A Novel Hybrid GWO-FPA Algorithm for Optimization Applications

Jeng-Shyang Pan[1], Thi-Kien Dao[3], Shu-Chuan Chu[2], and Trong-The Nguyen[3](✉)

[1] Fujian Provincial Key Laboratory of Big Data Mining and Applications, College of Information Science and Engineering, Fujian University of Technology, Fuzhou, China
jengshyangpan@fjut.edu.cn

[2] School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, Australia
jan.chu@flinders.edu.au

[3] Department of Information Technology, Haiphong Private University, Haiphong, Vietnam
jvnkien@gmail.com, vnthe@hpu.edu.vn

Abstract. The recent trend of research is to hybridize two or several numbers of variants to find out the better quality of solution in practical optimization applications. In this paper, a new approach hybrid Grey Wolf Optimizer (GWO)-Flower Pollination Algorithm (FPA) is proposed based on the combination of exploitation phase in GWO and exploration stage in FPA. The hybrid proposed GWOFPA improves movement directions and speed of the grey wolves in updating positions of FPA. The simulation uses six benchmark tests for evaluating the performance of the proposed method. Compared other metaheuristics such as Particle Swarm Optimization (PSO), FPA, and GWO, the simulation results demonstrate that the proposed approach offers the better performance in solving optimization problems with or without unknown search areas.

Keywords: Optimization · Hybrid GWO-FPA algorithm · Grey Wolf Optimizer · Flower Pollination Algorithm

1 Introduction

Highly efficient technique in searching the best possible results in the benchmark and practical applications is the global optimization method [1]. In optimization, only a few results are compared the best one which is known as the goal. Classical optimization approaches have some deficiencies of finding the optimal global solutions of optimization problems [2]. These shortcomings are primarily interdependent on their original search systems. These conventional algorithms are strongly under effects of choosing proper types of variables, objectives and constraints functions. They also do not grant a universal solution method that can be applied to find the global optimal solution of the duties were several types of constrained functions, variables, and objective are used [3]. For covering these deficiencies, a new technique with the name

J.-S. Pan et al. (eds.), *Advances in Smart Vehicular Technology, Transportation, Communication and Applications*, Smart Innovation, Systems and Technologies 86,
https://doi.org/10.1007/978-3-319-70730-3_33

of metaheuristics was originated, which is mainly developed from artificial intelligence research that originated by scientists or researchers. Nature-inspired methods are developed for solving the several types of hard global optimization functions without having to the full accommodate to each function.

Recently, scientists and scholars have developed several numbers of metaheuristics in order to find the best global optimal solution of benchmark and real-life applications. The nature-inspired techniques have been originated, some of them are Particle Swarm Optimization (PSO) [4], Genetic Algorithm (GA) [5], Grey Wolf Optimization (GWO) [6], Flower Pollination Algorithm (FPA) [7], Bat Algorithm (BA) [8]. Moreover, in the case of the hybrid convergence, nature-inspired algorithm hybridizations using batch modeling are combinations amid evolutionary techniques and methods of neighborhood or course [9].

This paper introduces a new hybrid model combining Grey Wolf Optimizer (GWO) and Flower Pollination Algorithm (FPA) named GWOFPA. The proposed algorithm comprises of best characteristics of both GWO and FPA. The performance of the proposed variant is tested on six standard benchmarks. The solutions are compared relying on the metaheuristics reported in the review of the literature.

The rest of paper is organized as follows. Sections 2 and 3 review the GWO and FPA respectively. Section 4 presents the proposed GWOFPA approach. Section 5 discusses the simulation results. Section 6 gives the conclusion.

2 Grey Wolf Optimizer (GWO)

A new population-based nature-inspired algorithm called Grey Wolf Optimization (GWO) was developed by Mirjalili et al. [6]. GWO approach mimics the hunting behavior and social leadership of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the command hierarchy. The first three best position (fittest) wolves are indicated as α, β and δ who guide the other wolves (ω) of the groups toward promising areas of the search space. The position of each wolf of the group is updated using the following mathematical equations:

The *encircling behavior* of each agent of the crowd is calculated by the following mathematical equations:

$$\vec{d} = \left| c \cdot \vec{x}_p^t - \vec{x}^t \right| \tag{1}$$

$$\vec{x}^{t+1} = \vec{x}_p^t - \vec{a} \cdot \vec{d} \tag{2}$$

The vectors a and c are formulate as below:

$$\vec{a} = 2l \cdot r_1 \tag{3}$$

$$\vec{c} = 2 \cdot r_2 \tag{4}$$

Hunting: In order to mathematically simulate the hunting behavior, we suppose that the alpha, beta and delta have better knowledge about the potential location of prey. The following equations are developed in this regard.

$$\vec{d}_{\alpha} = |\vec{c}_1 \cdot \vec{x}_{\alpha} - \vec{x}|,\ \vec{d}_{\beta} = |\vec{c}_2 \cdot \vec{x}_{\beta} - \vec{x}|,\ \vec{d}_{\delta} = |\vec{c}_3 \cdot \vec{x}_{\delta} - \vec{x}| \tag{5}$$

$$\vec{x}_1 = \vec{x}_{\alpha} - \vec{a}_1 \cdot (\vec{d}_{\alpha}),\ \vec{x}_2 = \vec{x}_{\beta} - \vec{a}_2 \cdot (\vec{d}_{\beta}),\quad \vec{x}_3 = \vec{x}_{\delta} - \vec{a}_3 \cdot (\vec{d}_{\delta}) \tag{6}$$

$$\vec{x}^{t+1} = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3} \tag{7}$$

Search for prey and attacking prey:

The $\vec{a}$ is random value in the gap $[-2a, 2a]$. When random value $|\vec{a}| < 1$ the wolves are forced to attack the prey. Searching for prey is the exploration ability and attacking the prey is the exploitation ability. The arbitrary values of $\vec{a}$ are utilized to force the search to move away from the prey. When $|\vec{a}| > 1$, the members of the population are enforced to diverge from the prey.

3 Flower Pollination Algorithm (FPA)

The biological flower pollination inspires a new population-based algorithm called FPA [7]. The pollination rules state in FPA as follows. The global pollination considered cross-pollination that pollinators obey Lévy flights. The local pollination is known as self-pollination. The reproduction probability recognized flower constancy which is proportional to the resemblance of the two flowers in concerned. FPA used a switching probability p ∈ [0, 1] to control between the local and global pollination. Assumed, FPA considered as global and local pollination. Thus the local pollination is modeled as follows.

$$\vec{x}_{ij}^{t+1} = \vec{x}_{ij}^{t} + u \times \left(\vec{x}_{ih}^{t} - \vec{x}_{ik}^{t}\right) \tag{8}$$

where $\vec{x}_{ih}^{t}$, $\vec{x}_{ik}^{t}$ are pollen of different flowers but they are in the same plant species. u is generated from the uniform distribution [0, 1]. A random walk for local process if x_{ih}^{t} and x_{ik}^{t} come from the same species or selected from the same population of plants.

Pollens of the flowers in the global pollination are moved by pollinators e.g. insects, and pollens can be carried for a long distances. This process guarantees pollination and reproduction of the fittest solution represented as. The flower constancy is expressed mathematically as:

$$\vec{x}_{ij}^{t+1} = \vec{x}_{ij}^{t} + \gamma \times L(\lambda) \times \left(\vec{x}_{ij}^{t} - g^{*}\right) \tag{9}$$

where x_i is solution vector at iteration t, and γ is a scaling factor to control the step size. Lévy flight can be used to mimic the characteristic transporting of insects over a long distance with various length steps, thus, $L > 0$.

$$L = \frac{\lambda\Gamma(\lambda) \times \sin\left(\frac{\pi\lambda}{2}\right)}{\pi \times s^{i+\lambda}}, (s \gg s_0) \tag{10}$$

where $\Gamma(\lambda)$ is the standard gamma function, and this distribution is valid for large steps $s > 0$. A variable p is switching probability or the proximity probability that can be used to change the global pollination to intensive local pollination and reverse.

4 Hybrid Grey Wolf Optimizer-Flower Pollination Algorithm (GWOFPA)

This section presents the implementation of the low-level coevolutionary to hybridize GWO and FPA by merging the functionality of both approaches. Two different techniques including exploitation and exploration are involved in generating final optimal solution to the optimization problem. By this modification, we extend the performance of exploitation in GWO with the fulfillment of the exploration in FPA to produce both approaches' strength. However, FPA applies the exploration phase to updating positions.

$$\vec{d}_\alpha = \begin{cases} u \times (|\vec{c}_1 \times \vec{x}_\alpha - \vec{x}|), & rand() < 0.5 \\ \gamma \times L(\lambda) \times (|\vec{c}_1 \times \vec{x}_\alpha - \vec{x}|), & rand() \geq 0.5 \end{cases} \tag{11}$$

where $\vec{d}_\alpha$ is a modified dominance coefficient from Eq. (1) is to exploit diversity search agent for the proposed algorithm. The same done with the alpha, beta, and omega are applied in Eq. (5). New agent position can be mathematically simulated as follows.

$$\vec{x}_1 = \vec{x}_\alpha - \vec{a}_1 \times \left(\vec{d}_\alpha\right) \tag{12}$$

where $\vec{x}_1$ is done first agent position. Do the same for the second and third locations ($\vec{x}_2$ *and* $\vec{x}_3$ from Eq. (5). Figure 1 shows the pseudocode of the GWOFPA algorithm.

```
GWOFPA algorithm
Initialization the parameters such as:
population X_i (i = 1,2,..N), d,a and c
Compute the fitness of each search member X_α, X_β, X_δ
    // the first, second & third best search member
while (t < Max. iter)
    for every search member
        Get the current position by Eqs. (2) and (7)
    end for
      Get a , c, d by Eqs. (3)(4)and (11)
      Calculate the fitness of all search member
      Update X_α, X_β, X_δ by Eq.(12)
      t = t+1
end while
return X_α
```

Fig. 1. Pseudo code of the GWOFPA algorithm

5 Experimental Results

In this section, we evaluate the proposed GWOFPA approach performance quality by executing a set of benchmark problems to test the solution quality, solution stability, convergence speed and ability to find the global optimum. Twenty runs of the testing functions in the experiments outcome values are averaged with different random seeds.

Table 1 lists the initialization for a set of the trial functions [10]. Maxgen column in Table 1 is a maximum number of iterations (it can be set to 500, 1000, 1500,.., 10000).

Table 1. Initialization for dimensions, max generations, and boundaries of the testing functions

Testing problems	Bounds	Dims	Maxgen
$f_1(x) = \sum_{i=1}^{n} \sin(x_i)\cdot(\sin(\frac{ix_i^2}{\pi}))^{2m},\quad m = 10$	±500	30	2000
$f_2(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	±5.12	30	2000
$f_3(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	0, π	30	2000
$f_4(x) = [e^{-\sum_{i=1}^{n}(x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^{n} x_i^2}]\prod_{i=1}^{n}\cos^2 x_i,\quad m = 5$	±20	30	2000
$f_5(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	0, 10	4	2000
$f_6(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	0, 10	4	2000

Let $X = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ be the real value vectors of m-dimensional for GWOFPA. We set the population size N (N can be set to 10, 20,.., 100) for the algorithms of GWOFPA, GWO, FPA, and PSO to 40 for all runs in the experiments. Table 2 displays the outcome of implement for testing problems in Table 1 for the optimization is to maximize and minimize the results.

Table 2. The obtained results of proposed GWOFPA, GWO, and FPA for the benchmark functions

$f_i(x)$	PSO		GWO		FPA		**GWOFPA**	
	f_{min}	f_{max}	f_{min}	f_{max}	f_{min}	f_{max}	f_{min}	f_{max}
1	1.61E+01	6.87E+04	8.80E−06	4.87E+04	7.96E−06	8.06E+04	**6.39E−06**	**7.89E+04**
2	1.39E+01	7.41E+14	7.50E−02	1.23E+12	7.27E−02	2.13E+11	**6.10E−02**	**5.71E+12**
3	3.25E+03	1.09E+05	3.96E−01	1.25E+05	4.12E−01	2.76E+05	**3.48E−01**	**1.12E+05**
4	1.79E+01	8.67E+01	2.02E+00	7.97E+01	2.00E+00	8.62E+01	**1.71E+00**	**8.65E+01**
5	8.88E+02	3.09E+08	3.23E+01	1.08E+08	3.88E+01	2.53E+08	**2.72E+01**	**2.54E+08**
6	9.09E+01	7.38E+04	4.36E+00	4.62E+04	4.79E+00	7.55E+04	**3.96E+00**	**6.63E+04**
AVG	1.08E+01	1.30E+02	6.53E+00	1.38E+02	7.68E+00	1.16E+02	**5.55E+00**	**1.29E+02**

We tested each benchmark function with 2000 iterations per a run. We also compare the simulation results of the proposed method with those obtained results of the previous algorithms such as the GWO, FPA, and PSO as shown in Tables 3 and 4. For further parameters setting could be found in [6, 7] with the initial range, the dimension and total iterations for all test functions in Table 1 for GWO, FPA, and PSO.

Table 3. Comparison of the the proposed GWOFPA with GWO, and FPA, with for solving the testing problems in term of the quality performance evaluation

Testing functions	Obtained results			Comparison	
	GWO	FPA	**GWOFPA**	with GWO	with FPA
1	8.80E−06	7.96E−06	6.39E−06	27%	20%
2	7.50E−02	7.27E−02	6.10E−02	19%	16%
3	3.96E−01	4.12E−01	3.48E−01	12%	16%
4	2.02E+00	2.00E+00	1.71E+00	15%	14%
5	3.23E+01	3.88E+01	2.72E+01	16%	30%
6	4.36E+00	4.79E+00	3.96E+00	9%	17%
Avge	**6.53E+00**	**7.68E+00**	**5.55E+00**	**16%**	**19%**

Table 4. Comparison of the proposed algorithm quality performance with PSO for testing problems

Testing functions	Execution time		Comp. times	Performances		Comp. qualities
	PSO	GWOFPA		PSO	GWOFPA	
1	1.4793	1.4371	2%	1.51E+00	1.22E+00	23%
2	1.8672	1.8686	3%	−4.17E+03	−5.09E+03	18%
3	1.9871	1.9503	4%	1.69E+02	1.37E+02	23%
4	0.7776	0.7834	1%	2.70E-03	2.50E-03	8%
5	1.8918	1.9072	2%	−2.34E+00	−3.09E+00	24%
6	1.9891	1.9792	1%	−8.15E+00	−9.82E+00	17%
Avg	**1.1703**	**1.1543**	**2%**	**−6.68E+02**	**−8.27E+02**	18%

Figures 2 and 3 show the experimental results for the first two benchmark functions over 20 times output obtained from the proposed GWOFPA, FPA, GWO, and PSO methods with the same iteration of 2000. Appearly, these figures show all of the cases of testing functions in the GWOFPA have performance quality higher the other algorithm regarding the accuracy and convergence rate.

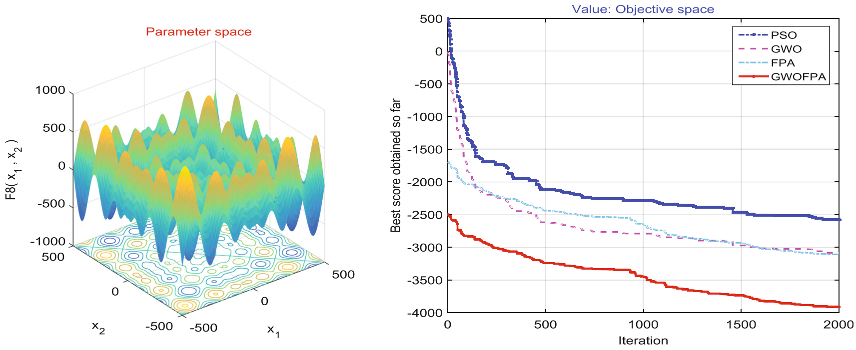

Fig. 2. Comparison the experimental result curves of the proposed GWOFPA with PSO, FPA, and GWO for the testing function f1

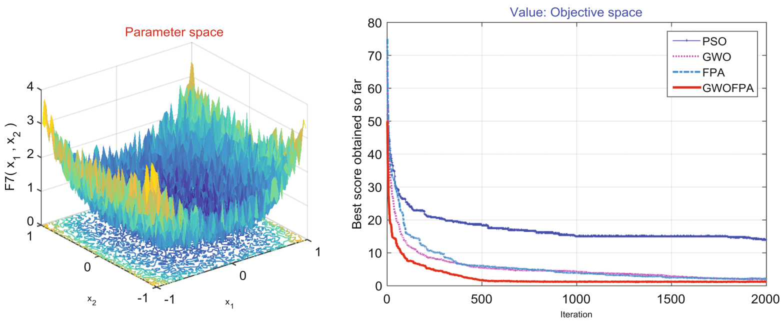

Fig. 3. Comparison the experimental result curves of the proposed GWOFPA with PSO, FPA, and GWO for the testing function f2

Table 3 displays the performance of the proposed GWOFPA in comparison with the algorithms of GWO, FPA for the testing functions. The result of the proposed algorithm on all of these cases of testing shows that the proposed algorithm provides 27% and 30% higher than those obtained from primary methods of GWO and FPA respectively. However, the figure for the minimum cases is only the increase 9% and 14% than the GWO and FPA respectively for a set of testing functions. In general, the proposed GWOFPA increases the average values of the cases 16% and 19% than obtained from the GWO and FPA methods are respectively for testing problems in terms of the convergence rates.

Table 4 displays the performing quality and running time comparison of the proposed GWOFPA with PSO method for the testing problems. The columns of correlation times and conditions are calculated as absolute of the obtained from GWOFPA minus that got from PSO then divided the received value of the GWOFPA method. The results of the proposed method on all of these cases of testing multimodal benchmark problems show that GWOFPA process almost increases higher quality and shorter running time than those obtained from PSO method. In general, the proposed algorithm achieved the standard cases of various tests for the convergence, and accuracy increased more than those obtained from the PSO method is 18%, and for the speed faster than that got from PSO method is 2% average respectively.

6 Conclusion

In this paper, we presented a novel hybrid approach for the optimization applications based on a combination of Grey Wolf Optimizer (GWO) and Flower Pollination Algorithm (PFA), namely GWOFPA. We use the location update equation of FPA for updating the positions of the grey wolves in GWO to explore and exploit the diversity of the algorithm efficiently. In the simulation, a set of functions are applied to verify the accuracy, convergent behavior, best global optimal solution of the newly developed approach. Results reveal that the proposed approach provides highly competitive solutions as compared to other algorithms.

References

1. Nguyen, T.-T., Pan, J.-S., Chu, S.-C., Roddick, J.F., Dao, T.-K.: Optimization localization in wireless sensor network based on multi-objective firefly algorithm. J. Netw. Intell. **1**, 130–138 (2016)
2. Dao, T., Pan, T., Nguyen, T.: A compact artificial bee colony optimization for topology control scheme in wireless sensor networks. J. Inf. Hiding Multimed. Sig. Process. **6**, 297–310 (2015)
3. Boussaïd, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. Inform. Sci. **237**, 82–117 (2013)
4. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43. IEEE, New York (1995)
5. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer (Long. Beach. Calif). **27**, 17–26 (1994)
6. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey Wolf Optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
7. Yang, X.S.: Flower pollination algorithm for global optimization. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 240–249 (2012)
8. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: González, J., Pelta, D., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) Studies in Computational Intelligence, pp. 65–74. Springer, Heidelberg (2010)
9. Pan, T.-S., Dao, T.-K., Nguyen, T.-T., Chu, S.-C.: Hybrid particle swarm optimization with bat algorithm. In: Advances in Intelligent Systems and Computing, pp. 37–47 (2015)