

Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 8034573, 10 pages
<https://doi.org/10.1155/2017/8034573>



Research Article

Firefly Mating Algorithm for Continuous Optimization Problems

**Amarita Ritthipakdee,^{1,2} Arit Thammano,^{1,2} Nol Premasathian,²
and Duangjai Jitkongchuen³**

¹Computational Intelligence Laboratory, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

²Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

³College of Innovative Technology and Engineering, Dhurakij Pundit University, Bangkok, Thailand

Correspondence should be addressed to Arit Thammano; arit@it.kmitl.ac.th

Received 15 January 2017; Revised 22 May 2017; Accepted 14 June 2017; Published 20 July 2017

Academic Editor: Leonardo Franco

Copyright © 2017 Amarita Ritthipakdee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a swarm intelligence algorithm, called firefly mating algorithm (FMA), for solving continuous optimization problems. FMA uses genetic algorithm as the core of the algorithm. The main feature of the algorithm is a novel mating pair selection method which is inspired by the following 2 mating behaviors of fireflies in nature: (i) the mutual attraction between males and females causes them to mate and (ii) fireflies of both sexes are of the multiple-mating type, mating with multiple opposite sex partners. A female continues mating until her spermatheca becomes full, and, in the same vein, a male can provide sperms for several females until his sperm reservoir is depleted. This new feature enhances the global convergence capability of the algorithm. The performance of FMA was tested with 20 benchmark functions (sixteen 30-dimensional functions and four 2-dimensional ones) against FA, ALC-PSO, COA, MCP SO, LWG SODE, MPSODDS, DFOA, SHPSOS, LSA, MPDPGA, DE, and GABC algorithms. The experimental results showed that the success rates of our proposed algorithm with these functions were higher than those of other algorithms and the proposed algorithm also required fewer numbers of iterations to reach the global optima.

1. Introduction

Optimization is a process of finding the most desirable solution to the problem of interest. Optimization problems can be divided into two main categories depending on the types of variables involved (discrete or continuous). A problem with discrete variables is known as a combinatorial optimization problem while a problem with continuous variables is known as a continuous optimization problem. The optimization algorithm proposed in this paper only emphasizes solving continuous optimization problems. Optimization algorithms are of many varieties ranging from simple search to much more complex ones that require a great deal of computational time. At present, among the widely researched types of optimization algorithms, self-organized algorithms employing procedures that emulate animal swarm behavior are very popular. Brief overviews of recent metaheuristic algorithms

based on firefly, ant, fish, bird, and bee are presented in the next paragraph.

Yang [1] developed a swarm intelligent algorithm based on the flashing behavior of fireflies. In nature, the firefly flashes its light to attract other fireflies for mating. The less bright firefly is attracted and moves towards the one that flashes brighter light. The experimental results in Yang's paper showed that the firefly algorithm outperformed both genetic algorithm and particle swarm optimization on the tested benchmarks. Similar results were also obtained by Baykasoğlu and Ozsoydan [2]. Even though the algorithm is very promising, further investigation is needed to improve on the convergence of the algorithm and to prevent trapping in local optima. Farahani et al. [3] proposed an improved firefly algorithm called Gaussian distribution firefly (GD-FF) algorithm. Three new strategies were introduced: (i) an adaptive step length that changes with time, (ii) a directed

movement towards the global best solution when there is no brighter firefly within its vicinity, and (iii) a social behavior that changes the position of each firefly based on a Gaussian distribution. The authors found that their algorithm got better result and converged to the global optimum faster than the original firefly algorithm. Rizk-Allah et al. [4], on the other hand, improved the firefly algorithm in a different way; they combined it with an ant colony optimization (ACO) algorithm. First, this hybrid algorithm is initialized using the ACO algorithm. Then the firefly algorithm is put to work as a local search to refine the solutions found by the ants. The hybrid was proven successful. Shin and Kita [5] improved the performance of particle swarm optimization (PSO) by including the information of the second global best and second personal best particles in their developed algorithm. The algorithm's performance was found to be better than the original PSO. Another variant of PSO was an algorithm called Levy flight particle swarm optimization (LFPSO) [6]. This algorithm combined PSO with Levy flight. Experimental results showed that the LFPSO clearly outperformed the original PSO and other PSO variants in terms of solution quality and robustness. A different kind of swarm intelligent algorithm based on the behavior of bees in finding food sources, called bee colony optimization (BCO), was also further developed by assigning two weights to each food source. One is a global weight and the other is a local weight. As a result, the proposed algorithm, named weighted bee colony optimization (wBCO) [7], performed better than the original BCO and many state-of-the-art algorithms. Artificial bee colony (ABC), another bee-inspired algorithm, was modified by Gao and Liu [8]. In their proposed algorithm, Levy flight, differential evolution, and particle swarm optimization were incorporated into the original ABC algorithm. Experimental results showed that the proposed algorithm achieved better performance than the original ABC algorithm and many of its variants. Niu et al. [9] proposed an improved version of the original fruit fly optimization algorithm (FOA), called DFOA. They modified the expression of the smell concentration judgement value and introduced DE/best/1 mutation strategy to replace the random search. The results showed that DFOA was superior to seven other evolutionary computation methods. Chen et al. [10] presented a new PSO variant called particle swarm optimization with an aging leader and challengers (ALC-PSO). In ALC-PSO, the leader of the swarm has a lifespan. If the leader is trapped in a local optimum, it will age quickly and new challengers will emerge to replace the old leader. As a result, ALC-PSO has an ability to escape from local optima and to find global optima in a short period of time. Genetic algorithm (GA) was also developed further by Thakur [11] who proposed a new crossover operator, called double Pareto crossover operator (DPX). DPX was used in combination with power mutation (PM) operator. He tested this algorithm with multimodal nonlinear functions and found that its success rate was higher than that of the standard GA.

In this paper, a swarm intelligence algorithm called firefly mating algorithm (FMA) is presented. In FMA, GA is used as the core of the algorithm. A new mating pair selection

method, inspired by the mating behavior of fireflies in nature, is introduced and incorporated into GA.

The rest of the paper is organized as follows: Sections 2 and 3 briefly discuss the original firefly algorithm and genetic algorithm, respectively; in Section 4, the biological background of the proposed algorithm is explained; in Section 5, the proposed model with a detailed description of each step is presented; in Section 6, a description of the experiments is given; in Section 7, experimental results are shown and compared; finally, Section 8 is the conclusion.

2. Firefly Algorithm

This section describes the original firefly algorithm. Firefly algorithm (FA) is a reliable and efficient metaheuristic algorithm capable of solving many real-world problems such as scheduling, optimization problems in dynamic environments, and economic load dispatch problem. This algorithm is influenced by the flashing behavior of fireflies to attract one another. It is constructed based on three rules [12]:

(i) All fireflies are unisex so that one firefly is attracted to all other fireflies.

(ii) The attractiveness of a firefly is proportional to its brightness. For any two fireflies, the dimmer one is attracted by (and thus moves towards) the brighter one. However, if there are no fireflies brighter than a given firefly, that firefly will move randomly.

(iii) The brightness of a firefly decreases as the distance from it increases. This is because light is absorbed when it passes through the medium. Therefore, the brightness (also attractiveness) of the firefly j seen by the firefly i is defined in (1).

$$\beta_j(r) = \beta_j(0) e^{-\gamma r^2}, \quad (1)$$

$$r = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_i^k - x_j^k)^2}, \quad (2)$$

where γ is a light absorption coefficient of the medium, r is the Euclidean distance between the firefly i and the firefly j , $\beta_j(0)$ is the brightness of the firefly j at $r = 0$, and x_i and x_j are the locations of the fireflies i and j , respectively.

If the firefly j is the brighter one, the value of its attractiveness regulates the movement of the firefly i according to the following equation:

$$x_i = x_i + [\beta_j(r)](x_j - x_i) + \alpha(\text{rand}), \quad (3)$$

where α is a randomization parameter and rand is a uniform random number in the range $[-0.5, 0.5]$. The function of the second term in (3) is to move the firefly i towards the firefly j . The function of the third term in (3) is to move the solution away from a local optimum when such incident takes place.

3. Genetic Algorithm

Genetic algorithm (GA) is an optimization technique inspired by the process of natural selection. GA starts with

an initial population which consists of a number of randomly generated chromosomes. A new population is created from the current one by means of 3 genetic operators: selection, crossover, and mutation. The selection operator stochastically chooses chromosomes to be included in the mating pool; the ones with higher fitness values are more likely to be chosen. Crossover operator selectively chooses some genes from the chromosomes of the parents and combines them into the offspring. Mutation operator randomly changes some genes of the offspring. This evolution process is repeatedly performed until any of the stopping criteria is met. The commonly used stopping criteria are as follows: (i) a predefined number of iterations is reached, (ii) the best solution does not improve for a predefined number of iterations, and (iii) a large percentage of the chromosomes in the population is the same.

4. Biological Inspiration of the Proposed Algorithm

There are over 2,000 species of fireflies around the world, but most of them are found in the tropical zone. They live under the water when they are larvae and on the ground and in the air when they are adults. Fireflies are social animals. They stay in a swarm on tree branches and lay eggs on the ground around the trees. The fascinating thing when observing fireflies is their light flash. The light emitted from their abdomens is a cold light through chemical reactions within their bodies. Fireflies emit flashing light for communication, luring preys, repelling predators, and attracting mates. Note that this research only focuses on the mating behavior of the firefly; therefore, the detailed relation of the flashing light signal to the mate selection process is further discussed in the next paragraph.

In the mating season, female fireflies release pheromones into the air to signal their readiness to mate. The pheromones are carried away in the direction controlled by the wind [13]. Male fireflies follow the pheromone trail and approach the females from a downwind direction [13]. More males are attracted to females who release more pheromones. Males then fly around the trees that the females perch on and flash courtship signal to attract females [14]. Females are more attracted to brighter males and response to those males by flashing their own lights. Then many rounds of mating take place during the night [15]. Males mate until they run out of sperm in their sperm reservoir while females can hold only a certain amount of sperms in their spermatheca. Sperms from fitter males are more likely to be chosen to fertilize a female's eggs; this is due to the following two explanations: (i) females have ways to discard low-quality sperms, including destroying them by their internally produced chemicals [16, 17]; (ii) one male's seminal fluid can incapacitate rival males' sperm within the female reproductive tract [16, 18]. It is nature's way of selective breeding.

5. Our Proposed Algorithm

This section describes a firefly mating algorithm (FMA), which is built on top of the GA. FMA incorporates a new

mating feature, inspired by the mating behavior of fireflies in nature, into GA. This new feature, which will be discussed in Section 5.2, significantly contributes to the capability of the proposed FMA.

FMA consists of three main processes: (i) a male selects a female according to the level of her released pheromone that he senses which changes according to wind speed and direction; (ii) a female selects a male according the light intensity of his flash; and (iii) a male or a female mates repeatedly until he runs out of sperms or her spermatheca is full, producing more able offspring for the next generation. This algorithm proceeds from the first step to the last in the 6 following steps.

5.1. Initialization. An initial population of N fireflies is randomly generated; half are assigned as males and the other half are females. Each firefly consists of d genes (which is equal to the number of variables in the problem). Additionally, the sizes of each female's spermatheca and each male's sperm reservoir, which are real numbers between 0 and 1, are randomly generated. Lastly, the fitness value of each firefly is calculated.

5.2. Selection of Mating Pairs. This step features a new method for determining the mating pairs. Unlike in the GA where mating pairs are selected by using a roulette wheel selection technique, our proposed algorithm introduces a new selection method based on a process of firefly mate selection. First, male fireflies are drawn to a female's location by following the pheromone that the female releases. Second, the female selects males based on their brightness. Finally, mating pairs are formed based on the mutual attraction of each pair. By introducing this concept to the algorithm, it provides a way to overcome the problem of getting stuck in local optima, as is often the case with the roulette wheel selection technique. A detail of this new selection method is described in the following subsection.

5.2.1. Determination of Female Sex Appeal. Female sex appeal is directly proportional to the amount of pheromone released by the female. The pheromone released by the female firefly diffuses downwind and reaches each male firefly in unequal amounts depending on 2 factors: (i) the distance and (ii) the speed and direction of the wind. In this subsection, therefore, the male fireflies determine the concentration level of each female's pheromone reaching them. A highly fit female that is farther away from a male can get selected by the male if her highly concentrated pheromone gets carried along by a high-speed wind towards the male. The concentration level of the female i 's pheromone reaching the male j , P_{ji} , is calculated by using

$$P_{ji} = f_i \times (\vec{D} \cdot \vec{W}), \quad (4)$$

where f_i is the fitness value of a female firefly i , \vec{W} is the wind vector which is randomly generated in each iteration, and \vec{D} is the difference vector between the position vector of a male and that of the female.

5.2.2. Determination of Male Sex Appeal. In this subsection, the female fireflies determine the appeal of each male in their vicinity. Similar to the original FA, the appeal of a male firefly is directly proportional to his brightness. Thus, by imitating (1), the appeal of the male firefly j seen by the female firefly i is calculated according to the following equation:

$$A_{ij} = f_j e^{-\gamma r^2}, \quad (5)$$

where A_{ij} is the appeal of the male firefly j seen by the female firefly i , γ is a constant in the range $[0, 1]$, r is the Euclidean distance between the male and the female (calculated according to (2)), and f_j is the fitness value of the male firefly j .

5.2.3. Calculation of the Mutual Attraction. According to the degree of mutual attraction, a number of males and females are paired together as potential parents. A pair whose mutual attraction value is the highest among all pairs is selected as the first mating pair. The mutual attraction between the female firefly i and the male firefly j , MA_{ij} , is defined as

$$MA_{ij} = A_{ij} + P_{ji}. \quad (6)$$

After each mating, the numbers of sperms in the male's sperm reservoir and in the female's spermatheca are updated. The update procedure is described in the next subsection. Next, the pair with the next highest value is selected if its male member still has some sperms left in the sperm reservoir and the female's sperm bucket has not become full yet.

5.2.4. Update Procedure for Male's Sperm Reservoir and Female's Spermatheca. At the beginning, the number of sperms in each male's sperm reservoir and the size of each female's spermatheca are randomly initialized within the range $[0, 1]$. When a pair of male and female mates, sperms are transferred from the male's sperm reservoir to the female's spermatheca, and the number of sperms in the male's sperm reservoir is reduced. For each mating, the number of sperms a male gives to the female he chose depends on her fitness. The number of sperms that a male can transfer to the female he chose is calculated by the following equation:

$$\eta_{ij} = \delta_j \times f_i, \quad (7)$$

where η_{ij} is the number of sperms transferred to the female at the time of mating, δ_j is the number of sperms in the sperm reservoir, and f_i is the fitness value of a female firefly i . It stands to reason that a fitter female should get more sperms because, then, she would produce more offspring that are fitter than those produced by other females.

After mating, the number of sperms in each male's sperm reservoir and the space of each female's spermatheca are checked. Once a male runs out of sperm or a female's spermatheca is full, he/she will be disqualified from mating in the next round. Equations (8) and (9) are used to update the

number of sperms in a male's sperm reservoir and a female's spermatheca, respectively.

$$\delta_j^{\text{new}} = \delta_j^{\text{old}} - \eta_{ij}, \quad (8)$$

$$\omega_i^{\text{new}} = \omega_i^{\text{old}} + \eta_{ij}. \quad (9)$$

Mating goes on until there is no qualified firefly left to form the mating pair.

5.3. Crossover Operation. When a male and a female mates, some of their genes are crossed over to form two new offspring. One of the two following cases is applied with a specific crossover procedure as follows.

Case 1. If the parents have never been mated before, the 2-point crossover operator (Figure 1) is used to create the offspring. The 2-point crossover operator starts from randomly selecting a start position and an end position in the parent chromosomes to be crossed over, and then the genes between these two positions are crossed over and placed in the same position in the chromosomes of the two offspring, each having the original genes from each parent before the crossover.

Case 2. If either of the parents has been mated before, the n -point crossover operator (Figure 2) is used to create the offspring. The n -point crossover operator starts from randomly selecting $n/2$ pairs of back-to-back start and end positions, then crossing over the genes from the parents between each pair of start and end position. The rest of the genes are retained.

5.4. Mutation Operation. After an incipient offspring is produced from mating, some of its genes are randomly changed (mutated) to new values within the range of the variables as shown in Figure 3. The mutation is performed in order to promote diversity of the population and to help avoid getting stuck in local optima.

5.5. Selection of the Population for the Next Generation. After all offspring are mutated, the N best fireflies out of the combined population of parents and offspring are selected to replace the old population of parents; in effect, only the more effective fireflies are selected to be the population of the next generation. The selection is done by sorting members by their fitness values and then selecting only the members with the higher fitness values that make up the total number of members of the initial population.

5.6. Termination. After the selection of a new population for the next generation, the current iteration is completed. The algorithm then moves on to perform the next iteration until the specified maximum number of iterations is reached or the best solution does not improve for a predefined number of iterations.

TABLE 1: Test functions.

Function ID	Equation	d	S	f_{\min}
f_1	$f(x) = \sum_{i=1}^d x_i^2$	30	$[-100, 100]^d$	0
f_2	$f(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	30	$[-10, 10]^d$	0
f_3	$f(x) = \sum_{i=1}^d ix_i^4$	30	$[-1.28, 1.28]^d$	0
f_4	$f(x) = \sum_{i=1}^d ix_i^4 + \text{random}(0, 1)$	30	$[-1.28, 1.28]^d$	0
f_5	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^d$	0
f_6	$f(x) = \sum_{i=1}^d ([x_i + 0.5])^2$	30	$[-100, 100]^d$	0
f_7	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$	30	$[-5.12, 5.12]^d$	0
f_8	$f(x) = \sum_{i=1}^d [y_i^2 - 10 \cos(2\pi y_i) + 10],$ $y_i = \begin{cases} x_i, & x_i < 0.5 \\ \text{round}(2x_i), & x_i \geq 0.5 \end{cases}$	30	$[-5.12, 5.12]^d$	0
f_9	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	30	$[-32.768, 32.768]^d$	0
f_{10}	$f(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^d$	0
f_{11}	$f(x) = \sum_{i=1}^d x_i ^{i+1}$	30	$[-1, 1]^d$	0
f_{12}	$f(x) = \frac{\pi}{d} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_d - 1)^2 \right\} + \sum_{i=1}^d u(x_i, 10, 100, 4),$ $y_i = 1 + \frac{1}{4} (x_i + 1),$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m; & x_i > a \\ 0; & -a \leq x_i \leq a \\ k(-x_i - a)^m; & x_i < -a \end{cases}$	30	$[-50, 50]^d$	0

TABLE I: Continued.

Function ID	Equation	d	S	f_{\min}
f_{13}	$f(x) = 0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^d (x_i - 1)^2 [1 + \sin^2(3\pi x_i)] + (x_d - 1)^2 [1 + \sin^2(2\pi x_d)] \right\} + \sum_{i=1}^d u(x_i, 5, 100, 4)$	30	$[-50, 50]^d$	0
f_{14}	$f(x) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i-2})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	30	$[-4, 5]^d$	0
f_{15}	$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} \{(w_i - 1)^2 [1 + 10 \sin(\pi w_i + 1)^2]\} + (w_n - 1)^2 [1 + \sin(2\pi w_d)]^2,$ $w_i = 1 + \frac{x_i - 1}{4}$	30	$[-10, 10]^d$	0
f_{16}	$f(x) = \sum_{i=1}^d i x_i^2$	30	$[-10, 10]^d$	0
f_{17}	$f(x) = 0.5 + \frac{\sin^2 \sqrt{\sum_{i=1}^d x_i^2} - 0.5}{(1 + 0.001 \sum_{i=1}^d x_i^2)^2}$	2	$[-100, 100]^d$	0
f_{18}	$f(x) = 418.9829d - \sum_{i=1}^d (x_i \sin \sqrt{ x_i })$	2	$[-500, 500]^d$	0
f_{19}	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^d$	-1.03163
f_{20}	$f(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \cdot [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 - 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^d$	3

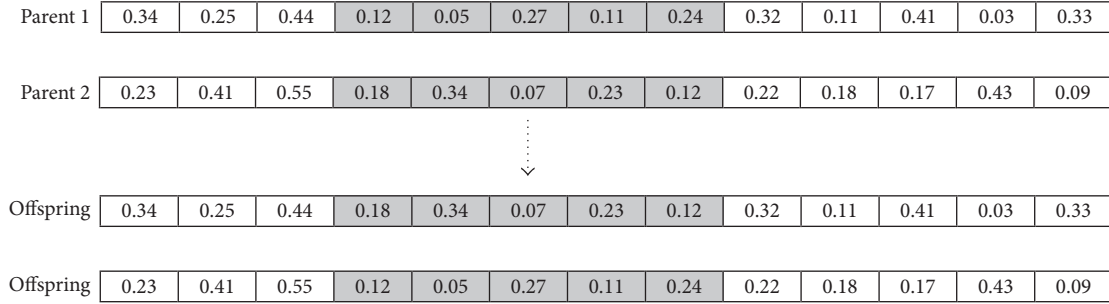


FIGURE 1: Example of 2-point crossover.

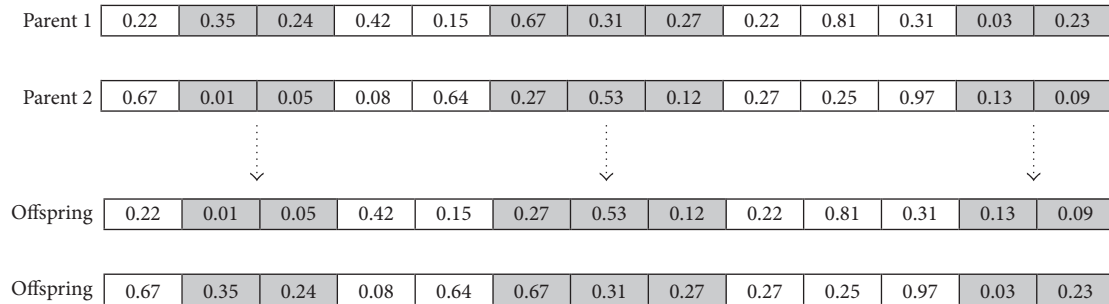


FIGURE 2: Example of n -point crossover.



FIGURE 3: Example of the mutation operation.

6. Description of the Experiments

Our FMA algorithm was performance-tested on Core-i7 computers, and the test results were compared against those of firefly algorithm (FA) and 11 other widely cited algorithms in the literature, namely, ALC-PSO [10], COA [19], MCP SO [20], LWGSODE [21], MPSODDS [22], DFOA [9], SHPSOS [23], LSA [24], MPDPGA [25], DE [26], and GABC [27]. These algorithms were chosen as benchmarks because they are variants of the prominent bioinspired algorithms such as particle swarm optimization, genetic algorithm, differential evolution algorithm, and artificial bee colony algorithm. Moreover, from the literature, their performance was superior to that of their original counterparts. Twenty well-known standard test functions ($f_1 - f_{20}$) were employed to test whether these algorithms would be able to solve all common types of problems. In particular, $f_1 - f_{16}$ are high-dimensional (30 dimensions) that represent computationally complex tasks while $f_{17} - f_{20}$ are two-dimensional multimodal functions. The descriptions of the test functions are shown in Table 1. Table 1 provides the following information about each test function: function ID, equation, dimension (d), search domain (S), and optimum value (f_{\min}). For each test function, five experimental repetitions were performed with different initial populations each time. Since the performance of the metaheuristic algorithms strongly depends on proper

TABLE 2: The parameter setting.

Parameters	Values
Population size	300
Maximum number of iterations	5000
Light absorption	0.2
Crossover rate	1.0
Mutation rate	0.2

selection of system parameters, as most of other intelligent systems, the parameters of FMA are varied to a wide extent in order to get the best out of the algorithm. As a result, the best value for each parameter is shown in Table 2. The performance of FMA was tested in terms of the following 3 aspects: (i) the best solution that an algorithm found, (ii) the average number of iterations to reach the global optimum (in the case that an algorithm has found it), and (iii) the success rate or the number of times that an algorithm was able to find the global optimum divided by the total number of trials.

7. Results and Discussion

The performance of all compared algorithms in finding the optimal solutions of 20 test functions is shown in Table 3.

TABLE 3: Performance comparison between FMA and other algorithms.

	FMA	EA	ALC-PSO	COA	MCPSO	LWGSODE	MPSODDS	DFOA	SHPPOS	LSA	MPPDGA	DE	GABC
f_1	0	0	1.135E-172	0	0	1.13E-8	1.579E-69	3.09E-7	0	1.0622E-19	0	0	3.8838E-17
f_2	0	0	1.121E-98	0	0	2.78E-4	1.96E-42	5.13E-3	0	2.21758E-7	—	0	—
f_3	0	4.64E-7	—	—	—	1.34E-16	—	—	0	—	—	—	—
f_4	0.36400	0.0113	—	3.4479E-5	—	—	—	—	4.58E-4	0.016268885	—	7.73E-7	—
f_5	0	0	3.729E-7	4.1238E-4	6.12	2.66E+1	0.604493	1.76E-5	—	0.560036507	0.00612658	0	1.7632E-3
f_6	0	1.57E-5	0	0	—	—	—	0	0	0	—	0	—
f_7	0	0	7.105E-15	0	7.08E-6	8.77E-7	187.44934	6.95E-7	0	40.7932709	0	0	0
f_8	0.25200	1.04E-9	0	—	—	—	—	0.0013	0	—	—	—	—
f_9	0	0	7.694E-15	8.8818E-16	6.38E-12	6.82E-5	2.23E-14	7.99E-15	4.44E-15	8.73041E-8	0.00395865	4.44E-16	7.9936E-15
f_{10}	0	1.38E-3	0	0	1.91E-14	1.44E-10	0.0275372	6.69E-6	5.84E-4	2.22045E-16	0.05830217	0	1.1102E-16
f_{11}	0	0	—	—	—	—	—	—	—	—	0	—	—
f_{12}	0	2.95E-4	—	1.5705E-32	2.54E-8	—	0.0540318	3.34E-8	2.64E-9	2.23008E-15	—	1.57E-32	—
f_{13}	0	0.0448	1.350E-31	1.3498E-32	—	—	—	8.2E-9	4.28E-8	2.75229E-19	—	1.3E-32	—
f_{14}	7.58E-10	1.85E-10	—	—	—	—	—	—	—	—	—	—	3.3927E-5
f_{15}	0	0.0144	—	—	—	—	—	—	—	—	—	—	5.5574E-17
f_{16}	0	0	—	—	—	—	—	—	0	—	—	—	4.8502E-17
f_{17}	0	0.0907	—	—	—	—	—	—	4.88E-3	—	—	—	—
f_{18}	0	0.0214	—	—	1.32E-3	—	—	—	2.52E-26	—	—	—	1.9843E-4
f_{19}	-1.03124	-1.03163	—	-1.0316	—	—	—	—	—	-1.031628453	—	—	—
f_{20}	3	3	—	3	—	—	—	—	—	3	—	3	—

TABLE 4: Success rates and average numbers of iterations achieved by FMA and FA.

	FMA		FA	
	Average number of iterations	Success rate	Average number of iterations	Success rate
f_1	126.8	100%	47.2	100%
f_2	51.6	100%	229.6	100%
f_3	32	100%	N/A	0%
f_4	N/A	0%	N/A	0%
f_5	214.4	100%	83	100%
f_6	158	100%	N/A	0%
f_7	84.2	100%	120	100%
f_8	N/A	0%	N/A	0%
f_9	62.8	100%	207.8	100%
f_{10}	90.8	100%	N/A	0%
f_{11}	80	100%	162	100%
f_{12}	133.8	100%	N/A	0%
f_{13}	101.5	40%	N/A	0%
f_{14}	N/A	0%	N/A	0%
f_{15}	113	100%	N/A	0%
f_{16}	102.4	100%	284	100%
f_{17}	280.6	100%	N/A	0%
f_{18}	171	100%	N/A	0%
f_{19}	N/A	0%	64.2	100%
f_{20}	67	100%	125.2	100%

The number shown in each cell of the table is the best result obtained by each algorithm. The symbol “—” denotes that the algorithm was not tested on that function. Compared to other algorithms, FMA surpassed them all with respect to the percentage of functions in which the algorithm successfully found the global optima. FMA was able to find the global optima in 16 out of 20 functions tested (shown in bold typeface in Table 3). FA found the global optima of 9 out of 20 test functions. It failed to find the global optima of test functions $f_3, f_4, f_6, f_8, f_{10}, f_{12}, f_{13}, f_{14}, f_{15}, f_{17}$, and f_{18} . ALC-PSO found the global optima of test functions f_6, f_8 , and f_{10} but failed to find the global optima of test functions f_1, f_2, f_5, f_7, f_9 , and f_{13} . COA found the global optima of test functions $f_1, f_2, f_6, f_7, f_{10}$, and f_{20} but failed to find the global optima of test functions $f_4, f_5, f_9, f_{12}, f_{13}$, and f_{19} . MCP SO found the global optima of test functions f_1 and f_2 but failed to find the global optima of test functions $f_5, f_7, f_9, f_{10}, f_{12}$, and f_{18} . LWGSODE and MPSODDS were not able to locate the global optima in any of the functions tested. DFOA successfully found the global optimum of only 1 out of 10 functions tested. SHPSOS found the global optima of test functions $f_1, f_2, f_3, f_6, f_7, f_8$, and f_{16} but failed to find the global optima of test functions $f_4, f_9, f_{10}, f_{12}, f_{13}, f_{17}$, and f_{18} . LSA found the global optima of test functions f_6 and f_{20} but failed to find the global optima of test functions $f_1, f_2, f_4, f_5, f_7, f_9, f_{10}, f_{12}, f_{13}$, and f_{19} . MPDPGA found the global optima of test functions f_1, f_7 , and f_{11} but failed to find the global optima of test functions f_5, f_9 , and f_{10} . DE found the global optima of test functions $f_1, f_2, f_5, f_6, f_7, f_{10}$, and f_{20} but failed to find the global optima of test functions f_4, f_9 ,

f_{12} , and f_{13} . DFOA successfully found the global optimum of only 1 out of 9 functions tested.

Besides being very effective, FMA also has good convergence performance. Table 4 shows the comparative convergence performance of FMA and FA on the following 2 aspects: the average number of iterations to reach the global optimum (in the case that an algorithm has found it) and the success rate (the number of times that an algorithm was able to find the global optimum divided by the total number of trials). “N/A” denotes that the algorithm was unable to find the optimal solution for that function. By examining the results, the following are summaries of the finding. First, FMA achieved 100% success rate for 15 out of 16 functions in which FMA found the global optima. This means that FMA almost always found the global optimum regardless of the difference in the initial population of each experimental trial. Second, as mentioned earlier, FMA was able to find the global optima in 16 out of 20 functions while FA only found 9. Therefore, only eight functions ($f_1, f_2, f_5, f_7, f_9, f_{11}, f_{16}$, and f_{20}) in which both FMA and FA found the global optima have been analyzed. Out of the 8 functions, FMA can achieve the global optima with far fewer iterations than FA for the functions $f_2, f_7, f_9, f_{11}, f_{16}$, and f_{20} , while FA has better convergence rate for the functions f_1 and f_5 .

The distinct advantage of FMA is the rich diversity of offspring that it generates. Two main causes for this kind of diversity are that FMA enables fireflies to perform multiple mating and that the level of female pheromone that a male senses depends not only on the distance between them but also on the wind speed and direction which is randomly

selected for each iteration. As a result, FMA is better able to converge to a global optimum of a function that has a wide and complex search space.

8. Conclusion

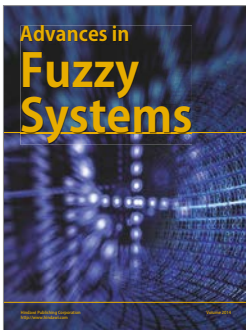
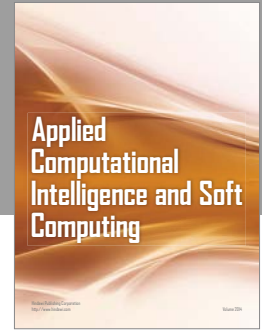
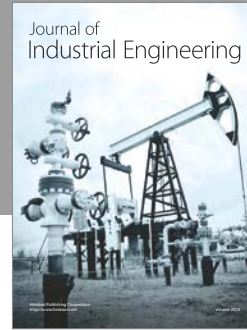
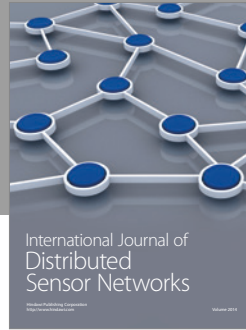
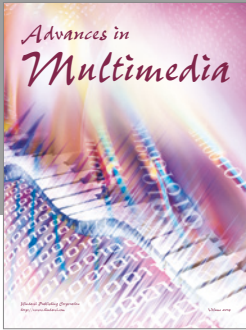
In this paper, a new mating pair selection technique, inspired by the mating behavior of fireflies in nature, is proposed and incorporated into the original firefly algorithm and genetic algorithm. The performance of FMA was tested with sixteen 30-dimensional benchmark functions and four 2-dimensional functions against several widely accepted algorithms. The test results showed that it was able to converge to the global optima in 16 of them. Moreover, compared to the firefly algorithm, FMA required fewer numbers of iterations for most of the functions tested. Because of this initial success, FMA is worth further investigation and application to combinatorial optimization problems.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Berlin, Germany, 2009.
- [2] A. Baykasoğlu and F. B. Ozsoydan, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3712–3725, 2014.
- [3] S. M. Farahani, A. A. Abshouri, B. Nasiri, and M. R. Meybodi, "A gaussian firefly algorithm," *International Journal of Machine Learning and Computing*, vol. 1, no. 5, pp. 448–453, 2011.
- [4] R. M. Rizk-Allah, E. M. Zaki, and A. A. El-Sawy, "Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 224, pp. 473–483, 2013.
- [5] Y.-B. Shin and E. Kita, "Search performance improvement of particle swarm optimization by second best particle information," *Applied Mathematics and Computation*, vol. 246, pp. 346–354, 2014.
- [6] H. Hakli and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Applied Soft Computing Journal*, vol. 23, pp. 333–345, 2014.
- [7] A. Moayedikia, R. Jensen, U. K. Wiil, and R. Forsati, "Weighted bee colony algorithm for discrete optimization problems with application to feature selection," *Engineering Applications of Artificial Intelligence*, vol. 44, article no. 2345, pp. 153–167, 2015.
- [8] W. F. Gao and S. Y. Liu, "A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, pp. 687–697, 2012.
- [9] J. Niu, W. Zhong, Y. Liang, N. Luo, and F. Qian, "Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization," *Knowledge-Based Systems*, vol. 88, pp. 253–263, 2015.
- [10] W.-N. Chen, J. Zhang, Y. Lin et al., "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.
- [11] M. Thakur, "A new genetic algorithm for global optimization of multimodal continuous functions," *Journal of Computational Science*, vol. 5, no. 2, pp. 298–311, 2014.
- [12] X. S. Yang, *Nature-inspired Metaheuristic Algorithm*, Luniver Press, Bristol, UK, 2010.
- [13] R. De Cock and E. Matthysen, "Sexual communication by pheromones in a firefly, *Phosphaenus hemipterus* (Coleoptera: Lampyridae)," *Animal Behaviour*, vol. 70, no. 4, pp. 807–818, 2005.
- [14] K. Demary, C. I. Michaelidis, and S. M. Lewis, "Firefly courtship: Behavioral and morphological predictors of male mating success in *Photinus greeni*," *Ethology*, vol. 112, no. 5, pp. 485–492, 2006.
- [15] A. South and S. M. Lewis, "Determinants of reproductive success across sequential episodes of sexual selection in a firefly," *Proceedings of the Royal Society B: Biological Sciences*, vol. 279, no. 1741, pp. 3201–3208, 2012.
- [16] K. C. Demary, "Sperm storage and viability in *Photinus* fireflies," *Journal of Insect Physiology*, vol. 51, no. 7, pp. 837–841, 2005.
- [17] C. L. Fry and G. S. Wilkinson, "Sperm survival in female stalk-eyed flies depends on seminal fluid and meiotic drive," *Evolution*, vol. 58, no. 7, pp. 1622–1626, 2004.
- [18] C. S. C. Price, K. A. Dyer, and J. A. Coyne, "Sperm competition between *Drosophila* males involves both displacement and incapacitation," *Nature*, vol. 400, no. 6743, pp. 449–452, 1999.
- [19] M. Li, H. Zhao, X. Weng, and T. Han, "Cognitive behavior optimization algorithm for solving optimization problems," *Applied Soft Computing Journal*, vol. 39, pp. 199–222, 2016.
- [20] J. Jie, J. Zhang, H. Zheng, and B. Hou, "Formalized model and analysis of mixed swarm based cooperative particle swarm optimization," *Neurocomputing*, vol. 174, pp. 542–552, 2016.
- [21] H. Cui, J. Feng, J. Guo, and T. Wang, "A novel single multiplicative neuron model trained by an improved glowworm swarm optimization algorithm for time series prediction," *Knowledge-Based Systems*, vol. 88, pp. 195–209, 2015.
- [22] X. Jin, Y. Liang, D. Tian, and F. Zhuang, "Particle swarm optimization using dimension selection methods," *Applied Mathematics and Computation*, vol. 219, no. 10, pp. 5185–5197, 2013.
- [23] F. Zhao, Y. Liu, C. Zhang, and J. Wang, "A self-adaptive harmony PSO search algorithm and its performance analysis," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7436–7455, 2015.
- [24] H. Shareef, A. A. Ibrahim, and A. H. Mutlag, "Lightning search algorithm," *Applied Soft Computing Journal*, vol. 36, pp. 315–333, 2015.
- [25] A. J. Umbarkar, M. S. Joshi, and W.-C. Hong, "Multithreaded parallel dual population genetic algorithm (MPDPGA) for unconstrained function optimizations on multi-core system," *Applied Mathematics and Computation*, vol. 243, pp. 936–949, 2014.
- [26] D. Jitkongchuen and A. Thammano, "A self-adaptive differential evolution algorithm for continuous optimization problems," *Artificial Life and Robotics*, vol. 19, no. 2, pp. 201–208, 2014.
- [27] J. Liu, H. Zhu, Q. Ma, L. Zhang, and H. Xu, "An artificial bee colony algorithm with guide of global & local optima and asynchronous scaling factors for numerical optimization," *Applied Soft Computing Journal*, vol. 37, pp. 608–618, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

