A Hybrid Multi-Objective Firefly-Sine Cosine Algorithm for Multi-Objective Optimization Problem

Gaohuizi Guo, Ning Zhang

Abstract**—**Firefly algorithm (FA) and Sine Cosine algorithm (SCA) are two very popular and advanced metaheuristic algorithms. However, these algorithms applied to multi-objective optimization problems have some shortcomings, respectively, such as premature convergence and limited exploration capability. Combining the privileges of FA and SCA while avoiding their deficiencies may improve the accuracy and efficiency of the algorithm. This paper proposes a hybridization of FA and SCA algorithms, named multi-objective firefly-sine cosine algorithm (MFA-SCA), to develop a more efficient meta-heuristic algorithm than FA and SCA.

Keywords**—**Firefly algorithm, hybrid algorithm, multi-objective optimization, Sine Cosine algorithm.

I. Introduction

SWARM intelligence algorithm has become an effective tool for numerical optimization. Up to now, a number of swarm intelligence algorithms have been mentioned in literature, such as Particle Swarm Optimization (PSO) [1], Ant colony optimization (ACO) [2], Artificial Bee Colony (ABC) [3], etc. FA [4] and SCA [5] are well known optimization algorithms in this category.

Recently, FA as an advanced metaheuristic algorithm was proposed by [4]. This algorithm simulates the behavior of fireflies based on their flash characteristics. Compared with some widely used metaheuristic algorithms, FA has the characteristics of fast convergence speed and simplicity. Thus, FA was widely used in optimization problems, such as flow shop scheduling problem and electric power plant planning problem. Multi-objective firefly algorithm (MOFA) was extended by Yang [6] for multi-objective optimization problem. Following this work, Marichelvam et al. [7] introduced a discrete FA to solve the multi-objective hybrid flow shop scheduling problem. Karthikeyan et al. [8] proposed a hybrid discrete FA to solve the multi-objective flexible job shop scheduling problem. Hidalgo-Paniagua et al. [9] used MOFA to solve the multi-objective path planning problem. Bozorg-Haddad et al. [10] used an extended multi-objective developed FA for hydropower energy generation. Lu et al. [11] proposed a hybrid MFA-SCA to solve a multi-objective multi-period regret minimization uncertain portfolio selection model with bankruptcy constraint.

Although FA algorithm has been widely used in optimization problems, there are still many shortcomings. Some researches showed that FA has the defects of premature convergence and easy to fall into local optimum [12], [13]. And the step sizes moved in the firefly are highly random, which may result in skipping the optimal solution. Moreover, SCA is a new population-based intelligent optimization algorithm proposed by [5]. The SCA algorithm includes several random variables and adaptive variables, and guarantees the diversity of the search solution as much as possible on the premise of ensuring the optimal solution vector. Therefore, inspired by the SCA search mechanism, we hybridize FA and SCA, termed MFA-SCA, for overcoming the shortcomings. In the hybrid MFASCA, a search strategy is proposed to update the dominant individuals. Besides, non-dominant individuals move towards other non-dominant individuals by employing the SCA movement strategy.

Gaohuizi Guo is with the School of Management and Engineering, Capital University of Economics and Business, Beijing, China, 100070 (e-mail: huizi@cueb.edu.cn).

Ning Zhang is with the School of Management and Engineering, Capital University of Economics and Business, Beijing, China, 100070 (corresponding author, e-mail: zhangning@cueb.edu.cn).

II. The Basic FA

FA is an advanced metaheuristic algorithm proposed by [4], which comes from the simplification and simulation of firefly group behavior. The FA follows the three rules: (I) Fireflies are unisex; (II) The attraction of a firefly is directly proportional to its brightness. For any two fireflies, a firefly with low brightness will be attracted to a firefly with high brightness. This attraction is inversely proportional to the distance between fireflies. As the distance between fireflies increases, the attraction gradually decreases; (III) The brightness of the firefly is determined by the objective function to be optimized.

The attractiveness is determined by the distance between two fireflies, so the attraction for firefly i and firefly j can be expressed as

$$\beta_{i,j}(r_{i,j}) = \beta_0 e^{-\gamma r_{r,j}^2}, \tag{1}$$

where β_0 is the maximum attraction, which is the attraction at the light source ($r = 0$). γ is the light absorption coefficient and represents the change of attraction. Its value has a great influence on the convergence speed of the algorithm. $r_{i,j}$is the cartesian distance between two fireflies,

$$r_{i,j} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})^2}, \tag{2}$$

where D is the dimension of the objective function to be optimized, $x_{i,k}$ and $x_{j,k}$ are the kth component element of x_i and x_j, respectively.

$$x_i^{t+1} = x_i^t + \beta_{i,j}(r_{i,j})(x_j^t - x_i^t) + \alpha_t \epsilon_i^t, \tag{3}$$

where α_t is a random number, and ϵ_i^t is a random number vector from a Gaussian, a uniform distribution, or some other distributions.

III. The Hybrid MFA-SCA

A. Initialization

Population initialization is the first step of population-based meta-heuristic algorithms. Meanwhile, it is a crucial task because it can affect the convergence speed and the quality of the final solutions. In most meta-heuristic algorithms, the initial population is randomly generated. This method is widely used in most population-based meta-heuristic algorithms due to its simplicity. For simplification, random initialization is also used in this paper. Random population of SN fireflies are generated using

$$x_{i,k} = lb_k * W_0 + rand(0,1) * (ub_k * W_0 - lb_k * W_0),$$

where $x_{i,k}$ is the amount of the kth portfolio's asset of the ith agent, $rand(0,1)$ is a random number uniformly distributed between 0 and 1, W_0 is the initial wealth, and ub_k and lb_k are upper and lower weight bounds of the kth asset, respectively.

If the initially generated value for the kth parameter of the ith firefly does not fit in the scope $[lb_k * W_0, ub_k * W_0]$, it is modified using the following expression:

- if $x_{i,k} > ub_k * W_0$, then $x_{i,k} = ub_k * W_0$,
- if $x_{i,k} < lb_k * W_0$, then $x_{i,k} = lb_k * W_0$.

B. The Movement of Dominant Fireflies

In order to improve the performance of basic FA, a search strategy for dominant fireflies is proposed. For a dominant firefly i, it moves towards the firefly j that dominates itself. The position update formula is described as

$$x_i^{t+1} = x_i^t + \beta_{i,j}(r_{i,j})(x_j^t - x_i^t) + r1 \times sin(r2) \times \epsilon_i^t, r4 < 0.5, \quad (4)$$

$$x_i^{t+1} = x_i^t + \beta_{i,j}(r_{i,j})(x_j^t - x_i^t) + r1 \times cos(r2) \times \epsilon_i^t, r4 \geq 0.5, \quad (5)$$

where parameters $r1$, $r2$ and $r4$ are the three parameters introduced by SCA. Parameters $r2$ and $r4$ are random numbers in different intervals. Parameter $r2$ represents the size of the disturbance, while $r4$ determines whether the individual moves sinusoidal or cosine. Parameter $r1$ is an adaptive variable, which represents the direction of the disturbance. The expression for the parameter $r1$ is

$$r1 = a - t\frac{a}{T}, \quad (6)$$

where T is the maximum number of iterations preset and a is a constant.

C. The Movement of Non-Dominant Fireflies

For non-dominant individuals, Yang [6] weights multiple objectives into single objective via the method of random weighted sum. If firefly i is not dominated by any other fireflies, its location is updated as below:

$$\varphi(x) = \sum_{l=1}^{n} \omega_l f_l,\ \sum_{l=1}^{n} \omega_l = 1, x_i^{t+1} = g_{best}^t + \alpha_t \epsilon_i^t, \quad (7)$$

where ω_l is a random number between 0 and 1, f_l is the lth objective function, and g_{best}^t is the optimal obtained by (7). However, this mechanism may lead to incomplete exploration. According to [5], SCA not only has good exploration capability, but also performs well in exploitation. So we introduce the movement strategy of SCA for non-dominant fireflies. The movement formula of the non-dominant individual can be expressed as

$$x_i^{t+1} = x_i^t + r1 \times sin(r2) \times |r3P^t - x_i^t|, r4 < 0.5, \quad (8)$$

$$x_i^{t+1} = x_i^t + r1 \times cos(r2) \times |r3P^t - x_i^t|, r4 \geq 0.5, \quad (9)$$

where x_i^t represents the current position of the ith dimensional solution vector of the individual in the tth iteration. Respectively, $r1$, $r2$ and $r4$ are adaptive and random variables as explained above. The parameter $r3$ represents a random weight, which is a random number from [0, 2]. If $r3 > 1$, it indicates that this iteration has significant influence on the approximation to the optimal value; otherwise, it is not significant. P^t is the position of any non-dominant individual in iteration t.

Finally, the pseudo-code of hybrid MFA-SCA is presented in Algorithm 1.

Algorithm 1: The pseudo-code of MFA-SCA

1: Objective functions $f_1(x_i), \cdots, f_k(x_i), x_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T$
2: Randomly generate SN fireflies x_i $(i = 1, 2, \cdots, SN)$
3: Initialize parameters in MFA-SCA
4: **While** $(t < MaxGeneration)$
5:   **For** $i = 1: SN$ all SN fireflies
6:     **For** $i = 1: SN$ all SN fireflies
7:       **If** x_j dominates x_i
8:         Update $r_{i,j}$, $\beta_{i,j}(r_{i,j}), r1, r2, r4$
9:         Update the ith firefly → the jth firefly by (4) or (5)
10:       **End if**
11:       **If** non-dominated solution fulfilled
12:         Update $r1, r2, r3, r4$
13:         Update the ith firefly → the jth firefly by (8) or (9)
14:       **End if**
15:     **End for**
16:   **End for**
17:   Put the non-dominate solutions in external archive
18: **End while**
19: Select the Pareto front from the external archive
20: Postprocess results and visualization

IV. Numerical Example

In this section, a numerical experiment is presented to demonstrate the effectiveness of the MFA-SCA algorithm. To evaluate the performance of the designed MFA-SCA algorithm, we compare it with the basic FA, SCA, PSO and GA.

Six widely adopted metrics are used to evaluate the performance of the proposed MFA-SCA algorithm. Moreover, we select five ZDT functions as benchmarks. The details of the five ZDT functions are listed in Table I.

TABLE I
MULTI-OBJECTIVE TEST FUNCTIONS UTILIZED IN THIS PAPER

Problem	Definition
ZDT1	$f_1(x) = x_1$
	$f_2(x) = g(x) * \left(1.0 - \sqrt{\frac{f_1}{g(x)}}\right)$
	$g(x) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
	$0 \le x_i \le 1, i = 1, \cdots, n$
ZDT2	$f_1(x) = x_1$
	$f_2 = g(x) * [1.0 - (x_1/g(x))^2]$
	$g(x) = 1 + \frac{9}{n-1}(\sum_{i=2}^{n} x_i)$
	$0 \le x_i \le 1, i = 1, \cdots, n$
ZDT3	$f_1(x) = x_1$
	$f_2 = g(x)[1 - \sqrt{x_1/g(x)} - x_1/g(x)\sin(10\pi x_1)]$
	$g(x) = 1 + \frac{9}{n-1}(\sum_{i=2}^{n} x_i)$
	$0 \le x_i \le 1, i = 1, \cdots, n$
ZDT4	$f_1(x) = x_1$
	$f_2 = g(x)[1 - (x_1/g(x))^2]$
	$g(x) = 1 + 10(n-1) + \sum_{i=1}^{n}(x_i^2 - 10\cos(4\pi x_i))$
	$0 \le x_1 \le 1, -5 \le x_i \le 5, i = 1, \cdots, n$
ZDT6	$f_1(x) = 1 - e^{-4x_1} * \sin^6(6\pi x_1)$
	$f_2 = 1 - (\frac{f_1(x)}{g(x)})^2$
	$g(x) = 1 + 9 * [(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$
	$0 \le x_i \le 1, i = 1, \cdots, n$

The real Pareto optimal fronts of all benchmark functions involved are known. The Inverted Generation Distance (IGD) [14], the Generation Distance (GD) [14], the Maximum Spread (MS) [15] and the Spacing [16] are used as evaluation parameters.

(1) Inverted Generation Distance (IGD)

$$IGD = \frac{\sqrt{\sum_{i=1}^{m} d_i^2}}{m},$$

where m is the number of solutions in the true Pareto front, and d_i is the Euclidean distance between each of the solutions and the nearest member from the set of non-dominated solutions found by the algorithm. If $IGD = 0$, the solutions are evenly distributed on the true Pareto frontier. If $IGD > 0$, it means that the solutions may have poor diversity.

(2) Generation Distance (GD)

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n},$$

where n is the number of non-inferior solutions obtained by the algorithm, and d_i is the minimum distance from the ith solution to the real Pareto optimal solution set. If $GD = 0$, the resulting non-dominated solution belongs to the real Pareto optimal solution set. It reflects the degree of approximation between the optimal solution set obtained by the algorithm and the real Pareto optimal solution set.

(3) Maximum Spread (MS)

$$MS = \sqrt{\frac{1}{k}\sum_{l=1}^{k} \delta_l}, \delta_l = \left(\frac{\min(f_l^{max}, F_l^{max}) - \max(f_l^{min}, F_l^{min})}{F_l^{max} - F_l^{min}}\right),$$

where f_l^{max} and f_l^{min} are the maximum and minimum values of the function value of the lth objective function of the Pareto optimal frontier obtained by the algorithm, F_l^{max} and F_l^{min} are the maximum and minimum values of the function value of the lth objective function of the real Pareto frontier. k is the number of objective functions. If $MS = 1$, it indicates that the true Pareto frontier is completely covered by the solutions obtained by the algorithm. MS is the metric representing the coverage of the Pareto frontier of the algorithm to the true Pareto front.

TABLE II
COMPARISON OF PERFORMANCE RESULTS FOR FIVE ALGORITHMS

			MFA-SC	FA	SCA	PSO	GA
IGD	ZDT1	Mean	**0.002860**	0.007234	0.011792	0.007075	0.007508
		Std	**0.000891**	0.002542	0.013664	0.002217	0.002873
	ZDT2	Mean	**0.010632**	0.022029	0.013231	0.016450	0.021813
		Std	**0.003643**	0.008579	0.004674	0.018346	0.006953
	ZDT3	Mean	**0.006444**	0.009099	0.009827	0.015636	0.013884
		Std	**0.002278**	0.002955	0.003067	0.003950	0.005253
	ZDT4	Mean	**0.005079**	0.020723	0.018165	0.007822	0.012304
		Std	**0.001426**	0.006166	0.007885	0.002793	0.004667
	ZDT6	Mean	**0.006666**	0.010397	0.013167	0.014693	0.008145
		Std	**0.001389**	0.006323	0.005836	0.003770	0.006140
GD	ZDT1	Mean	**0.000556**	0.000581	0.000684	0.000572	0.001092
		Std	0.000101	0.000160	0.000132	**0.000028**	0.000158
	ZDT2	Mean	0.001406	0.001504	0.000934	**0.000630**	0.003081
		Std	0.000379	0.000836	0.000563	**0.000066**	0.000861
	ZDT3	Mean	**0.000466**	0.000559	0.000730	0.001284	0.001546
		Std	**0.000116**	0.000158	0.000131	0.000309	0.000497
	ZDT4	Mean	0.000715	**0.000430**	0.000949	0.000814	0.001461
		Std	**0.000116**	0.000179	0.000266	0.000156	0.000337
	ZDT6	Mean	**0.000202**	0.000311	0.000275	0.000216	0.000761
		Std	**0.000022**	0.000059	0.000059	0.000030	0.000112
MS	ZDT1	Mean	**0.999082**	0.932220	0.812656	0.830495	0.896555
		Std	**0.002445**	0.097031	0.201386	0.094205	0.073931
	ZDT2	Mean	**0.868995**	0.748130	0.759269	0.738695	0.762865
		Std	**0.073443**	0.153958	0.096346	0.255171	0.126673
	ZDT3	Mean	**0.967455**	0.951600	0.960974	0.724964	0.918257
		Std	**0.046204**	0.073045	0.067321	0.114628	0.114019
	ZDT4	Mean	**0.919843**	0.762493	0.667330	0.847205	0.915493
		Std	**0.062578**	0.159058	0.107288	0.075307	0.094733
	ZDT6	Mean	**0.792226**	0.785368	0.725480	0.505887	0.786655
		Std	**0.065450**	0.149158	0.156857	0.097615	0.140393
Spacing	ZDT1	Mean	**0.049482**	0.092751	0.051762	0.052032	0.131580
		Std	**0.014376**	0.027327	0.029388	0.015624	0.041059
	ZDT2	Mean	0.133252	0.303143	0.068866	**0.040232**	0.357714
		Std	0.044552	0.182032	0.034969	**0.016414**	0.144461
	ZDT3	Mean	**0.064085**	0.100596	0.104284	0.137508	0.243071
		Std	**0.019687**	0.028600	0.031787	0.026083	0.075723
	ZDT4	Mean	**0.066580**	0.100126	0.097518	0.093033	0.236209
		Std	**0.015129**	0.040062	0.038292	0.021954	0.080608
	ZDT6	Mean	0.030920	0.044462	0.047643	**0.022111**	0.104623
		Std	**0.005296**	0.012522	0.023382	0.006233	0.025939

(4) Spacing

$$Spacing = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}_l - d_l)^2},$$

where n is the number of non-dominated solutions obtained by the algorithm, d_l is the distance between the ith non-dominated solution corresponding to the target vector and the nearest target vector, and d_l is the mean of the $\bar{d}_l$. If $Spacing = 0$, it means that the solutions are evenly distributed.

Table II summarizes means and standard deviations of IGD, GD, MS and Spacing in all algorithms for the five test functions. The best results are marked in hold. We observed that MFA-SCA outperforms the other four algorithms in terms of IGD and MS. For GD and Spacing, MFA-SCA algorithm is superior to the other algorithms in most cases, although PSO have better performance than the proposed algorithm for ZDT2. In general, in most cases, the results obtained by the hybrid MFA-SCA algorithm are better than those obtained by the other algorithms in terms of above four metrics. That is to say, the proposed MFA-SCA algorithm outperforms the other meta-heuristic algorithms.

Finally, the correlation coefficient ω of IGD, GD, MS and Spacing obtained by five algorithms are evaluated by conducting Shapiro-Wilk W test under the confidence level of 95%. The closer to 1, the more normally distributed the dataset is. Table III also shows that the values of IGD, GD, MS and Spacing obtained by MFA-SCA are relatively more normally distributed than those obtained by other algorithms in most cases.

TABLE III
THE CORRELATION COEFFICIENT W OF DIFFERENT PERFORMANCE METRICS OBTAINED BY CONDUCTING SHAPIRO-WILK W TEST

		MFA-SC	FA	SCA	PSO	GA
ZDT1	IGD	**0.973090**	0.907640	0.623230	0.960430	0.885480
	GD	**0.979540**	0.889190	0.914480	0.976770	0.965520
	MS	0.628200	0.752110	0.744260	**0.937880**	0.853500
	Spacing	**0.978830**	0.879420	0.675940	0.959040	0.895000
ZDT2	IGD	**0.975550**	0.965390	0.962110	0.628890	0.967280
	GD	**0.975690**	0.875730	0.661220	0.970000	0.972540
	MS	**0.951090**	0.879500	0.948670	0.836870	0.948010
	Spacing	**0.988680**	0.959740	0.873860	0.978860	0.953260
ZDT3	IGD	**0.954230**	0.922540	0.954050	0.949700	0.871460
	GD	**0.977100**	0.914950	0.953970	0.942840	0.886540
	MS	0.593570	0.634330	0.507230	**0.946760**	0.653810
	Spacing	**0.962260**	0.941120	0.880730	0.883920	0.847310
ZDT4	IGD	**0.958360**	0.942900	0.930220	0.909490	0.919210
	GD	**0.959400**	0.954320	0.888270	0.958490	0.924580
	MS	0.896430	0.929500	**0.983330**	0.988300	0.563980
	Spacing	**0.968410**	0.953660	0.871880	0.961900	0.928230
ZDT6	IGD	0.881740	0.836080	0.886150	0.944470	**0.981350**
	GD	**0.963350**	0.947740	0.922330	0.958440	0.957350
	MS	0.953880	0.883790	0.934570	0.951150	**0.984310**
	Spacing	**0.974070**	0.902000	0.829550	0.970230	0.911520

V. CONCLUSIONS

In this paper, we design a hybrid MFA-SCA algorithm combing FA and SCA. Then, we present a numerical example to illustrate the effectiveness of the proposed approach. The experimental results demonstrate that the proposed MFA-SCA algorithm has better performance than the other algorithms. In future work, we may apply the proposed MFA-SCA to solve more real-world multi-objective optimization problems, such as vehicle routing problem.

REFERENCES

[1] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," in: Neural Networks, 1995. Proceedings, IEEE International Conference on, volume 4, IEEE, 1995, pp. 1942–1948.
[2] M. Dorigo, G. Di Caro, "Ant colony optimization: a new meta-heuristic," in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 2, IEEE, 1999.
[3] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization." Techn. Rep. TR06, Erciyes Univ. Press, Erciyes, 2005.
[4] X.S. Yang, Nature-inspired metaheuristic algorithms, Luniver Press, Beckington, 2008.
[5] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," Knowledge-Based Systems, vol. 96, pp. 120–133, 2016.
[6] X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," Engineering with Computers, vol. 29, no. 2, pp. 175–184, 2013.
[7] M. K. Marichelvam, T. Prabaharan, and X. S. Yang, "A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems," IEEE Transactions on Evolutionary Computation, vol. 18, no. 2, pp. 301–305, 2014.
[8] S. Karthikeyan, P. Asokan, S. Nickolas, and T. Page, "A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems," International Journal of Bio-Inspired Computation, vol. 7, no. 6, pp. 386–401, 2015.
[9] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multiobjective path planning problem in mobile robotics with a firefly-based approach," Soft Computing, vol. 21, pp. 949–964, 2017.
[10] O. Bozorg-Haddad, I. Garousi-Nejad, and H. A. Loáiciga, "Extended multi-objective firefly algorithm for hydropower energy generation," Journal of Hydroinformatics, vol. 19, no. 5, pp. 734–751, 2017.
[11] S. Lu, N. Zhang, Y. Qiu, and Y. Gao, "A multi-period regret minimization model for uncertain portfolio selection with bankruptcy constraint," Journal of Intelligent and Fuzzy Systems, vol. 37, no. 6, pp. 8417–8439, 2019.
[12] Y. Aref and K. Cemal, "A modified firefly algorithm for global minimum optimization," Applied Soft Computing, vol. 62, pp. 29–44, 2018.
[13] J. Zhang, Y.F. Teng and W. Chen, "Support vector regression with modified firefly algorithm for stock price forecasting," Applied Intelligence, vol. 49, pp. 1658–1674, 2019.
[14] D. A. Van, V. Gary and B. Lamont. "Multiobjective evolutionary algorithm research: Ahistory and analysis, " Evolutionary Computation, vol. 8, pp. 125–147, 1998.
[15] H. Xing, Z. Wang, T. Li, H. Li and R. Qu. "An improved MOEA/D algorithm for multi-objective multicast routing with network coding," Applied Soft Computing, vol. 59, pp. 88–103, 2017.
[16] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Cellular Immunology, vol. 37, pp. 1–13, 1995.