The main contributions of this work are described as follows:

- Adapting convergence factor nonlinearly to preferably balance the exploration and exploitation. The convergence factor in GWO is designed to balance the processes of exploration and exploitation, and decreases linearly during the iteration, which limits the performance of search process for optimum solution. The nonlinear dynamic convergence factor with exponential function can improve the exploitation process and convergence speed.
- Adding linear weighted position factor to enhance the social hierarchy of grey wolves. GWO assumes the first three ranks in the social hierarchy of grey wolves know better about the potential location of prey (optimum solution). So the other grey wolves should update their positions during each iteration based on the best three solutions obtained so far. In GWO the first three ranks of grey wolves have the same impact on position updating process, but it is a proper way to emphasize the difference.
- Testing FGWO with 29 benchmark functions and comparing against well-known meta-heuristics to evaluate the performance. The benchmark functions consist of unimodal, multimodal, fixed-dimension multimodal and composite functions. They are employed with different characteristics to benchmark the performance of algorithms from different perspectives, including exploitation ability, exploration ability, ability to escape from local minima, and convergence speed.
- Applying FGWO on Resource-Constrained Project Scheduling Problem (RCPSP) to evaluating the ability to solve engineering problem. RCPSP is a general engineering problem, and various kinds of meta-heuristics have been proposed to solve RCPSP. FGWO and compared well-known meta-heuristics were benchmarked with the standard dataset to evaluate the performance of solving RCPSP.

The rest of this paper is organized as follows. Section 2 presents GWO algorithm. Section 3 outlines the proposed FGWO algorithm. Experimental results and analysis of popular benchmark functions are provided in Section 4. Section 5 demonstrates the applicability of the proposed algorithm for solving RCPSP. A discussion of advantages and shortcomings of compared meta-heuristics is given in Section 6. Finally, Section 7 concludes the work and analyzes possible future studies.

2. **Grey Wolf Optimizer Algorithm.** The GWO algorithm was proposed by Mirjalili et al. in 2014 [9]. The main inspirations of GWO algorithm are the social relationship and hunting behavior of grey wolves. Grey wolves live in a pack mostly with a very strict social dominant hierarchy: The leaders called alphas are responsible for decision making. The second level of grey wolves called betas are the subordinate wolves to help the alphas making decision or other activities in the pack. The third-ranking in the hierarchy of grey wolves is delta. Delta wolves submit to alphas and betas and dominate the other ranking wolves. If a wolf is not in the rankings mentioned above, he/she is called omega. Omega

wolves play the role of scapegoat and submit to all the other dominant wolves. The group hunting of grey wolves includes tracking and approaching the prey, encircling the prey until it stops, attacking towards the prey.

The GWO algorithm mathematically models and simulates the social hierarchy and group hunting mechanism of grey wolves and performs optimization. The best solution is considered as the alpha (α) and the second and third optimal solutions are assumed to be beta (β) and delta (δ), respectively. The rest of the optional solutions are named omega (ω). The optimization (hunting) process in the GWO algorithm is led by alpha, beta and delta, and the omega wolves follow these three wolves to update their positions.

During the hunt, the grey wolves firstly encircle prey. The following equations are designed to mathematically model the encircling behavior:

$$D = |C \cdot X_p - X(t)| \tag{1}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{2}$$

where t is the current iteration, $X_p(t)$ indicates the position of the prey, and the position of the grey wolf is X. A and C indicate coefficient parameters and are calculated as follows:

$$A = 2a \cdot r_1 - a \tag{3}$$

$$C = 2 \cdot r_2 \tag{4}$$

where r_1 and r_2 are generated randomly in $[0, 1]$. a indicates the convergence factor linearly decreased from 2 to 0 during the iterations. The parameter a is calculated as follows:

$$a = 2 - t \cdot 2/t_{\max} \tag{5}$$

where t is the current iteration and $t_{\max}$ indicates the total number of iteration.

When hunting the prey, the wolves are guided by the alpha with beta and delta participating occasionally. To simulate the hunting behavior in GWO algorithm mathematically, the alpha (the best optional solution), beta and delta are assumed to know better about the potential location of prey. Hence during each iteration, the best three solutions obtained so far are saved and force other search agents to update positions in accordance with the best solution. The position of search agents is updated as follows:

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{6}$$

where X_1, X_2, X_3 are calculated as follows respectively:

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \tag{7}$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \tag{8}$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \tag{9}$$

where X_α, X_β, X_δ indicate the best three solutions at a certain iteration t, and D_α, D_β, D_δ are determined as following equations respectively:

$$D_\alpha = |C_1 \cdot X_\alpha - X| \tag{10}$$

$$D_\beta = |C_2 \cdot X_\beta - X| \tag{11}$$

$$D_\delta = |C_3 \cdot X_\delta - X| \tag{12}$$

3. **Factored Grey Wolf Optimizer Algorithm (FGWO).** In GWO, the convergence factor a to balance exploration and exploitation is linearly decreased from 2 to 0, which limits the performance of exploitation ability and convergence speed. Besides, according to the principle of social hierarchy in grey wolves, GWO updates the positions of wolves with the mean value of alpha, beta and delta during each iteration, while it is more reasonable to enhance the different impacts. In this session the proposed FGWO algorithm is discussed, the improvements of GWO algorithm mainly focus on dynamically adapting the convergence factor with an exponential function and position factor with linear weight.

3.1. **Nonlinear convergence factor.** The swarm intelligence optimization algorithm has the process of exploration and exploitation during the search process. In GWO algorithm the search process firstly creates a random population of candidate solutions. The alpha, beta, and delta estimate the probable position of prey and each candidate solution updates the distance from the prey. Generally speaking, a swarm intelligence algorithm with good performance should be able to search the search space effectively in the initial iterations to find out optimal solutions as most as possible, while over the course of iterations the exploitation is more considerable to perform better accuracy and convergence behavior.

In GWO algorithm, Equation (3) emphasizes the exploration and exploitation respectively during the optimization. When $|A| > 1$ the candidate solutions tend to diverge from the prey for exploration while $|A| < 1$ converge towards the prey for exploitation. The value of $|A|$ is decided by the parameter a decreased from 2 to 0 linearly. In the proposed FGWO algorithm, a nonlinear dynamic convergence factor with exponential function is adapted to improve the exploration and exploitation. The value of a is calculated as follows:

$$a(t) = \rho - \exp\left(-\mu \cdot (t_{\max} - t)/t_{\max}\right)^3 \tag{13}$$

where t is the current iteration and $t_{\max}$ indicates the total number of iteration. ρ indicates the initial value of convergence factor, and in this paper the value of ρ is 1.5. μ is the adjustment coefficient, and in this paper the value of μ is 25. The settings of the values ρ and μ are justified in the following experiments in Subsection 4.5.

3.2. **Linear weighted position factor.** In GWO algorithm the best three candidate solutions are considered having better knowledge of the potential location of the prey, and oblige other search agents to update positions (by Equations (6)) according to the position of alpha, beta, and delta. The equation considers the alpha, beta, and delta having the same impact on the other search agents, while in GWO algorithm the alpha is the fittest solution and considered to know best of the potential position of the prey, and beta is fitter than delta as well. In this way, it is more reasonable to reflect the difference among alpha, beta and delta when updating positions of other search agents. A linear weight factor of position is added to emphasize the difference mentioned above. ω_1, ω_2 and ω_3 are the weight factors of alpha, beta and delta respectively, and calculated as follows:

$$\omega_1 = 3 - t/t_{\max} \tag{14}$$

$$\omega_2 = 2 - t/t_{\max} \tag{15}$$

$$\omega_3 = 1 - t/t_{\max} \tag{16}$$

where t is the current iteration and $t_{\max}$ indicates the total number of iteration.

The positions of other search agents are calculated as follows:

$$X(t+1) = \left(\omega_1 \cdot X_1 + \omega_2 \cdot X_2 + \omega_3 \cdot X_3\right) / \left(\omega_1 + \omega_2 + \omega_3\right) \tag{17}$$

4. **Results on Benchmark Experiment.** In this session the proposed FGWO algorithm is benchmarked on 29 test functions [10, 11]. The benchmark functions used are employed with different characteristics to benchmark the performance of the FGWO algorithm from different perspectives, and are divided into four groups as listed in Tables 1, 2, 3 and 4: unimodal, multimodal, fixed-dimension multimodal, and composite functions. As shown in Tables 1, 2, 3 and 4, Dim is dimension of the function, Range indicates the boundary of the search space, and $f_{\min}$ indicates the optimum.

To verify the results, the FGWO algorithm is compared to basic GWO algorithm and well-known or recent algorithms: Particle Swarm Optimization (PSO) [12] as the best swarm-based algorithm, and Ant Lion Optimizer (ALO) [13] algorithm. Each algorithm is run 30 times on each test function and the collected statistical results (average and

Table 1. Unimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	30	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0

Table 2. Multimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]$	-418.9829×5
$F_9(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0
$y_i = 1 + \frac{x_i + 1}{4}$			
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, & x_i < -a, \end{cases}$			
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

TABLE 3. Fixed-dimension multimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	$[-65, 65]$	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5, 5]$	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	$[1, 3]$	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
$F_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.5363

standard deviation of the best solution) are reported in Tables 5, 7, 9 and 11. For unimodal, multimodal and fixed-dimension multimodal functions, each of the benchmark functions is solved using 30 search agents over 1000 iterations. For composite functions, each benchmark function is solved using 30 search agents over 100 iterations. The average and standard deviation show which algorithm behaves more stable. Besides, p-value of Wilcoxon rank-sum test [14] is conducted to determine the significance level of two algorithms in Tables 6, 8, 10 and 12. If a p-value is less than 0.05, it shows that the difference between two compared algorithms is statistically significant.

4.1. **Results of FGWO on unimodal test functions.** According to the results on the unimodal benchmark functions of the algorithms in Table 5, it is obvious that the proposed FGWO algorithm is able to provide very competitive results; this algorithm outperforms other algorithms on most of the benchmark cases including F1, F2, F3, F4, and F7. Besides, the p-values in Table 6 are much less than 0.05, thus certify that it is statistically significant of the superiority. Considering the unimodal functions are suitable for benchmarking exploitation, it can be stated the superior performance of FGWO algorithm in terms of high exploitation. High exploitation helps the FGWO algorithm to exploit the optimum accurately and converge towards it rapidly.

4.2. **Results of FGWO on multimodal test functions.** Tables 7 and 9 show the results of the algorithm running on multimodal benchmark functions. Table 7 shows that FGWO algorithm outperforms other algorithms on F10, and provides the second best results on F9, F11. The p-values presented in Tables 8 and 10 also certify that the FGWO algorithm performs significantly better results. In contrast to unimodal functions, the multimodal functions are suitable for benchmarking the exploration ability, and the results show that the FGWO algorithm has competitive performance in terms of exploration.

TABLE 4. Composite benchmark functions

Function	Dim	Range	$f_{\min}$
$F_{24}(CF1)$ $f_1, f_2, f_3, \ldots, f_{10}$ = Sphere Function $[\sigma_1, \sigma_1, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_1, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	30	$[-5, 5]$	0
$F_{25}(CF2)$ $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1, \sigma_1, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_1, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	30	$[-5, 5]$	0
$F_{26}(CF3)$ $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1, \sigma_1, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_1, \lambda_3, \ldots, \lambda_{10}] = [1, 1, 1, \ldots, 1]$	30	$[-5, 5]$	0
$F_{27}(CF4)$ f_1, f_2 = Ackley's Function, f_3, f_4 = Rastrigin's Function, f_5, f_6 = Weierstrass Function, f_7, f_8 = Griewank's Function, f_9, f_{10} = Sphere Function $[\sigma_1, \sigma_1, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_1, \lambda_3, \ldots, \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	30	$[-5, 5]$	0
$F_{28}(CF5)$ f_1, f_2 = Rastrigin's Function, f_3, f_4 = Weierstrass Function, f_5, f_6 = Griewank's Function, f_7, f_8 = Ackley's Function, f_9, f_{10} = Sphere Function $[\sigma_1, \sigma_1, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_1, \lambda_3, \ldots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	30	$[-5, 5]$	0
$F_{29}(CF6)$ f_1, f_2 = Rastrigin's Function, f_3, f_4 = Weierstrass Function, f_5, f_6 = Griewank's Function, f_7, f_8 = Ackley's Function, f_9, f_{10} = Sphere Function $[\sigma_1, \sigma_1, \sigma_3, \ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_1, \lambda_3, \ldots, \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	30	$[-5, 5]$	0

4.3. **Results of FGWO on composite test functions.** The results of the algorithm running on composite benchmark functions in Table 11 show that FGWO algorithm outperforms other algorithms in terms of standard deviation on 5 test functions, and provides the best average solution on F29. The p-values presented in Table 12 also certify that the

Table 5. Results of unimodal benchmark functions

F	Type	FGWO	GWO	PSO	ALO
F_1	Ave	**5.7e-122**	4.74e-59	8.05e-09	1.10e-05
	Std	**1.9e-121**	8.33e-59	1.81e-08	9.87e-06
F_2	Ave	**3.9e-72**	8.79e-35	0.00050	24.631
	Std	**6.86e-72**	8.96e-35	0.00094	38.4351
F_3	Ave	**9.1e-31**	2.43e-15	16.8676	939.593
	Std	**4.92e-30**	8.02e-15	9.1336	471.491
F_4	Ave	**1.7e-32**	1.82e-14	0.63922	11.7768
	Std	**5.92e-32**	4.06e-14	0.19912	4.2592
F_5	Ave	27.4121	**26.9333**	59.7079	213.966
	Std	0.84336	**0.61989**	36.6992	343.899
F_6	Ave	1.8377	0.61624	**1.26e-08**	9.14e-06
	Std	0.47875	0.31855	**3.41e-08**	5.25e-06
F_7	Ave	**0.00078**	0.00088	0.07216	0.09940
	Std	**0.00040**	0.00056	0.02025	0.02796

Table 6. p-values of the Wilcoxon ranksum test over unimodal benchmark functions

F	FGWO	GWO	PSO	ALO
F_1	N/A	3.0199e-11	3.0199e-11	3.0199e-11
F_2	N/A	3.0199e-11	3.0199e-11	3.0199e-11
F_3	N/A	3.0199e-11	3.0199e-11	3.0199e-11
F_4	N/A	3.0199e-11	3.0199e-11	3.0199e-11
F_5	N/A	0.0030339	0.026077	9.5139e-06
F_6	N/A	9.9186e-11	3.0199e-11	3.0199e-11
F_7	N/A	0.59969	3.0199e-11	3.0199e-11

Table 7. Results of multimodal benchmark functions

F	Type	FGWO	GWO	PSO	ALO
F_8	Ave	-3713.11	-6007.66	**-6168.29**	-5699.08
	Std	**447.459**	909.590	1490.22	1298.96
F_9	Ave	10.528	**0.6963**	50.592	78.3694
	Std	26.3122	**1.6158**	13.4431	21.8135
F_{10}	Ave	**7.99e-15**	1.58e-14	0.055096	2.2665
	Std	**0**	3.01e-15	0.30052	0.64592
F_{11}	Ave	0.00165	**0.00088**	0.01157	0.01351
	Std	0.00442	**0.00270**	0.01250	0.01068
F_{12}	Ave	0.15839	0.03255	**0.00346**	10.5394
	Std	0.12165	**0.01200**	0.01893	4.5593
F_{13}	Ave	1.329	0.49442	**0.003265**	0.5874
	Std	0.22101	0.19481	**0.005776**	2.6956

FGWO algorithm performs significantly better results on F24 and F26. Composite functions are very challenging test beds for meta-heuristics. Since composite functions test the exploration and exploitation combined, they are suitable for benchmarking exploration and exploitation simultaneously. Moreover, due to the massive number of local optima

Table 8. p-values of the Wilcoxon ranksum test over multi-modal benchmark functions

F	**FGWO**	**GWO**	**PSO**	**ALO**
F_8	N/A	2.8716e-10	1.8731e-07	1.6179e-11
F_9	N/A	0.56012	3.3715e-09	1.7557e-09
F_{10}	N/A	1.5139e-12	1.2118e-12	1.2118e-12
F_{11}	N/A	0.64259	5.2776e-09	1.8801e-09
F_{12}	N/A	3.3384e-11	6.6955e-11	3.0199e-11
F_{13}	N/A	4.5043e-11	3.0199e-11	8.891e-10

Table 9. Results of fixed-dimension multimodal benchmark functions

F	**Type**	**FGWO**	**GWO**	**PSO**	**ALO**
F_{14}	Ave	9.1205	3.7117	3.3276	**1.4611**
	Std	4.6213	3.6931	2.9007	**0.81189**
F_{15}	Ave	0.00245	0.00366	**0.00075**	0.00486
	Std	0.00608	0.00760	**0.00026**	0.01241
F_{16}	Ave	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**
	Std	2.21e-06	5.1e-09	**6.71e-16**	4.52e-14
F_{17}	Ave	0.39806	0.39792	**0.39789**	**0.39789**
	Std	0.00022	0.00013	**0**	4.83e-14
F_{18}	Ave	8.4001	**3**	**3**	**3**
	Std	20.5505	1.13e-05	**9.37e-16**	5.09e-13
F_{19}	Ave	−3.8622	**−3.8616**	−3.8628	−3.8628
	Std	0.00034	0.00239	**2.71e-15**	1.92e-14
F_{20}	Ave	−3.2584	−3.2505	**−3.2784**	−3.2625
	Std	**0.05351**	0.07030	0.05827	0.06047
F_{21}	Ave	−6.7933	**−9.646**	−7.3909	−7.7817
	Std	2.4166	**1.5462**	3.1091	2.5786
F_{22}	Ave	−7.9817	**−10.2253**	−9.365	−6.6613
	Std	1.5923	**0.97034**	2.4084	3.2301
F_{23}	Ave	−8.2357	**−10.3557**	−9.6787	−7.3523
	Std	1.7443	**0.98727**	2.2604	3.5518

Table 10. p-values of the Wilcoxon ranksum test over fixed-dimension multimodal benchmark functions

F	**FGWO**	**GWO**	**PSO**	**ALO**
F_{14}	N/A	1.107e-06	1.4166e-06	1.4854e-09
F_{15}	N/A	2.5974e-05	0.0033386	1.7479e-05
F_{16}	N/A	3.0199e-11	1.7203e-12	3.018e-11
F_{17}	N/A	4.1825e-09	1.2118e-12	3.0066e-11
F_{18}	N/A	0.46427	9.3482e-12	3.018e-11
F_{19}	N/A	0.0010035	1.2118e-12	3.0161e-11
F_{20}	N/A	0.14945	0.00028918	0.015014
F_{21}	N/A	8.4848e-09	0.075743	0.012212
F_{22}	N/A	4.1997e-10	3.3438e-06	0.42896
F_{23}	N/A	3.8202e-10	4.1727e-07	0.44642

TABLE 11. Results of composite benchmark functions

F	**Type**	**FGWO**	**GWO**	**PSO**	**ALO**
F_{24}	Ave	190.804	**81.6855**	90.1232	129.877
	Std	**38.1956**	81.8135	95.9132	96.8169
F_{25}	Ave	178.786	173.429	**158.595**	175.207
	Std	**66.2705**	91.0175	109.196	93.8774
F_{26}	Ave	513.137	280.679	259.138	380.618
	Std	**47.9687**	114.429	94.6106	190.806
F_{27}	Ave	507.276	450.472	**441.440**	522.280
	Std	**32.3087**	134.817	146.365	139.169
F_{28}	Ave	171.420	141.748	**105.196**	187.603
	Std	**73.498**	150.825	135.713	134.009
F_{29}	Ave	**716.851**	862.667	780.187	829.026
	Std	178.785	**117.160**	184.985	153.355

TABLE 12. p-values of the Wilcoxon ranksum test over composite benchmark functions

F	**FGWO**	**GWO**	**PSO**	**ALO**
F_{24}	N/A	3.6459e-08	4.084e-05	0.00090307
F_{25}	N/A	0.97052	0.20095	0.56922
F_{26}	N/A	1.8567e-09	5.5727e-10	1.0188e-05
F_{27}	N/A	0.028129	0.0010035	0.56922
F_{28}	N/A	0.14532	0.018368	0.18577
F_{29}	N/A	0.49178	0.42039	0.11882

in such functions, the ability to escape from local minima can be examined. The results show that the FGWO algorithm has competitive performance in terms of exploration, ability to escape from local minima and exploitation simultaneously.

4.4. **Analysis of FGWO algorithm.** In this subsection, four new metrics are employed to further observe the performance of the proposed FGWO algorithm. The quantitative metrics are shown as follows:

- Search history – the position of grey wolves from the first to the final iteration;
- Trajectory – the position of grey wolves of the first dimension from the first to the final iteration;
- Fitness history – the average fitness of grey wolves from the first to the final iteration;
- Convergence – the fitness of alpha from the first to the final iteration.

Some of the benchmark functions are chosen and solved by 30 search agents over 200 iterations. The results are shown in Figure 1. The search history of position during optimization, as depicted in the second column in Figure 1, shows that the FGWO algorithm tends to search the promising regions and exploit the optimum of the search space. The fourth column in Figure 1 depicts the position of the first dimension of the first grey wolf over 200 iterations. It can be seen abrupt changes during initial iterations while over the course of iterations they decrease gradually. It can guarantee that the algorithm eventually converges to a point of the search space. In addition, the last two columns of Figure 1 illustrate the average fitness history of all grey wolves and the convergence curve which represents the fitness of the alpha, respectively. It can be observed that there are similar descending behaviors of average fitness and convergence curves, which demonstrate that

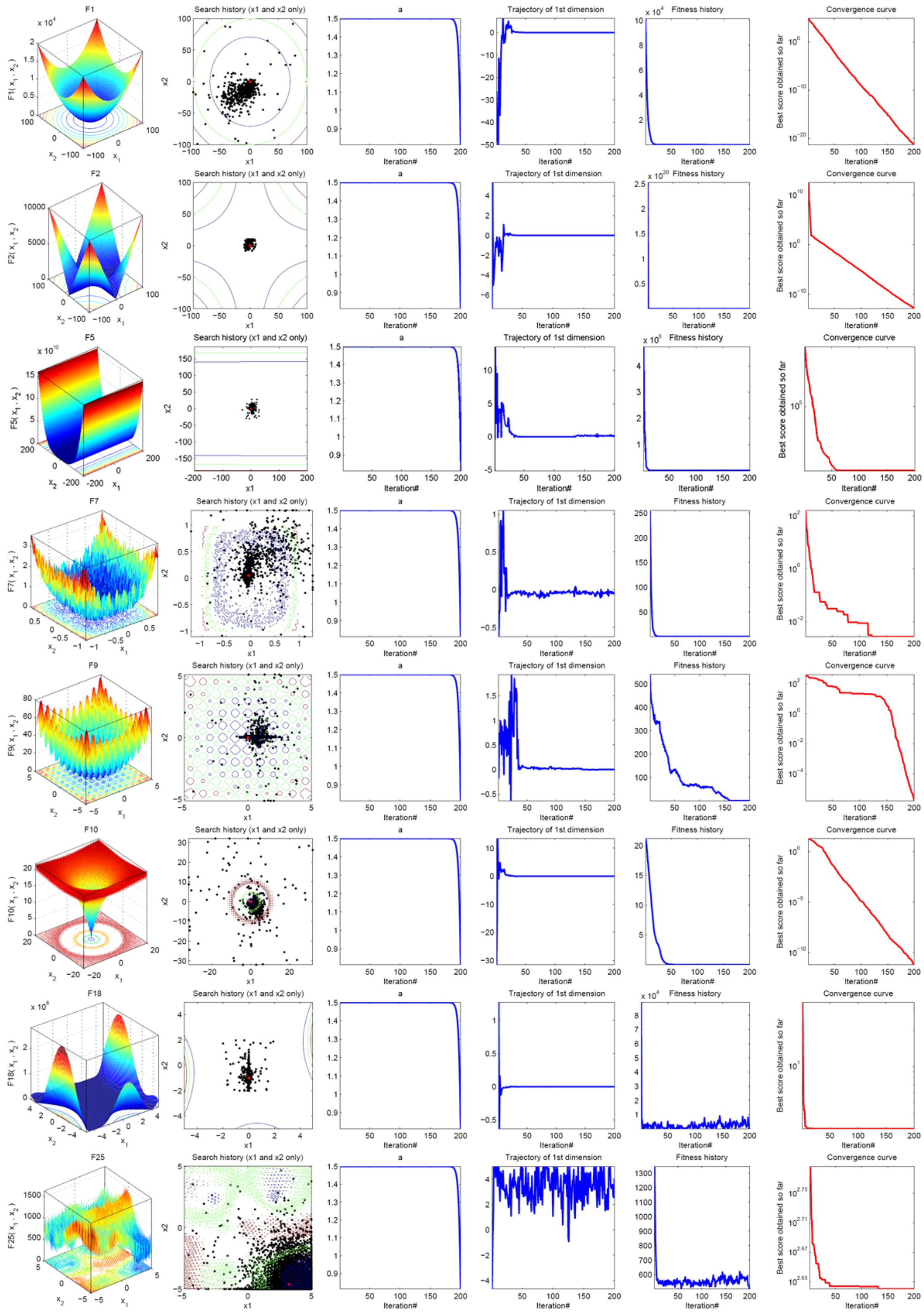

FIGURE 1. Search history, trajectory in the first dimension, average fitness of all grey wolves, and convergence rate

the proposed FGWO algorithm can improve the overall fitness of the initial population. It can also be seen the accelerated trend in convergence curve and the approximation of the global optimum as iteration increases. This is due to the emphasis on exploitation by adjusting the convergence factor and enhancing the weight factor of alpha over iteration, which accelerate the convergence towards the optimum.

4.5. **Experiments on the settings of values ρ and μ.** In this subsection, the settings of values ρ and μ in Equation (13) are evaluated. Considering there are two types of values to be evaluated in an equation, one of the values is changed while the other one stays the same in the experiments to verify the impact on the search process. In the experiments on the settings of value ρ, ρ is set to be 0.5, 1.0, 1.5, 2.0, 2.5 respectively, and μ is set to be 25. In the experiments on the settings of value μ, μ is set to be 15, 20, 25, 30, 35 respectively, and ρ is set to be 1.5. Each setting is benchmarked 10 times on the test functions with 30 search agents over 100 iterations, and some of the results are shown in Figure 2 and Figure 3. The results show that FGWO performs better when ρ is set 1.5 and μ is set 25.

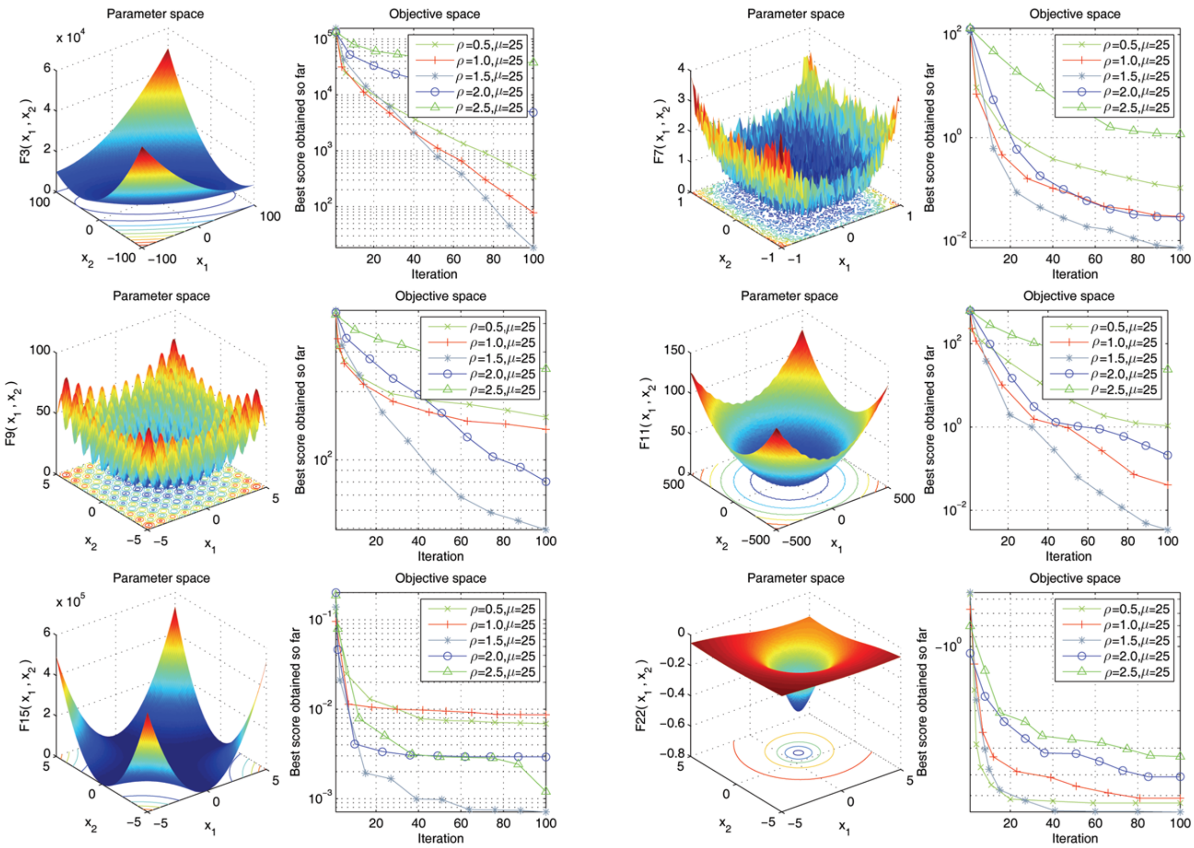

FIGURE 2. Experiments on the settings of value ρ ($\rho = 0.5$, 1.0, 1.5, 2.0, 2.5, $\mu = 25$, on $F3$, $F7$, $F9$, $F11$, $F15$, $F22$)

5. **Application of FGWO on RCPSP.** Resource-Constrained Project Scheduling Problem (RCPSP) is a general problem containing resources of limited availability, activities of certain resource requests and durations. RCPSP is confined to meet the resources and durations constraints and achieves a certain objective. Since RCPSP is an NP-hard problem, various kinds of meta-heuristics have been proposed to search for optimal solutions

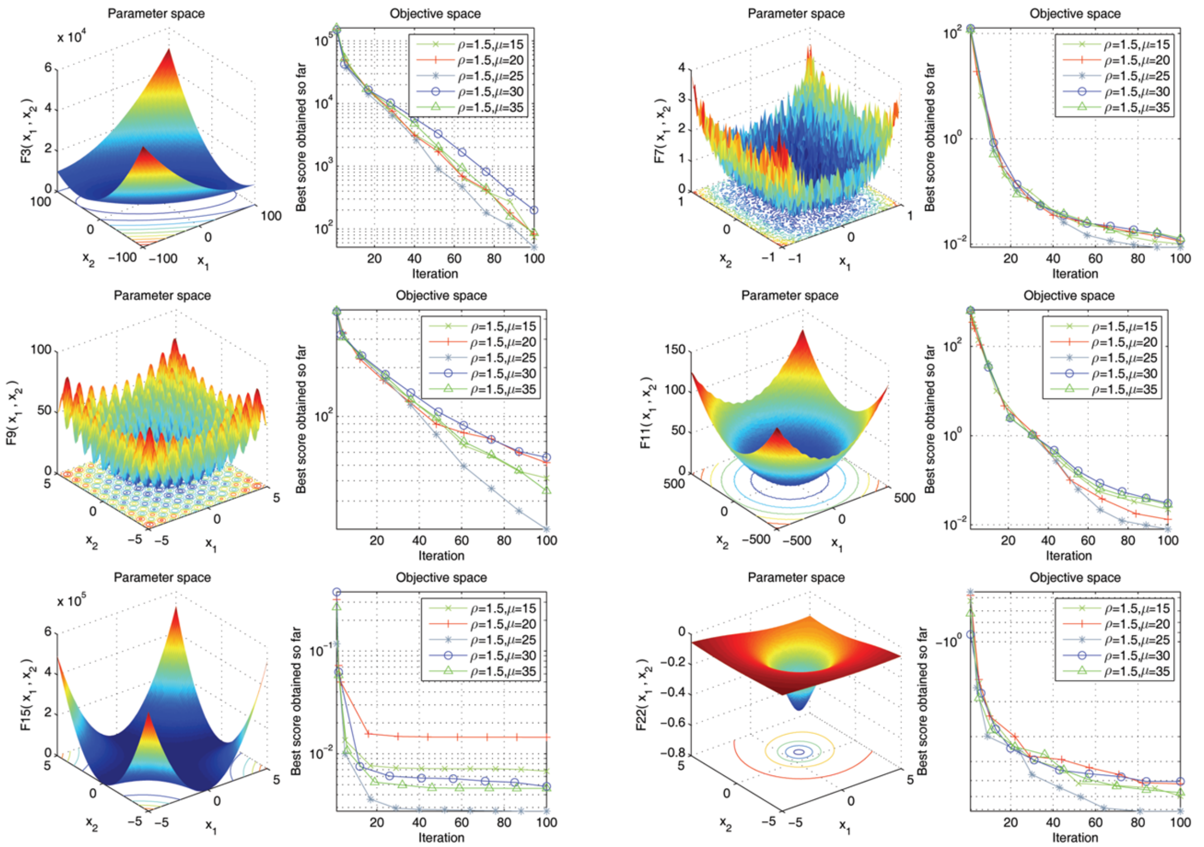

Figure 3. Experiments on the settings of value μ ($\mu = 15, 20, 25, 30, 35$, $\rho = 1.5$, on $F3$, $F7$, $F9$, $F11$, $F15$, $F22$)

[15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Some of the most popular algorithms are Genetic Algorithm (GA) [21], Particle Swarm Optimization (PSO) [18, 22, 23, 24], Cuckoo Search (CS) [26], Ant Colony Optimization (ACO) [19], and Simulated Annealing (SA) [25]. These meta-heuristics are benchmarked with the standard dataset of the Project Scheduling Problem Library (PSPLIB) [27].

The typical RCPSP studied in this paper is described as follows: (1) Finding the minimal makespan schedule is the objective; (2) The number of activities is $N+2$, and each activity i has processing duration d_i ($i = 0, \ldots, N+1$). Activities are non-preemptive during management. The activity 0 and $N+1$ are pseudo activities which indicate the start and finish of schedule, respectively. (3) Activities have precedence constraint. For example, if P_i is the set of immediate predecessors of activity i, activity i is not allowed to start until the finish of all the immediate predecessors P_i. (4) There are renewable resources in various types with constant amount at each time or period. Let R be the available amount of resource Q. Each activity i requires resource r_{ik} unit of resource k ($k = 1, \ldots, q$) per unit of activities' execution time d_i. An activity-on-node network of RCPSP example is shown in Figure 4, and the parameters (d_i, r_{ik}) of activity i are illustrated to indicate the activity duration and required amounts of various types of resources.

Considering the above description of RCPSP and the activity-no-node representation example, RCPSP can be formulated as follows:

$$\min\{\max f_i | i = 1, 2, \ldots, N\} \tag{18}$$

subject to:

$$f_j - f_i \geq d_i \quad \forall j \in P_i;\ i = 1, 2, \ldots, N \tag{19}$$

$$\sum\nolimits_{A_t} r_{ik} \leq R_k, \quad k = 1, 2, \ldots, K;\ t = s_1, s_2, \ldots, s_N \tag{20}$$

where N represents the number of activities involved in a project, f_i is the finish time of activity i, d_i indicates the duration time of activity i, and P_i is the set of preceding activities. R_k represents available amount of resource k, and r_{ik} is the amount of resource k required by activity i. A_t indicates the set of ongoing activities at t and s_i is the start time of activity i. Equation (18) represents the objective, Equation (19) and Equation (20) represent precedence constraints and resource constraints, respectively.

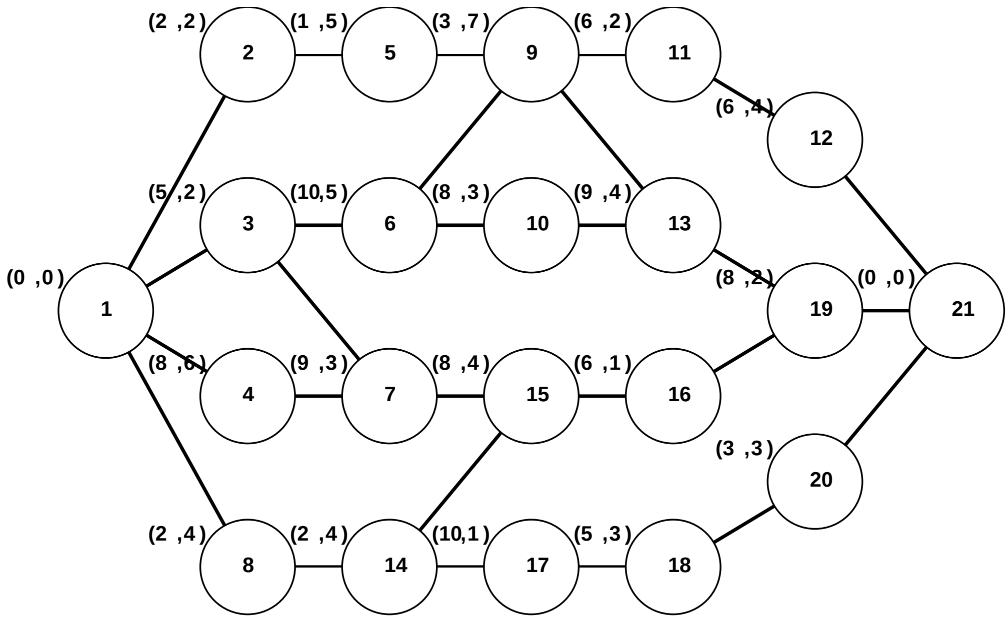

Figure 4. Example of RCPSP activity graph

In order to investigate the FGWO algorithm for RCPSP, the typical project example shown in Figure 4 is analyzed, and PSPLIB standard datasets are used as benchmark. Figure 5 shows the solutions achieved using 4 algorithms (FGWO, GWO, PSO and ALO) to solve the project example above. The corresponding optimal schedules described sequences and start times (or finish times) of the activities (dummy activities not included), as well as the allocation profiles of the resources required. The schedule with smaller latest finish time is the better. The results show that FGWO can perform scheduling the same as GWO, PSO and ALO in duration time.

For comparing the performances of FGWO to other algorithms on RCPSP, the instances in the well-known PSPLIB are used as benchmark and simulated. The standard dataset of PSPLIB contains subset J30, J60 and J120, which include 30, 60 and 120 activities respectively. In PSPLIB benchmark, the number of instances for J30 and J60 is 480, respectively, and that for J120 is 600. Hence, there are total of 1560 instances. For each instance, FGWO and comparing algorithms were run independently with 30 search agents over 200 iterations. Statistical results (optimal found and average deviation) are collected and reported in Table 13. FGWO was compared with GWO, PSO and ALO.

For all dataset, FGWO performs best compared to other algorithms in optimal found and average deviation. The results show that the FGWO is better than other algorithms for solving RCPSP with different scales.

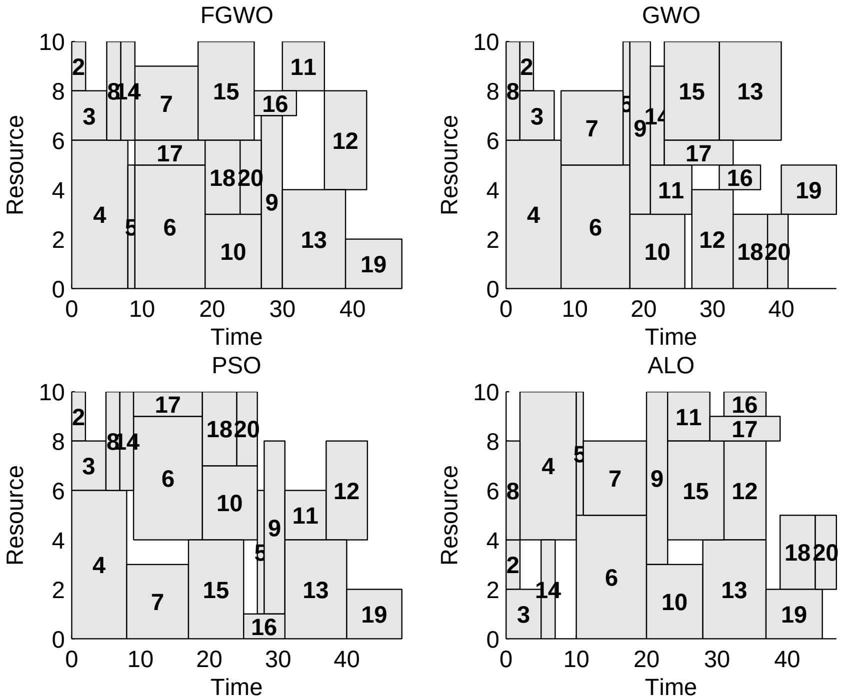

FIGURE 5. Solutions by 4 algorithms (FGWO, GWO, PSO and ALO)

TABLE 13. Results of PSPLIB standard dataset

Dataset	Statistic	FGWO	GWO	PSO	ALO
J30	Opt.Found	341	335	328	331
	AvDev. (%)	0.433	0.5636	0.6417	0.6039
J60	Opt.Found	202	200	196	197
	AvDev. (%)	13.1466	13.4039	14.4017	14.2636
J120	Opt.Found	119	114	118	116
	AvDev. (%)	35.1466	36.636	36.5756	36.6017

TABLE 14. Discussion of advantages and shortcomings (High > Medium > Low)

	FGWO	GWO	PSO	ALO
Exploitation ability	High	Medium	Medium	Low
Exploration ability	Medium	High	High	Medium
Ability to escape from local minima	Medium	Low	High	Low
Convergence speed	High	Medium	Medium	Low

6. **Discussion.** In this section the advantages and shortcomings of compared algorithms are discussed and shown in Table 14. Firstly, the results on unimodal benchmark functions showed the superior exploitation of FGWO. Secondly, the exploration ability of FGWO was tested by the multimodal functions, and the results showed FGWO outperformed ALO, but was not better than GWO and PSO. Thirdly, the results of the composite functions showed high ability to escape from local minima, but was not better than PSO. Finally, the convergence analysis showed the superior convergence ability of FGWO.

7. **Conclusions.** This work proposed a factored grey wolf optimization algorithm called FGWO. A nonlinear convergence factor was adapted and linear weighted position factor was added to improve the optimization. The performance of FGWO algorithm was tested on 29 benchmark functions. The FGWO algorithm was compared to three well-known and recent algorithms: GWO, PSO, and ALO. The p-values of Wilcoxon statistical tests were also conducted to compare the algorithms. The FGWO algorithm was finally applied to RCPSP. The results showed that FGWO provides highly competitive performances and outperforms other algorithms in most of the benchmark functions and engineering problems. For future work, we intend to research on better performance of the ability to escape from local minima.

Acknowledgment. This work is partially supported by the Special Fund for National High-Tech R&D Program (863 Program) under Grant NO. 2015AA015802. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] W. A. Hussein, S. Sahran and S. N. H. S. Abdullah, The variants of the bees algorithm (BA): A survey, *Artificial Intelligence Review*, vol.47, no.1, pp.67-121, 2017.
[2] J. Perez, F. Valdez, O. Castillo and O. Roeva, Bat algorithm with parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions, *IEEE the 8th International Conference on Intelligent Systems (IS)*, pp.120-127, 2016.
[3] X. Yang and A. H. Gandomi, Bat algorithm: A novel approach for global engineering optimization, *Engineering Computations*, vol.29, no.5, pp.464-483, 2012.
[4] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE International Symposium on Neural Networks*, vol.4, pp.1942-1948, 1995.
[5] A. Ghodousian and M. R. Parvari, A modified PSO algorithm for linear optimization problem subject to the generalized fuzzy relational inequalities with fuzzy constraints (FRI-FC), *Information Sciences*, vols.418-419, pp.317-345, 2017.
[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1989.
[7] O. Olorunda and A. P. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, *Evolutionary Computation*, pp.1128-1134, 2008.
[8] L. Lin and M. Gen, Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation, *Soft Computing*, vol.13, no.2, pp.157-168, 2009.
[9] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, vol.69, no.3, pp.46-61, 2014.
[10] M. Molga and C. Smutnicki, *Test Functions for Optimization Needs*, 2005.
[11] X. S. Yang, Test problems in optimization, *Mathematics*, vol.2, no.2, pp.63-86, 2010.
[12] Y. Shi and R. Eberhart, A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, pp.69-73, 1998.
[13] S. Mirjalili, The ant lion optimizer, *Advances in Engineering Software*, vol.83, pp.80-98, 2015.
[14] J. Derrac, S. García, D. Molina and F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm & Evolutionary Computation*, vol.1, no.1, pp.3-18, 2011.
[15] A. Gonzalez-Pardo, J. Ser and D. Camacho, Comparative study of pheromone control heuristics in ACO algorithms for solving RCPSP problems, *Applied Soft Computing*, vol.60, pp.241-255, 2017.
[16] J. Xiao, Z. Wu, X. Hong, J. Tang and Y. Tang, Integration of electromagnetism with multi-objective evolutionary algorithms for RCPSP, *European Journal of Operational Research*, vol.251, no.1, pp.22-35, 2016.
[17] M. Vanhoucke and J. Coelho, An approach using SAT solvers for the RCPSP with logical constraints, *European Journal of Operational Research*, vol.249, no.2, pp.577-591, 2016.
[18] Q. Jia and Y. Seo, An improved particle swarm optimization for the resource-constrained project scheduling problem, *The International Journal of Advanced Manufacturing Technology*, vol.67, no.9, pp.2627-2638, 2013.

[19] J. Yan, C. Zhao and H. Dong, Resource constrained project scheduling problem based on improved ant colony algorithm, *The 26th Chinese Control and Decision Conference (2014 CCDC)*, pp.1560-1563, 2014.

[20] K. S. Hindi, H. Yang and K. Fleszar, An evolutionary algorithm for resource-constrained project scheduling, *IEEE Trans. Evolutionary Computation*, vol.6, no.5, pp.512-518, 2002.

[21] H. Zhang, H. Xu and W. Peng, A genetic algorithm for solving RCPSP, *International Symposium on Computer Science and Computational Technology*, pp.246-249, 2008.

[22] J. Joy, S. Rajeev and V. Narayanan, Particle swarm optimization for resource constrained-project scheduling problem with varying resource levels, *Procedia Technology*, vol.25, pp.948-954, 2016.

[23] R. M. Chen, Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem, *Expert Systems with Applications*, vol.38, no.6, pp.7102-7111, 2011.

[24] C. Meng, A new particle swarm optimization with multi-speed updating formulas, *Journal of Network New Media*, 2013.

[25] P. P. Das and S. Acharyya, Simulated annealing variants for solving resource constrained project scheduling problem: A comparative study, *International Conference on Computer and Information Technology*, pp.469-474, 2011.

[26] K. Bibiks, F. Hu, J. P. Li and A. Smith, Discrete cuckoo search for resource constrained project scheduling problem, *IEEE International Conference on Computational Science and Engineering*, pp.240-245, 2015.

[27] R. Kolisch and A. Sprecher, PSPLIB – A project scheduling problem library: OR software – ORSEP operations research software exchange program, *European Journal of Operational Research*, vol.96, no.1, pp.205-217, 1997.