



Improving firefly-based Multi-Objective Optimization based on Attraction Law and Crowding Distance

Farid shayesteh, Reihaneh Kardehi Moghaddam *

Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract Multi-objective optimization problems are so designed that they simultaneously minimize several objectives functions (which are sometimes contradictory). In most cases, the objectives are in conflict with each other such that optimization of one objective does not lead to the optimization of another ones. Therefore, we should achieve a certain balance of goals to solve these problems, which usually requires the application of an intelligent method. In this regard, use of meta-heuristic algorithms will be associated with resolved problems. In this paper, we propose a new multi-objective firefly optimization method which is designed based on the law of attraction and crowding distance. The proposed methods efficiency has been evaluated by three valid test functions containing convex, nonconvex and multi discontinuous convex Pareto fronts. Simulation results confirm the significant accuracy of proposed method in defining the Pareto front for all three test functions. In addition, the simulation results indicates that proposed algorithm has higher accuracy and greater convergence speed, compared to other well known multi-objective algorithms such as non-dominated sorting genetic algorithm, Bees algorithm, Differential Evolution algorithm and Strong Pareto Evolutionary Algorithm.

Keywords Multi-objective Optimization, Crowding Distance, Firefly Algorithm, Law of Attraction, Pareto Front.

AMS 2010 subject classifications 49M37.

DOI: 10.19139/soic-2310-5070-382

1. Introduction

Most optimization problems in engineering science are non-linear and have many restrictions. For this reasons, finding optimum solutions to solve these problems requires strong, efficient and, yet, simple algorithms [1]. Some of the so far discussed optimization problems have been single-objective. Such problems are considered as a scalar optimization problem because the objective function always reaches a global optimal value or a scalar value [2]. However, in the real world, multifunctional objectives should be simultaneously optimized, which is very complex and difficult and defined as multi-objective optimization [4, 5]. Unlike a single-objective optimization problem, which necessarily minimizes an optimal solution, a multi-objective optimization problem follows and minimizes several objectives and most of the goals may be in conflict with each other [3]. When examining a multi-objective optimization problem, we do not insist on an answer. We are looking for different scenarios, meaning that we expect multi-objective problem solving to obtain at least two or more different scenarios. Therefore, there is a level of answers and we will not have answers better than this level, which is called Pareto front. The aim of solving multi-objective optimization problems is providing a set of undefeated answers to a problem [4].

Various methods have been developed for solving multi-objective optimization algorithms; for example, the methods that convert a multi-objective problem into a single-objective problem, which include the methods of weighted sum, goal programming, goal attainment and ϵ - constraint. All of these methods, which are not very

*Correspondence to: Reihaneh Kardehi Moghaddam (Email:R.K_moghaddam@mshdiau.ac.ir). Department of Electrical Engineering, Mashhad branch, Islamic Azad University, Mashhad, Iran.

accurate, should be implemented multiple times in order to create a Pareto front [1, 3]. Another group includes the methods that solve the problem as it is, such as evolutionary algorithms. These methods are all intelligent and can solve the problem with the least information possible. Total Pareto front points are obtained in these methods, using evolutionary algorithms [5]. In this regard, one of the powerful multi-objective algorithms, which is the founder of population-based multi-objective algorithms, is the multi-objective genetic algorithm [6]. Moreover, multi-objective differential evolution algorithms are extremely powerful and suitable for solving optimization problems [7]. This algorithm was developed and presented as a multi-objective adaptive differential algorithm in 2013 [8]. Most of the meta-heuristic algorithms are based on collective intelligence, and multi-objective particle swarm optimization algorithm (MOPSO) can be a good example in this regard. This algorithm is inspired by the group behavior of birds and fish to find food [9, 10]. Another algorithm based on collective intelligence is the firefly algorithm (FA), which is a meta-heuristic algorithm inspired by nature. This algorithm was provided by Yang in 2010, originated from the natural behavior of fireflies of light emission. Although this algorithm is very similar to other swarm intelligence algorithms and its implementation is convenient, this method is very effective and can better solve optimization problems, compared to other conventional algorithms [11]. Next, Yang presented multi-objective firefly algorithm in 2013 through developing the firefly algorithm and creating Pareto front. This algorithm has been implemented based on weighted sum model. In this algorithm, an objective function is defined for each firefly, and each firefly is evaluated in relation to other fireflies on the basis of this objective function. Less attractive fireflies are attracted by more attractive fireflies. If no non-dominated answer is found, an optimal answer is obtained using weighted sum method and fireflies are attracted by this point. At the end of the algorithm, a set of non-dominated answers is obtained in the form of optimal Pareto front [12]. During recent years many powerful algorithms for solving multi-objective problems have been proposed. In [14] a new multi-objective algorithm inspired from the navigation of grass hopper swarms in nature, is introduced. Grasshopper Optimization Algorithm (GOA) includes benefits such as effective exploration and fast convergence. The special adaptive mechanism in this algorithm smoothly balances exploration and exploitation. These characteristics make the GOA algorithm potentially able to cope with the difficulties of a multi-objective search space and outperform other techniques. In [15] a multi-objective version of the recently proposed Ant Lion Optimizer (ALO) called Multi-Objective Ant Lion Optimizer (MOALO) is innovated. A repository is first employed to store non-dominated Pareto optimal solutions obtained so far. Solutions are then chosen from this repository using a roulette wheel mechanism based on the coverage of solutions as antlions to guide ants towards promising regions of multi-objective search spaces. This algorithm mimics the hunting mechanism of antlions and the interaction of their favorite prey, ants, with them. Similar to other population-based algorithms, ALO approximates the optimal solutions for optimization problems with employing a set of random solutions. This set is improved based on the principles inspired from the interaction between antlions and ants. In [16] a modified version of MOABC, named evolutionary multi-objective ABC (EMOABC) is also proposed which incorporates the characteristics of simulated evolution (SE) algorithm for improved local search. The effect of control parameters of MOABC is investigated. Comparison of EMOABC with MOABC and the standard ABC (SABC) illustrates the better performance of this algorithm. [17], proposes an efficient decomposition-based archiving approach (DAA) inspired from the decomposition strategy for dealing with multi-objective optimization. In DAA, the whole objective space is uniformly divided into a number of subspaces according to a set of weight vectors. At each generation, only one non-dominated solution lying in a subspace is chosen to be used for updating the external archive in consideration of its diversity. At each generation, only non-dominated solutions that are close to the weight vectors are to be saved in the archive. In [18], a new non-dominated sorting algorithm, termed Hierarchical Non-Dominated Sorting (HNDS) is introduced. Unlike other non-dominated sorting algorithms, HNDS does not need comparing each solution with all others before determining its Pareto rank; it first sorts all solutions by their first objective values in ascending order, which means that the first individual is a non-dominated solution and can never be dominated by any solutions further down the list. In the above works performed in the past two years, new and powerful multi-purpose optimization methods were used for effective Pareto optimal front enhancement and increase of the accuracy and speed of convergence. However, our proposed algorithm is able to find the Pareto optimal front in any shape with a very precise approximation since it used the law of attraction related to the firefly algorithm and a very powerful and efficient criterion called the crowding distance. In this algorithm, all responses are uniformly distributed on all Pareto optimal fronts. Therefore, the convergence

accuracy and speed of our proposed algorithm was higher, compared to the mentioned algorithms. Simulation results on test functions and comparison to other efficient multi-purpose algorithms approves the efficiency of the proposed algorithm. This method is suitable for solving very complex engineering problems. Therefore in this article, a new multi-objective firefly-based optimization algorithm is proposed which uses the law of attraction and crowding distance. This algorithm is assessed by three different test functions to verify the capability of defining convex, nonconvex and discontinuous Pareto fronts. The ultimately comparison of proposed algorithm to other multi-objective algorithms shows the impressive improvement in sense of accuracy and convergence speed. The second section of the article deals with definitions and basic principles. In the third section, a brief introduction of the proposed method and also algorithm parameters setting are provided. The fourth section is allocated to the simulation results of the proposed method and comparison of the results with other evolutionary algorithms. Finally, the last section provides the conclusion.

1.1. Definitions and Basic Principles

Before introducing the new multi-objective optimization algorithm, which includes quality and regularity together, we define the practical concepts, as follows:

Definition 1: non-dominated solution

A solution is called non-dominated or Pareto-Optimal if none of the objective functions could obtain better values without regarding some of the other objective values.

Definition 2: domination

Considering two components x and y , x dominates y when the following two conditions are established: a) $\forall i: x_i \leq y_i$, for all x components, y component is not better than x in any respect.

b) $\exists i_0: x_{i_0} < y_{i_0}$, which means that x is strictly better than y at least in one respect.

Definition 3: Pareto front

As a set of non-dominated solutions, which are selected as optimal. Considering to figure 1, Pareto front includes answers among the possible answers of the problem, which cannot be dominated by any other answer. The purpose of solving a multi-objective optimization problem is to find a set of non-dominated answers to that problem.

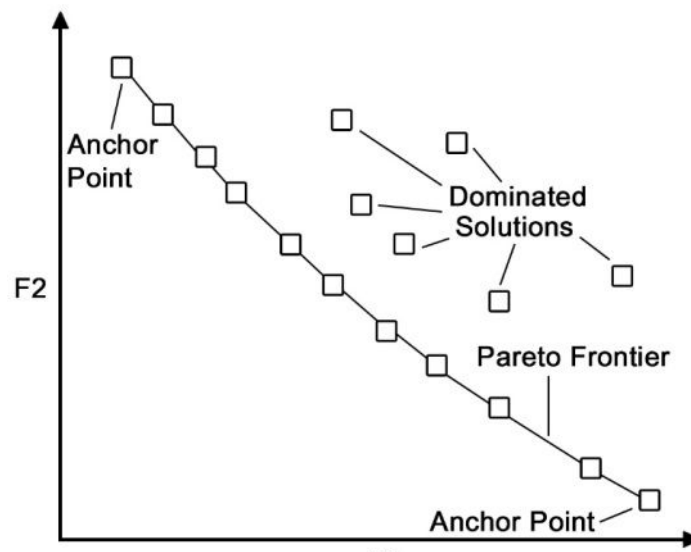


Figure 1. Pareto front formed for two objective functions

In the sequel, the crowding distance will be defined. It is a mathematical measure and is used to obtain a solution for assessing a state in which two components do not defeat each other and are not superior to each other. Crowding distance of each particle to the next member, previous member, first member and last member is determined and its greater value is better due to the covering of a larger area and helping the diversity of answers. Therefore, crowding

distance determines which particle has more importance for us, mathematically defined, as follows: Definition 4: crowding distance Consider two objective functions that should be optimized together, and also let us consider N particles indexed by $i=1, \dots, N$ as indicated in Figure 2. Crowding distance for the particle i is calculated by the following equation with regard to Figure 2.

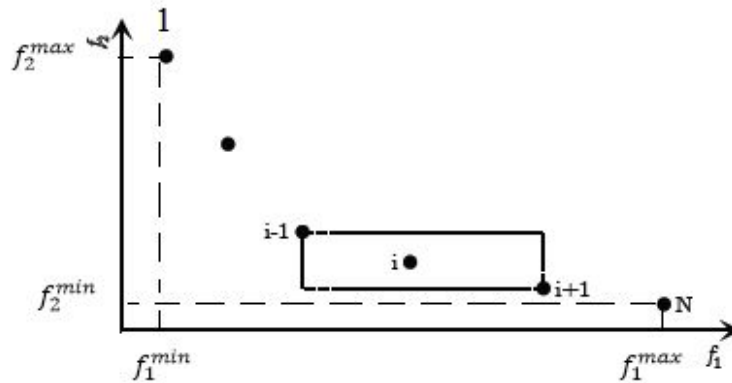


Figure 2. An example of calculating the crowding distance of the particle i

$$d_i^1 = \|f_1^{i+1} - f_1^{i-1}\| / \|f_1^{\max} - f_1^{\min}\|, \quad (1)$$

$$d_i^2 = \|f_2^{i+1} - f_2^{i-1}\| / \|f_2^{\max} - f_2^{\min}\|, \quad (2)$$

$$d = d_i^1 + d_i^2, \quad (3)$$

In general, crowding distance is calculated from the following equation:

$$d_i = d_i^1 + d_i^2 + \dots + d_i^m = \sum_{j=1}^m d_i^j, \quad (4)$$

$$d_i^j = \|f_j^{\text{pre}} - f_j^{\text{next}}\| / \|f_j^{\max} - f_j^{\min}\|, \quad (5)$$

d_i^j is the crowding distance related to the i th answer from the j th objective and d_i is the crowding distance of the i th particle [13].

2. Improved Firefly-based Multi-objective Optimization Algorithm

Since the proposed multi-objective algorithm is the modified form of the single-objective firefly algorithm, we first mentioned to the single-objective firefly algorithm.

2.1. Firefly Algorithm

The firefly algorithm has been designed based on the optical behavior patterns of fireflies. Accordingly, this algorithm follows the three laws below: First law: Fireflies are unisexual and gender plays no role in attracting them. Second law: Attractiveness of each firefly is proportional to its brightness. For two fireflies, the firefly with less light is attracted by other fireflies. The amount of attractiveness is proportional to brightness and has an inverse

relationship with distance. By increasing the distance between two fireflies, the attractiveness rate decreases. If no firefly is brighter than the others, that particular firefly randomly selects the movement towards another firefly. Third law: The brightness rate of each firefly is specified by determining the objective function value, which should be optimized. For this optimization method, brightness can be simply proportional to the objective function value. As the light intensity and attractiveness affect the movement of fireflies in this algorithm, we should define other parameters. For simplicity, we can always assume that attractiveness of a firefly, which is determined by its brightness, is specified by an objective function. However, the attractiveness rate is relative and should be regarded as an observer among other fireflies. Furthermore, the attractiveness rate will vary based on the distance between two fireflies. Attractiveness of a firefly is obtained from the following equation:

$$\beta = \beta_0 e^{-\gamma r^2}, \quad (6)$$

In which β_0 is the initial attractiveness in $r = 0$; γ is the light absorption coefficient and r is the distance between two fireflies. The distance between the two fireflies i and j in the situation x_i and x_j is obtained from the equation below:

$$r_{ij} = \|X_i - X_j\|; \quad (7)$$

However, r can be calculated using other distance criteria. Movement of the firefly i , which is attracted by the firefly j , is calculated from the following equation:

$$X_i = X_i + \beta_0 e^{-\gamma r^2} (X_j - X_i) + \alpha \epsilon_i \quad (8)$$

In the above equation, the next situation of i th firefly is calculated from the sum of three terms; the first term (x_i) is the current situation of the firefly; the second term is related to the law of attraction, the firefly with less light moves towards the firefly with more brightness; and the third term is the random movement of the firefly in cases when there is no superiority between two fireflies in terms of light absorption. γ parameter specifies the attractiveness changes, and its value determines the convergence speed and behavior of the firefly algorithm. ϵ_i component can have uniform distribution or Gaussian distribution, which follows a longer route in the search space. Its function is similar to a mutation in genetic algorithm. Equation three is named as the absorption formula[19].

2.2. The proposed Multi-objective Firefly Algorithm

In the proposed multi-objective optimization algorithm, undefeated answers are obtained using the condition of dominated and non-dominated solutions, the law of attraction of the firefly algorithm. In conditions where the two components (fireflies) do not defeat each other and have no superiority to one another, a secondary measure called the crowding distance is used for evaluation. Moreover, the best answers are considered as the approximate Pareto front by arranging the answers obtained.

The process of the multi-objective firefly algorithm is defined, as follows:

First step: defining the objective functions $f_1(x), f_2(x), \dots, f_M(x)$, considering to $x = (x_1, x_2, \dots, x_n)^T$.

Second step: setting the parameters and randomly initializing the population of P fireflies with uniform distribution.

Third step: calculating the objective function corresponding to each firefly.

Fourth step:

For each firefly like i ,

for each firefly like j ,

if i is not equal to j :

If PF_j dominated PF_i ,

The firefly i is attracted by the firefly j using the law of attraction (equation eight).

Fifth step: If some answers do not defeat each other (no non-dominated answer was found), the crowding distance should be calculated for each answer with regard to equation five.

Arranging the new population based on the selection of the firefly with greater crowding distance. Controlling to see if the new population is not outside the answer space.

Sixth step: combining the answers obtained from the absorption method and crowding distance.

Seventh step: returning to fourth step in case of not meeting the stop condition.

Eighth step: Selection of P non-defeated answers obtained in the form of correct Pareto front.

Ninth step: end

Flowchart of the proposed multi-objective algorithm for two objective functions has been provided in Figure 3.

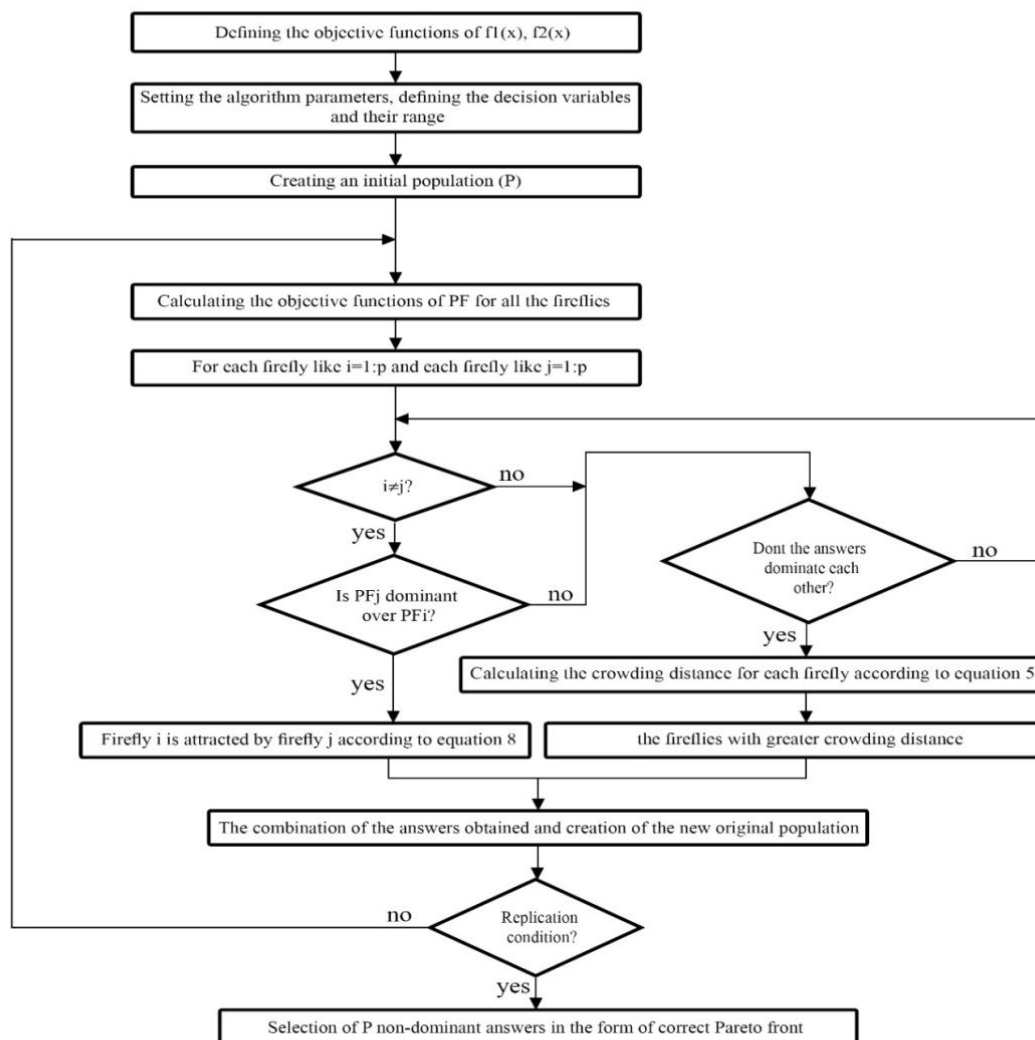


Figure 3. Flowchart of the proposed firefly algorithm

Table 1. DESCRIPTION OF THE TEST FUNCTIONS OF ZDT FAMILY

Function name	Objective functions	Reference
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{x_1/g(x)})$	[22]
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)[(1 - x_1/g(x))^2]$	[22]
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - (x_1/g(x)) \sin(10\pi x_1)]$	[23]
Note	$g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n - 1)$	

2.3. Algorithm Parameters Setting:

Each collective intelligence algorithm depends on its parameters. Therefore, setting the algorithm parameters is extremely crucial. Parameters of the firefly algorithm include β_0, γ is considered between 0.1 and 20.

In this work, we have $\beta_0 = 1, \gamma = 1, \alpha_0 = 0.25$ for all the simulations.

The algorithm termination condition can be provided in different ways, the most important of which is reaching an acceptable level of answers, limited run time or limited iteration number. In this work, we use iteration number as the stop condition, which means that after n times of algorithm implementation, if the number of iterations reaches n , the algorithm stops. Algorithm execution speed becomes slower by increasing the number of iterations, whereas the algorithm accuracy increases. Another important parameter that has a great impact on the accuracy and time of the algorithm implementation is the number of the initial population size. Similar to the iteration condition, increasing the population size will be associated with higher algorithm accuracy and, at the same time, lower execution speed. In this work, the iteration number is equal to 100 and during all simulations, the number of the initial population has been considered to be 500.

3. Numerical Results

The proposed multi-objective firefly algorithm was implemented in MATLAB software and its efficiency was evaluated by a set of multi-objective test functions.

3.1. Multi-objective Test Functions

When a new algorithm is proposed, one of the fundamental issues is using a standard method to prove its validity and effectiveness. To this end, numerous test functions have been introduced for single-objective and multi-objective optimization [20]. In this paper, three quite different test functions with convex, non-convex and discontinuous Pareto fronts were used, which were selected from the ZDT family due to their confirmed functions for evaluating the performance of multi-objective optimization methods and finding the Pareto optimal sets. All the test functions of ZDT family consist of two objective functions [21], defined as follows:

ZDT1 function, which has an optimal convex Pareto front

ZDT2 function, which has an optimal non-convex Pareto front

ZDT3 function, which adds a distinctive characteristic for the front. Optimal Pareto front is composed of several discontinuous convex parts and the sinusoidal characteristic of this function has caused this discontinuity but not in all the parameter space. Full description of ZDT1, ZDT2 and ZDT3 functions has been presented in Table 1.

In all three ZDT1, ZDT2 and ZDT3 functions, x_i was and the range of each variable was between 0 and 1 [22, 23].

The optimal Pareto front of all the functions in Table 1 when $g=1$ is calculated. The Pareto front obtained for the test function ZDT1 by the proposed algorithm and its comparison with the optimal Pareto front with the population size of 500 and iteration number of 100 are provided in Figure 4.

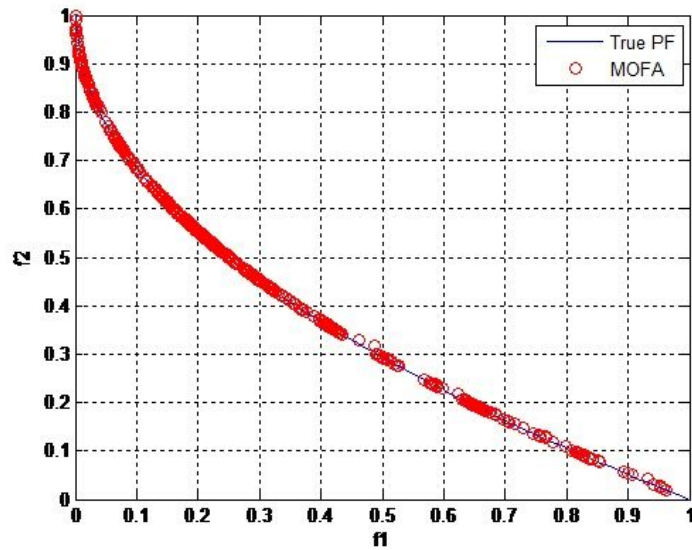


Figure 4. Pareto front of the test function ZDT1 using true PF and MOFA

With regard to Figure 4, it is observed that the front Pareto estimated by the proposed algorithm has well covered all over the optimal Pareto of ZDT1 test function. The Pareto front obtained for the test function ZDT2 by the proposed algorithm and its comparison with the optimal Pareto front with the population size of 500 and iteration number of 100 are provided in Figure 5:

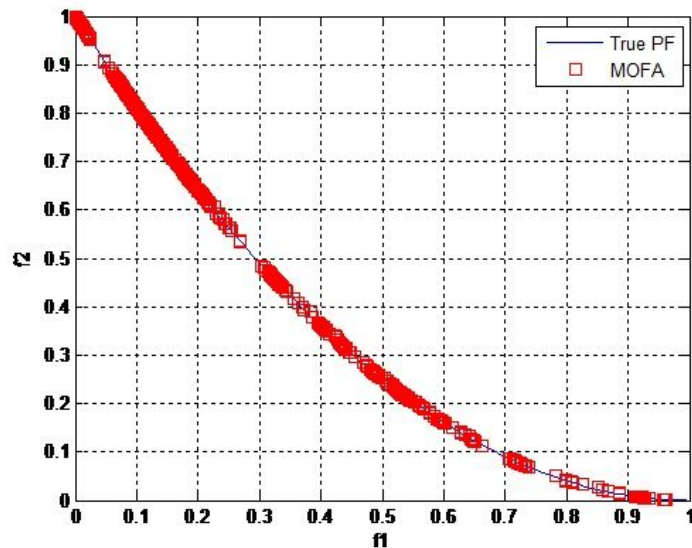


Figure 5. Pareto front of the test function ZDT2

As it is illustrated in Figure 5, it is observed that the front Pareto estimated by the proposed algorithm has well covered all over the optimal Pareto of ZDT2 test function. The Pareto front obtained for the test function ZDT3

by the proposed algorithm and its comparison with the optimal Pareto front with the population size of 500 and iteration number of 100 are provided in Figure 6.

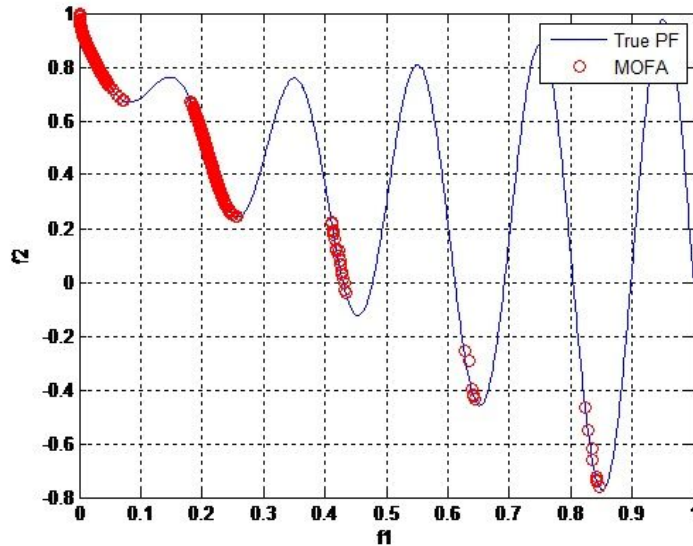


Figure 6. Pareto front of the test function ZDT2

With respect to Figure 6, we come to the conclusion that the front Pareto estimated by the new MOFA algorithm has well covered all over the optimal Pareto of ZDT3 test function. In figures 4-6, the horizontal axis of the objective function is f1 and horizontal axis of the objective function is f2.

3.2. Study of Error Criterion and Convergence Rate

In this section, the distance or error between the estimated Pareto front and optimal Pareto front, denoted by PF estimation error, is defined by the following equation:

$$E_f = \|PF^e - PF^t\|^2 = \sum_{j=1}^N (PF_j^e - PF_j^t)^2; \tag{9}$$

In equation 9, N is the number of pareto points; PF^e is the Pareto front obtained by the algorithm and PF^t is the optimal Pareto front.

Convergence property can be used to assess the algorithm accuracy. Therefore, it is better to use relative measurement according to the following equation:

$$D_g = \frac{1}{N} \sqrt{E_f}; \tag{10}$$

In which, D_g is the convergence rate [3, 12]. Figures 7,8,9 revealed the convergence rate of the proposed multi-objective algorithm with the test functions ZDT1, ZDT2 and ZDT3 in each iteration process. As it is observed in Figure 7, the convergence rate reached its minimum amount in the iteration of about 160, which was indicative of the high accuracy of the proposed algorithm. With regard to the Figure 8, the convergence rate reached its minimum amount in the iteration of about 120, which indicates the efficiency of proposed MOFA algorithm. Figure 9 indicated the fact that the convergence rate reached its minimum amount in the iteration of about 260, which revealed the high accuracy of the proposed method. Considering figures 1-9, it can be concluded that the estimated Pareto front covered all over the optimal Pareto front and quickly converged. The amount of PF estimation error for all test functions are provided in Table 2.

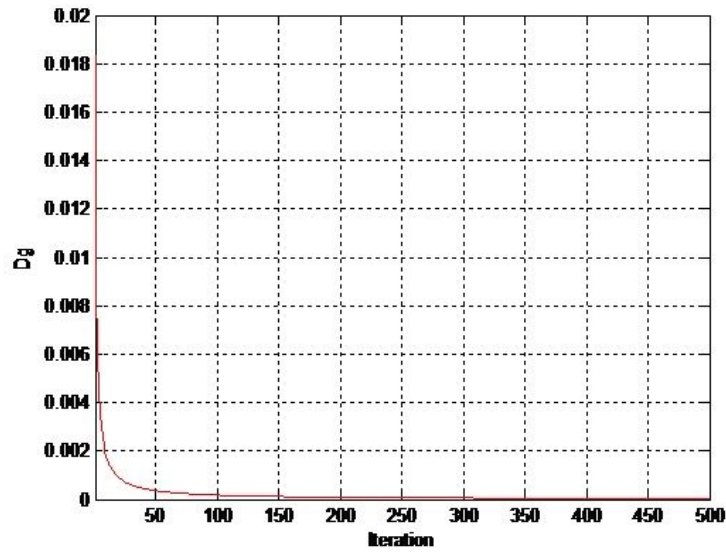


Figure 7. Convergence rate (vertical axis) of the test function ZDT1 for 500 iterations

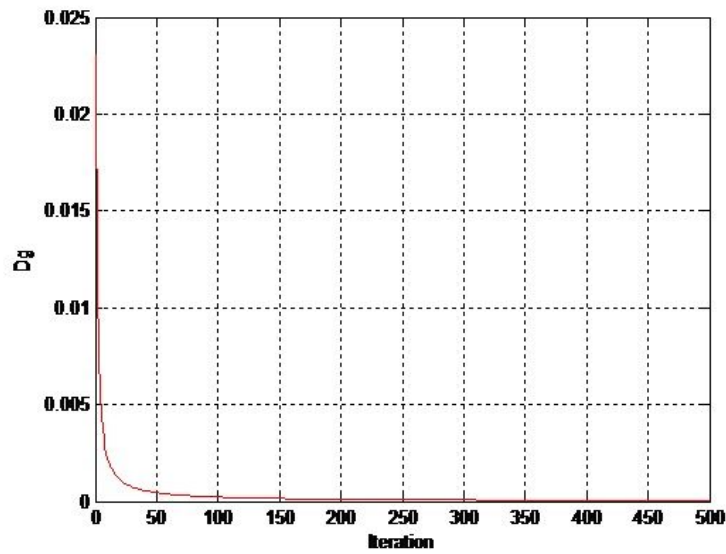


Figure 8. Convergence rate (vertical axis) of the test function ZDT2 for 500 iterations

Table 2. THE AMOUNT OF CONVERGENCE DISTANCE FOR THE POPULATION SIZE OF 500 PEOPLE

Test functions	Convergence rate with 100 iterations	Convergence rate with 500 iterations
ZDT1	2.09e-6	1.03e-5
ZDT2	2.34e-6	1.45e-5
ZDT3	2.71e-6	1.98e-5

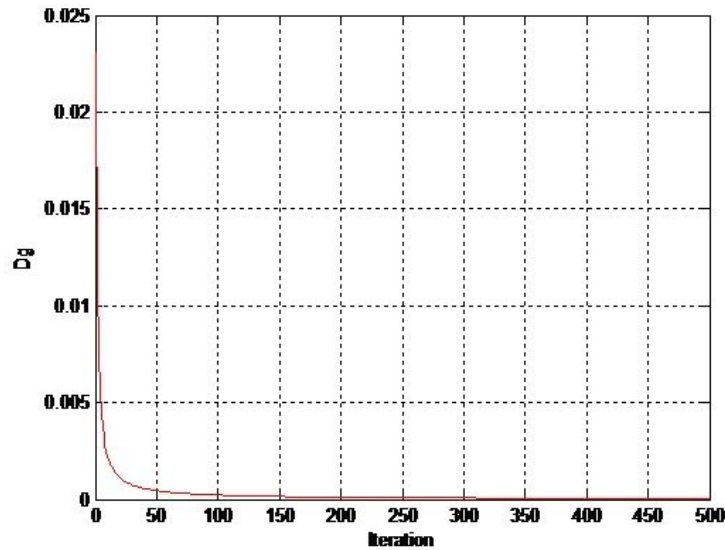


Figure 9. Convergence rate (vertical axis) of the test function ZDT3 for 500 iterations

Table 3. COMPARING THE AMOUNT OF PF ESTIMATION CONVERGENCE RATE OF THE PROPOSED ALGORITHM AND WEIGHTED SUM ALGORITHM

Test functions	Crowding distance (proposed algorithm)	weighted sum (Yangs algorithm)
ZDT1	2.63e-6	6.01e-5
ZDT2	4.89e-6	8.36e-5
ZDT3	7.34e-6	9.95e-5

3.3. A Comparative Study of Algorithm

Here, the performance of the proposed multi-objective firefly algorithm was compared to the answers obtained from Yang research, which used the weighted sum model instead of crowding distance. For fair comparison, both algorithms were used on the same population size and number of iteration respectively 500 and 500. The results obtained for both algorithms are presented in Table 3.

With regard to Table 3, it was observed that PF estimation error of the proposed algorithm for all test functions was less than the desired algorithm. Nevertheless, the implementation speed of the proposed multi-objective algorithm is lower than the weighted sum algorithm considering the calculation of the crowding distance in the new multi-objective firefly algorithm.

In Figure 10, the convergence rate for both weighted algorithm and crowding distance algorithm are displayed, related to the test function ZDT1. With regard to Figure 10, we observed that in the weighted sum method, the amount of the convergence distance D_g in the iteration of about 310 reached the lowest value. On the other hand, in the crowding distance method, the amount of the convergence distance D_g in the iteration of about 190, it reached the minimum value. This suggests that the error of the crowding distance method was less than the weighted sum method.

In order to compare the performance of the proposed multi-objective firefly algorithm, we selected several evolutionary algorithms performed in previous articles. Multi-objective optimization methods, selected to be compared with the proposed algorithm, include genetic algorithm (NSGAI), Bees algorithm, Differential Evolution algorithm (DE) and Strong Pareto Evolutionary Algorithm (SPEA). Comparison was made through measuring the convergence distance, results of which are presented in Table 4 .[12] It can be concluded from

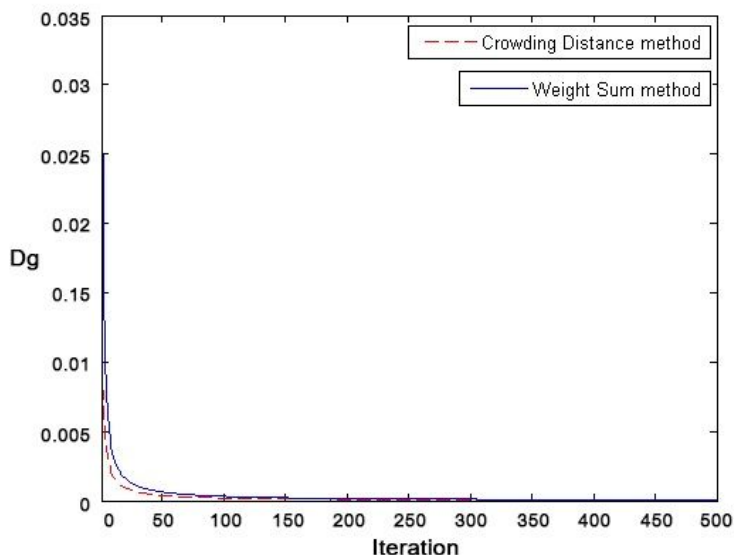


Figure 10. Comparison of the convergence rate for both algorithms

Table 4. COMPARING THE AMOUNT OF PF ESTIMATION CONVERGENCE RATE OF THE PROPOSED ALGORITHM AND WEIGHTED SUM ALGORITHM

Optimization method	ZDT1	ZDT2	ZDT3	Reference
NSGA-II	1.14e-1	7.24e-2	3.33e-2	[24]
BEES	1.91e-1	1.69e-2	2.40e-2	[25]
DEMO	1.18e-3	7.55e-4	1.08e-3	[26]
SPEA	4.75e-2	1.34e-3	1.78e-3	[27]
MOFA	1.97e-4	1.52e-4	1.90e-4	[12]
CDB-MOFA	2.91e-5	2.46e-5	2.11e-5	—

Table 4 that the results obtained from the proposed multi-objective algorithm for all the test functions was better, compared to the other evolutionary algorithms.

4. Conclusion

In this article, a new multi-objective firefly optimization method was provided. The efficiency of this algorithm was assessed by three famous test functions, and the proposed algorithm was compared to other multi-objective evolutionary algorithms. The results of comparison demonstrated that the proposed algorithm had higher accuracy and convergence speed, compared to other evolutionary algorithms. Therefore, the proposed algorithm is suitable for solving the optimization problems, in which accuracy is a priority.

REFERENCES

1. Coello C., Lamont B., and Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*, New York Springer, Vol. 5, 2009.
2. Abraham A., and Jain L. *Evolutionary multiobjective optimization. In Evolutionary Multiobjective Optimization*, Springer London, pp. 1-6, 2005.

3. Jun H. B., Cusin M., Kiritsis D., and Xirouchakis P, *A multi-objective evolutionary algorithm for EOL product recovery optimization: turbocharger case study*, International Journal of Production Research, 45(18-19), 4573-4594, 2007.
4. Yang X., *Nature-inspired optimization algorithms*, Elsevier, 2014.
5. Nakayama H, Yun Y.,and Yoon M, *Sequential approximate multiobjective optimization using computational intelligence.*, Springer Science and Business Media, 2009.
6. Konak A,Coit D. W., and Smith A. E, *Multi-objective optimization using genetic algorithms.*, Reliability Engineering and System Safety, 91(9), 992-1007, 2006.
7. Zhang Q., and Li H., *MOEA/D: A multiobjective evolutionary algorithm based on decomposition*, IEEE Transactions on evolutionary computation, 11(6), 712-731, 2007.
8. Venske, M. Gonalves, R. A., and Delgado M. R, *ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm*, Neurocomputing, 127, 65-77, 2014.
9. Sh Y., and Eberhart R, *A modified particle swarm optimizer*, In Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on pp. 69-73, 1998.
10. Zhang L. B., Zhou C. G., Liu X. H., Ma Z. Q., Ma M.,and Liang Y. C, *Solving multi objective optimization problems using particle swarm optimization*, The IEEE Congress Vol. 4, pp. 2400-2405,2003.
11. Yang X. S, *Firefly algorithm, stochastic test functions and design optimisation*, International Journal of Bio-Inspired Computation, 2(2), 78-84, 2010.
12. Yang X. S, *Multiobjective firefly algorithm for continuous optimization*, Engineering with Computers, 29(2), 175-184, 2013.
13. Fortin F. A., and Parizeau M, *Revisiting the nsga-ii crowding-distance computation*, In Proceedings of the 15th annual conference on Genetic and evolutionary computation pp. 623-630, 2013.
14. S.Z. Mirjalili,SH. Saremi, and H.Faris, *Grasshopper optimization algorithm for multi-objective optimization problems*, Springer Science+Business Media, DOI 10.1007/s10489-017-1019-8, LLC 2017.
15. S.Z. Mirjalili,Pr.Jangir, and SH. Saremi, *Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems*, Springer Science+Business Media, DOI 10.1007/s10489-016-0825-8, 2018.
16. Amani Saad, Salman A. Khan,and Amjad Mahmood, *A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design*, Elsevier,Swarm and Evolutionary Computation, 2018.
17. Yong Zhang , Dun-wei Gong , Jian-yong Sun ,and Bo-yang Qu, *A decomposition-based archiving approach for multi-objective evolutionary optimization*, Elsevier, Information Sciences , 2018.
18. Chunteng Bao, Lihong Xu, Erik D. Goodmanb, and Leilei Cao, *A Novel Non-Dominated Sorting Algorithm for Evolutionary Multi-objective Optimization*, Journal of Computational Science , 2017.
19. Apostolopoulos T., and Vlachos A., *Application of the firefly algorithm for solving the economic emissions load dispatch problem*, International Journal of Combinatorics., 2010.
20. Deb K., Thiele L., Laumanns M., and Zitzler, *Scalable test problems for evolutionary multiobjective optimization*, Springer London pp. 105-145, 2005.
21. Robic T., and Filipc B, *DEMO: Differential evolution for multiobjective optimization*, In International Conference on Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg pp. 520-533, 2005.
22. Shukla P. K., and Deb K, *On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods*, European Journal of Operational Research, 181(3), 1630-1652, 2007.
23. Zitzler E., Deb K.,and Thiele L, *Comparison of multiobjective evolutionary algorithms: Empirical results* Evolutionary computation, 8(2), 173-195, 2000.
24. Deb K., Pratap A., Agarwal S., and Meyarivan T, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, . IEEE transactions on evolutionary computation, 6(2), 182-197, 2002.
25. Pham D. T., and A Ghanbarzadeh , *Multi-objective optimisation using the bees algorithm*, Proceedings of IPROMS Conference, 2007.
26. Robic T., and Filipc B, *DEMO: Differential evolution for multiobjective optimization*, In International Conference on Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg pp. 520-533, 2005.
27. Madavan N. K, *Multiobjective optimization using a Pareto differential evolution approach*. In Evolutionary Computation, CEC02. Proceedings of the IEEE Congress on Vol. 2 pp. 1145-1150, 2002.