An Improved Firefly Algorithm for Optimization Problems

Amarita Ritthipakdee[1], Arit Thammano[2], Nol Premasathian[3], and Bunyarit Uyyanonvara[4]

***Abstract*— Optimization problem is one of the most difficult and challenging problems that has received considerable attention over the last decade. Researchers have been constantly investigating better ways to solve it. Recently, one optimization technique called firefly algorithm has gained the interest of many researchers. This algorithm is a type of swarm intelligence algorithm based on the reaction of a firefly to the light of other fireflies. In this paper, we propose an improvement on the original firefly algorithm. The proposed algorithm takes into account not only the firefly's reaction to light but also the following contributing factors: firefly's gene exchange, its pheromone, and the impact the wind has on pheromone dispersion. We tested our proposed algorithm against the traditional firefly algorithm and the original genetic algorithm with six standard benchmark functions and found that our algorithm is not only more effective but also faster than the other two algorithms.**

I. Introduction

Optimization problem is one of the most challenging problems in the field of operation research. The goal of the optimization problem is to find the set of variables that results into the optimal value of the objective function, among all those values that satisfy the constraints. Many new types of optimization algorithms have been explored. One of them is a nature-inspired type. Algorithms of this type are such as an ant colony optimization (ACO) algorithm proposed by Marco Dorigo in 1992 which has been successfully applied to scheduling problems. ACO is inspired by the ants' social behavior of finding their food sources and the shortest paths to their colony, marked by their released pheromone [1, 2]. Another example of this type of algorithms is a particle swarm optimization (PSO) algorithm developed by Kennedy and Eberhart in 1995. PSO is based on the swarming behavior of schools of fish and bird in nature. PSO has been successfully applied to a wind energy forecasting problem [1] where wind energy is estimated based on two meta-heuristic attributes of swarm intelligence. A firefly algorithm is yet another example. It is a population-based algorithm inspired by the social behavior of fireflies [3, 4]. Fireflies communicate by flashing their light. Dimmer fireflies are attracted to brighter ones and move towards them to mate [5]. FA is widely used to solve reliability and redundancy problems. A species of firefly called Lampyride also used pheromone to attract their mate [6].

Another well-known nature-inspired algorithm is genetic algorithm (GA). GA is inspired by the process of natural evolution. It starts with a population of chromosomes and effects changes by genetic operators. Three key genetic operators are crossover, mutation, and selection operators [7]. Our algorithm proposed in this paper combines attributes of firefly mating and its pheromone dispersion by the wind with the genetic algorithm. GA is used as the core of our algorithm while the attributes mentioned are used to compose a new selection operator.

[1,2]Computational Intelligence Laboratory, [3]Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520 Thailand (e-mail: [1]s3660104@kmitl.ac.th, [2]arit@it.kmitl.ac.th, and [3]nol@it.kmitl.ac.th).
[4]Sirindhorn International Institute of Technology, Thammasat University, Bangkok, Thailand (e-mail: bunyarit@siit.tu.ac.th).

II. Background

A. Firefly Algorithm

There are three idealized rules incorporated into the original Firefly algorithm (FA) [4]: i) all fireflies are unisex so that a firefly is attracted to all other fireflies; ii) a firefly's attractiveness is proportional to its brightness seen by other fireflies, and so, for any two fireflies, the dimmer firefly is attracted by the brighter one and moves towards it, but if there are no brighter fireflies nearby, a firefly moves randomly; and iii) the brightness of a firefly is proportional to the value of its objective function. According to the above three rules, the degree of attractiveness of a firefly is calculated by the following equation:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (1)$$

where β is the degree of attractiveness of a firefly at a distance r, β_0 is the degree of attractiveness of the firefly at r = 0, r is the distance between any two fireflies, and γ is a light absorption coefficient. The distance r between firefly i and firefly j located at x_i and x_j respectively is calculated as a Euclidean distance:

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

The movement of the dimmer firefly i towards the brighter firefly j in terms of the dimmer one's updated location is determined by the following equation:

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r^2}(x_j - x_i) + \alpha(\text{rand} - \frac{1}{2}) \qquad (3)$$

The third term in (3) is included for the case where there is no brighter firefly than the one being considered and rand is a random number in the range of [0, 1].

B. Genetic Algorithm

Genetic algorithm (GA) is inspired by the process of natural evolution from parents to their offspring. An initial population of possible solutions is randomly selected and then operated on by GA operators. Three key GA's operators are crossover, mutation, and selection operators. The crossover operator exchanges genes between a mating pair and creates an offspring. The mutation operator randomly changes the genes of an offspring, and the selection operator selects a group of offspring to be the next generation of population.

III. Proposed Algorithm

The proposed algorithm uses GA as its core. The new mating pair selection method is introduced to improve the exploration capability of the algorithm. This new selection method is inspired by two natural behaviors of fireflies in attracting mating partners: flashing their lights and releasing the pheromone. The detailed procedure of the proposed algorithm is described as follows:

A. Generation of the Initial Population

In the first step, an initial population of N chromosomes is randomly generated. Then a fitness function is applied to evaluate the fitness of all chromosomes in the population.

B. Mating Pair Selection

This step selects N/2 mating pairs from the population. In a traditional GA, mating pairs are typically selected by using a roulette wheel selection method. On the other hand, in our proposed model, mating pairs are selected based on two natural behaviors of fireflies in attracting mating partners: i) the behavior of using the flashing light and ii) the behavior of using pheromone and its dispersion by the wind.

In nature, female fireflies perch on a tree and release pheromone while male fireflies, attracted to the females' pheromone in the air, circle around them and flash their light. Each female responds to the brightest male she sees by flashing her light synchronously with his. Then mating begins. The following paragraphs discuss how the above behaviors are used to determine the mating pairs.

According to the behavior of using the flashing light to attract mating partners, the degree of attractiveness of chromosome i (or firefly i) is calculated by the following equation:

$$\beta_i = \beta_{i0} e^{-\gamma r^2} \qquad (4)$$

where β_i is the degree of attractiveness of chromosome i, β_{i0} is the fitness value of chromosome i, γ is the light absorption coefficient, and r is a Euclidean distance between chromosome i and chromosome j.

Besides using the flashing light to attract mating partners, some firefly species in nature, such as Lampyridae, use pheromone to attract a mate as well. In this proposed algorithm, not only the attraction by pheromone but also the impacts of the wind, dispersing the pheromone in some directions and lowering its concentration in the air, are taken into account. The concentration of the pheromone of firefly i at any given point is calculated by the following equation:

$$Phe_i = \beta_{i0} - W_{strength} \cos(\theta_1 - \theta_2) e^{-r} \qquad (5)$$

where β_{i0} is the fitness value of firefly i that releases the pheromone, $W_{strength}$ is wind strength in the range of [0, 1], θ_1 is the angle between firefly i and firefly j in the range of [0^o, 360^o], and θ_2 is the angle the wind makes with the x axis in the range of [0^o, 360^o].

By combining the above two behaviors, the average attractiveness between chromosome i and chromosome j is obtained as follows:

$$A_{ij} = \frac{(\beta_i \times Phe_i) + (\beta_j \times Phe_j)}{2} \qquad (6)$$

After the average attractiveness of each pair of chromosomes is calculated, the pair with the highest value is selected as the first mating pair. Then the pairs with the next highest values are selected consecutively until N/2 pairs are attained.

C. Crossover

Crossover exchanges genes between two parent chromosomes and produces two offspring. Each offspring inherits some characteristics of each parent. There are many types of crossover operators. In this paper, only one-point crossover and two-point crossover are being considered. In one-point crossover, a crossover point, which is between the first and the last genes of the parent chromosomes, is randomly selected. Then, all genes beyond that crossover point are exchanged to form two new chromosomes.

In Fig. 1, for example, if the crossover point is 4, the crossover of 01010 and 11001 creates two new offspring which are 01001 and 01110.

	Parents						**Offspring**					
Rand				***4***							***4***	
Parent 1	0	1	0	1	0	Child 1	0	1	0	***0***	***1***
Parent 2	1	1	0	0	1	Child 2	0	1	1	***1***	***0***

Figure 1. One-point crossover.

In two-point crossover, two crossover points are randomly selected. As shown in Fig. 2, each parent chromosome is divided into three segments. The segment of chromosomes of the two parents which are between the two crossover points, genes 2, 3 and 4 in this example, are exchanged.

	Parents						**Offspring**				
Rand	***1***				***5***		***1***				***5***
Parent 1	0	0	1	1	0	Child 1	0	***1***	***0***	***0***	0
Parent 2	1	1	0	0	1	Child 2	1	***0***	***1***	***1***	1

Figure 2. Two-point crossover.

D. Mutation

In order to avoid getting stuck at a local optimum, mutation is applied to each offspring one by one. Mutation effects small random changes to the genes in order to create diversity. In this research, mutation is done by randomly selecting a gene in the chromosome and flipping it over.

E. Selection of the Next Generation

During this step, a selection operator selects the top N chromosomes from a set of parent and offspring chromosomes for progression to the next generation.

F. Termination

When one termination criterion or more is satisfied, the algorithm stops looping and returns the best chromosome in the current population. Otherwise, it goes back to step B.

IV. Results and Discussion

The performance of the proposed model was tested for its performance in finding optimal solutions with six standard benchmark functions [8]: Sphere, Ackley, Levy, Matyas, Booth, and Three-hump camel functions. These six functions are briefly described below:

A. Sphere Function

$$f(\vec{x}) = \sum_{i=1}^{m} x_i^2 \tag{7}$$

where i = 1, 2, …, m. m is the number of dimensions of the problem space. The search space is restricted to $-5.12 \leq x_i \leq 5.12$. The global minimum of this function is equal to zero, attained at $\vec{x} = (0, 0, ..., 0)$.

B. Ackley Function

$$f(\vec{x}) = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}} - e^{\frac{1}{m}\sum_{i=1}^{m}\cos(2\pi x_i)} \tag{8}$$

where i = 1, 2, …, m. m is the number of dimensions of the problem space. The search space is restricted to $-15 \leq x_i \leq 30$. The global minimum of this function is equal to zero, attained at $\vec{x} = (0, 0, ..., 0)$.

C. Levy Function

$$f(\vec{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{m-1}\left[(y_i - 1)^2\left(1 + 10\sin^2(\pi y_i + 1)\right)\right] + (y_m - 1)^2\left(1 + 10\sin^2(2\pi y_m)\right) \tag{9}$$

$$y_i = 1 + \frac{x_i - 1}{4} \tag{10}$$

The search space of this function is $-10 \leq x_i \leq 10$. The global minimum is located at $\vec{x} = (1, 1, ... , 1)$ with a function value of 0.

D. Matyas Function

$$f(\vec{x}) = 0.26\left(x_1^2 + x_2^2\right) - 0.48x_1x_2 \tag{11}$$

The search space of this function is $-10 \leq x_i \leq 10$. The global minimum of this function is equal to zero, attained at $\vec{x} = (0, 0)$.

E. Booth Function

$$f(\vec{x}) = \left(x_1 + 2x_2 - 7\right)^2 + \left(2x_1 + x_2 - 5\right)^2 \tag{12}$$

The search space of this function is $-10 \leq x_i \leq 10$. The global minimum of this function is equal to zero, attained at $\vec{x} = (1, 3)$.

F. Three-hump Camel Function

$$f(\vec{x}) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2 \tag{13}$$

The search space of this function is $-5 \leq x_i \leq 5$. The global minimum of this function is equal to zero, attained at $\vec{x} = (0, 0)$.

The performance of our proposed algorithm was compared with those of the traditional firefly algorithm and genetic algorithm. Table I, II, and III show the values of the parameters used in these experiments. For each set of parameter values, all three algorithms were run 10 times. Each run started with a different initial population and stopped when one of the termination criteria was met. The performances of these three algorithms were measured in terms of their ability to reach the optimal solution.

Table IV, V, and VI present the performances of our proposed algorithm, the firefly algorithm, and the genetic algorithm in finding the optimal solutions of the four benchmark functions. "Average iteration" is the mean of the number of iterations executed to obtain the optimal solution.

TABLE I. PARAMETERS USED IN THE PROPOSED ALGORITHM

Parameters	Values
Number of Populations (N)	10, 20, 30, 40
Light Absorption Coefficient (γ)	2
Crossover Rate	1
Mutation Rate	0.2
Maximum Number of Generations	1,000
Wind Strength	[0, 1]
Wind angle (θ_2)	[0^o, 360^o]

TABLE II. PARAMETERS USED IN FIREFLY ALGORITHM

Parameters	Values
Number of Populations (N)	10, 20, 30, 40
Light Absorption Coefficient (γ)	2
α	0.5
Maximum Number of Generations	1,000

TABLE III. PARAMETERS USED IN GENETIC ALGORITHM

Parameters	Values
Number of Populations (N)	10, 20, 30, 40
Crossover Rate	1
Mutation Rate	0.2
Maximum Number of Generations	1,000

"Success rate" is the number of times the algorithm successfully located the global optimum out of 100 trials. "Average time" is the mean of the length of time spent to obtain the optimal solution.

The experimental results show that the proposed algorithm outperforms both the original firefly algorithm and the original genetic algorithm in terms of the success rate. Moreover, the average numbers of iterations to obtain optimal solutions required by the proposed algorithm are lower than those of the other two algorithms.

TABLE IV. EXPERIMENTAL RESULTS OF THE PROPOSED ALGORITHM

Function	N	One-point crossover			Two-point crossover		
		Average Iteration	Success Rate (%)	Average Time (Sec.)	Average Iteration	Success Rate (%)	Average Time (Sec.)
Sphere	10	44.50	100	0.140136	56.60	100	0.206702
	20	44.40	100	0.362126	42.90	100	0.304669
	30	40.50	100	1.003501	34.00	100	0.858094
	40	34.10	100	0.963845	36.00	100	1.359827
Ackley	10	124.50	20	2.714305	106.71	70	2.688368
	20	90.50	60	0.633476	98.83	60	0.595218
	30	71.00	70	1.257834	88.89	90	1.755377
	40	79.33	90	3.038048	71.88	80	3.402800
Levy	10	84.30	100	0.478891	63.30	100	0.307523
	20	57.50	100	0.560573	50.80	100	0.354617
	30	43.60	100	0.789364	41.30	100	0.807373
	40	36.20	100	1.355643	36.40	100	1.282797
Matyas	10	503.00	40	2.274087	516.67	30	2.798571
	20	384.60	50	8.162268	263.50	50	2.673472
	30	484.67	75	7.951059	435.50	60	2.574411
	40	289.60	80	6.361230	214.40	70	7.451153
Booth	10	87.50	20	24.946641	512.50	20	26.850712
	20	399	10	85.726177	89.50	20	76.832269
	30	432.50	20	98.546332	478.83	60	206.227708
	40	223.63	80	106.712956	355.50	40	185.375549
Three-hump camel	10	152.38	80	10.347262	167.57	70	14.057109
	20	98.88	80	28.192548	122.40	50	57.851137
	30	99.75	80	59.861841	89.38	80	54.061484
	40	77.22	90	62.272257	70.13	80	96.478422

TABLE V. EXPERIMENTAL RESULTS OF THE TRADITIONAL FIREFLY ALGORITHM

Function	N	Average Iteration	Success Rate (%)	Average Time (Sec.)
Sphere	10	632.50	20	0.775529
	20	413.00	10	1.319305
	30	531.50	20	1.083772
	40	829.10	100	0.646800
Ackley	10	324.20	10	0.345611
	20	435.60	30	0.392470
	30	580.20	40	0.482667
	40	724.70	70	0.895902
Levy	10	149.00	10	0.902156
	20	172.00	30	0.888335
	30	133.33	30	0.917366
	40	147.28	70	0.506923
Matyas	10	115.00	10	0.546960
	20	198.00	10	0.577840
	30	143.32	40	0.340200
	40	184.71	60	0.284732
Booth	10	N/A	0	N/A
	20	N/A	0	N/A
	30	N/A	0	N/A
	40	N/A	0	N/A
Three-hump camel	10	29.42	70	0.342584
	20	13.50	80	0.374392
	30	19.50	90	0.318742
	40	25.50	100	0.031985

TABLE VI. EXPERIMENTAL RESULTS OF THE TRADITIONAL GENETIC ALGORITHM

Function	N	Average Iteration	Success Rate (%)	Average Time (Sec.)
Sphere	10	237.60	30	0.489192
	20	462.00	40	2.456908
	30	389.67	60	1.579418
	40	551.00	70	3.363200
Ackley	10	144.00	10	1.419394
	20	160.54	10	2.249834
	30	187.00	30	2.395860
	40	164.00	40	2.270024
Levy	10	304.00	10	1.453955
	20	326.00	10	3.186401
	30	359.00	10	3.640327
	40	801.00	10	2.422948
Matyas	10	153.40	100	0.285534
	20	253.50	100	0.655994
	30	428.14	70	1.705949
	40	699.33	60	3.078126
Booth	10	N/A	0	N/A
	20	N/A	0	N/A
	30	N/A	0	N/A
	40	N/A	0	N/A
Three-hump camel	10	N/A	0	N/A
	20	N/A	0	N/A
	30	N/A	0	N/A
	40	N/A	0	N/A

V. CONCLUSION

This paper proposed a new optimization algorithm motivated by the nature of firefly mating. In the proposed algorithm, GA is used as the core of the algorithm. The natural behaviors of fireflies in attracting mating partners are used in designing a new selection operator. The performance of our proposed algorithm was tested against those of the traditional firefly algorithm and the original genetic algorithm with six standard benchmark functions: Sphere, Ackley, Levy, Matyas, Booth, and Three-hump camel functions. Salient test results are as follows: i) the success rates of our proposed algorithm were higher than those of the other two algorithms, especially on tests with the Sphere and Levy functions where the success rates were 100%; ii) the average numbers of iterations to obtain optimal solutions required by our proposed algorithm were lower than those of the other two algorithms on all six benchmark functions; that is, our algorithm converged to the optimal solutions faster.

References

[1] R. Rahmani, R. Yusof, M. Seyedmahmoudian, and S. Mekhilef, "Hybrid technique of ant colony and particle swarm optimization for short term wind energy forecasting," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 123, pp. 163-170, December 2013.

[2] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," *Proceedings of the IEEE Congress on Evolutionary Computation*, 1999, pp. 1470-1477.

[3] B. Bhushan and S. S. Pillai, "Particle swarm optimization and firefly algorithm: performance analysis," *Proceedings of the 3rd IEEE International Advance Computing Conference (IACC)*, 2013, pp. 746 – 751.

[4] P. R. Srivatsava, B. Mallikarjun, and X. –S. Yang, "Optimal test sequence generation using firefly algorithm," *Swarm and Evolutionary Computation*, vol. 8, pp. 44-53, February 2013.

[5] A. H. Gandomi, X. -S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, issue 1, pp. 89-98, 2013.

[6] R. De Cock and E. Matthysen, "Sexual communication by pheromones in a firefly, Phosphaenus hemipterus (Coleoptera: Lampyridae)," *Animal Behaviour*, vol. 70, issue 4, pp. 807-818, 2005.

[7] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 57-69, 2014.

[8] A. V. Levy and A. Montalvo, "The tunnelling algorithm for the global Minimization of functions," *SIAM Journal of Scientific and Statistical Computing*, vol. 6, pp. 15-29, 1985.