A Novel Group-Based Firefly Algorithm with Adaptive Intensity Behaviour

Adam Robson[1] [a], Kamlesh Mistry[2] [b] and Wai Lok Woo[2] [c]
[1]*School of Computer Science, Faculty of Technology, University of Sunderland, Sunderland, U.K.*
[2]*Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K.*

Keywords: Firefly Algorithm (FA), Intensity Behaviour, Attraction, Optimisation, Swarm Intelligence.

Abstract: This paper presents novel modifications to the Firefly Algorithm (FA) that manipulate the functionality of the intensity and attractiveness of fireflies through the incorporation of grouping behaviours into the movement of the fireflies. FA is one of the most well-known and actively researched swarm-based algorithms, gaining notoriety for the powerful search capability offered and overall computational simplicity. While the FA is an effective optimisation algorithm, it is unfortunately susceptible to the issue of premature convergence and oscillations within the swarm, which can lead to suboptimal performance. In the original FA formulation, at each iteration fireflies will instinctively move towards the most intensely bright firefly which is in closest proximity to them. The algorithm proposed in this paper manipulates the movement of the fireflies through modification of this intensity and attraction relationship, allowing the swarm to move in different ways, ultimately increasing the search diversity within the swarm. While group-based FAs have been proposed previously, the group-based FAs presented in this paper utilise a different approach to creating groups, implementing groupings based upon firefly performance at each iteration, resulting in continually varying groupings of fireflies, to further increase search diversity and maintain computational simplicity.

1 INTRODUCTION

Swarm intelligence has been an increasingly important and popular field throughout the last decade and is inspired by the collective behaviours of social swarms of animals and insects that occur within nature (Qi et al., 2017). These swarms are typically made up of a collection of unsophisticated agents that demonstrate a coordinated behaviour to achieve the desired goal of the swarm. Agents within the swarm interact with each other, creating a decentralised and self-organising swarm. Examples of notable and frequently used Swarm Intelligence optimisation methods are Ant Colony Optimisation (ACO), Artificial Bee Colony (ABC), Firefly Algorithm (FA) and Particle Swarm Optimisation (PSO) (Wang and Liu, 2019).

Swarm intelligence methods have been applied in a variety of optimisation problem areas, such as forecasting (Altherwi, 2020), scheduling (Bacanin et al., 2022), medical diagnosis (Nayak et al., 2020) and structural design (Chou and Ngo, 2017), demonstrating good performance and successful outcomes (Wang and Liu, 2019). FA has shown itself to be an effective and powerful optimisation technique in a variety of optimisation problems, but it is susceptible to issues such as oscillations in the swarm during the search process (Wang et al., 2017), and premature convergence (Qi et al., 2017). This paper proposes a novel Group-based Firefly Algorithm (GBFA) to overcome these issues. In the original FA, at each iteration fireflies will instinctively move toward the most intensely bright firefly which is in closest proximity to them. The FA variant proposed in this paper manipulates the behaviour of fireflies within the swarm through modification of the intensity and attraction relationship. This modification allows the swarm to move in different ways, ultimately increasing search diversity within the swarm, which prevents the

[a] https://orcid.org/0000-0003-2752-3381
[b] https://orcid.org/0000-0001-9371-7833
[c] https://orcid.org/0000-0002-8698-7605

Robson, A., Mistry, K. and Woo, W.
A Novel Group-Based Firefly Algorithm with Adaptive Intensity Behaviour.
DOI: 10.5220/0011672200003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 1*, pages 233-239
ISBN: 978-989-758-623-1; ISSN: 2184-433X
Copyright © 2023 by SCITEPRESS – Science and Technology Publications, Lda. Under CC license (CC BY-NC-ND 4.0)

occurrence of issues such as premature convergence or oscillations within the swarm. While group-based Firefly Algorithms have been proposed previously in work such as (Tong et al, 2017), (Suganya and Murugavalli, 2019) and (Cao et al, 2022), the group-based FA presented in this paper utilises a different approach to creating groups. Groupings are implemented at each iteration, based upon individual firefly performance, resulting in continually varying groupings of fireflies, allowing increased search diversity within the swarm.

This paper is organised as follows: Section 2 presents the standard FA, and a brief review of previous applications and research. Section 3 contains an overview of the GBFA proposed in this paper and briefly discusses related work. Section 4 shows the experiment design and describes the optimisation benchmark functions used to test the algorithm. Section 5 presents and discusses the results obtained from the proposed algorithm, and a comparison with the standard FA and recent research within the field. Section 6 provides a summary of the paper and the findings.

2 FIREFLY ALGORITHM

The Firefly Algorithm (FA) was originally developed in 2008 by Yang, with advances and applications noted in (Yang and He, 2013). It is a relatively new swarm intelligence algorithm and has been successfully applied to a variety of optimisation problems such as vehicle route planning, data fitting, scheduling, resource allocation (Ariyaratne et al., 2019). FA has also generated promising results for optimisation in engineering systems with applications in optimising power flow, micro-hydro applications, and the optimisation of electromagnetic devices (Parwanti et al., 2021).

FA is a population-based stochastic search algorithm, with similarities in functionality to Particle Swarm Optimisation (PSO). FA is a nature inspired algorithm, and behind the algorithmic design concepts of FA are the luminescence attribute, behaviour, and movement of tropical fireflies (Jain et al., 2021). Subsequently, FA has become a well-known optimisation technique for complex optimisation problems (Napalit and Ballera, 2021). Fireflies are individual agents within the swarm, and each uses its luminescence property to indicate their position within a search domain. Attractiveness of fireflies and movements within the swarm are based around the attractiveness of a firefly. FA is based on the following idealised rules (Yang and He, 2013):

(i) All fireflies within the swarm are unisex and therefore all fireflies will be attracted to one and other regardless of gender.
(ii) The attractiveness of a firefly is proportional to the brightness, with brightness decreasing as distance increases.
(iii) For any two flashing fireflies, the less bright of the two will move toward the brighter one.
(iv) If there is no brighter firefly, it will move randomly.
(v) The brightness of an individual firefly is determined by the objective function.

The attractiveness of a firefly is directly proportional to the intensity of brightness visible to adjacent fireflies, and attractiveness of a firefly is defined in (1),

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

with β as the attractiveness, r as the distance, and where β_0 is the attractiveness at $r = 0$. Firefly movement is based on the level of attraction to another firefly, as shown in (2),

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t \quad (2)$$

where the i^{th} firefly is attracted to move towards the j firefly if it has a higher intensity of brightness. With the first term representing the current location of firefly i and the second term representing movement from one position to another, due to attraction to firefly j. With the parameter r_{ij} being the Euclidian distance between the two fireflies. The light absorption coefficient is represented by γ, where $\gamma = 1$. β_0 is the original light attractiveness of each firefly at $r = 0$, and in the event that $\beta_0 = 0$, the firefly will take a simple random walk. The third term of (2) is a randomisation with α_t acting as the randomisation parameter, and ϵ_i^t is a vector containing random numbers drawn from a Gaussian distribution. The framework of FA can be broken down into the three stages. The first stage is initialisation. In this stage, the objective function, $f(x)$, is defined and a population of n fireflies are generated through the expression shown in (3), where i is the population ($i = 1,2, \dots N$), d is the dimension, with low and up representing the upper and lower bounds of the dimension.

$$x_{i,d} = low + rand(0,1)(up - low) \quad (3)$$

The second stage focuses on firefly movement based upon attraction. Each solution generated (x_i), is compared with all other solutions within the population of fireflies (x_j). Firefly x_i will change position based upon (2), if the objective function result of x_j is better than x_i. Each firefly is then evaluated based upon updated positions and sorted. At the third stage, the stopping criteria is checked, and the algorithm will end if the stopping criteria is satisfied. If the stopping criteria is not satisfied, the second stage will repeat. Pseudo code describing the functionality of the standard FA algorithm is as follows:

```
/* Define objective function
f(x_i), x_i = (x_{1,1}, ..., x_{i,d})
/* Initialise population of fireflies
x_i (i = 1,2, ..., n)
/* Begin
 while (t < MaxIteration)
   for i = 1 to n do
     for j = 1 to n do
      /* Calculate brightness
      b_i = f(x_i), b_j = f(x_j)
      /* Determine movement
       if (b_j > b_i)
         Move x_i towards x_j
       end if
     end for j
   end for i
Rank fireflies and set current best
end while
/* End
```

While the FA is a powerful optimisation algorithm, it is still susceptible to issues such as swarm oscillations and premature convergence, usually occurring because of fireflies moving towards non-optimal solutions within their local vicinity (Suganya and Murugavalli, 2019). The next section of this paper discusses the proposed algorithm to address these issues.

3 PROPOSED FIREFLY ALGORITHM

This paper proposes a novel Group-based Firefly Algorithm (GBFA) to alleviate the issues standard FA is susceptible to, such as oscillations within the swarm during the search process, resulting in decreased search diversity (Wang et al., 2017), and premature convergence (Qi et al., 2017). FA implementations which utilise grouping behaviours have been previously proposed in research such as (Tong et al, 2017) and (Cao et al, 2022) and have demonstrated positive results and successes in preventing premature convergence or swarm oscillations. Oscillations within the swarm is usually caused by too many or too few attractions within the swarm during the search process (Wang et al., 2017), and therefore manipulation of attraction of fireflies within the swarm is an important area for research.

In their work, (Tong et al, 2017) attempted to solve the problem of premature convergence by creating a modified evolutionary mechanism, with the swarm divided into fixed groups, each with different model parameters. While they achieved positive results, showing a good balance between exploration and exploitation, the implementation was not without flaws, the main being that the overall computational simplicity of the FA was reduced through these augmentations. In other work, (Cao et al, 2022) proposed groupings based upon visual fields and an observer strategy and encouraged collaboration between groups by having individual fireflies existing in multiple groups. Again, while the work of Cao et al. shows promising results, the implied overhead of the additional behaviours added to the algorithmic design of FA sacrifices computational simplicity for more powerful search results.

While it is important to increase search diversity within the swarm, reductions in computational simplicity can have negative impacts on real-time systems implementing swarm intelligence algorithms that require updating to changes in the problem domain, such as vehicle route planning (Chandrawati and Sari, 2018), or controlling drone swarms (Siemiatkowska and Stecz, 2021). It is therefore important to try and obtain a balance between ensuring computational simplicity and algorithmic performance. The GBFA proposed in this paper seeks to alleviate issues with the standard FA, by increasing search diversity with the addition of dynamic groups, whilst maintaining computational simplicity.

3.1 Group-Based Firefly Algorithm Functionality

In the standard FA, fireflies will instinctively move toward the most intensely bright firefly that is in closest proximity to them. In the GBFA algorithm, the intensity and movement relationship is manipulated through the addition of dynamic groupings to each iteration. At each iteration, fireflies are ranked based on their brightness. Groupings are then dynamically allocated based upon the position of fireflies within this ranking and will fluctuate at each iteration based upon the new rankings. For example, if the group size is set to five, the highest ranked firefly (the current best) will be the leader of the first group, which contains the fireflies ranked second to fifth. The second group leader will be the firefly ranked at the sixth position, with the fireflies ranked seventh to tenth forming the rest of that group. Example groupings are visualised in Figures 1 and 2. Each of the groupings is assigned a group leader, which all other fireflies within the group will move toward. Movement is handled in the same way as the standard FA, as shown in equation (2), except with the caveat that group members move only toward the leader of their group. If a group member has the same value returned from the objective as the group leader, the group member will complete the same random walk as in the standard FA. This modification increases search diversity and allows the swarm to move in different ways, overcoming issues such as premature convergence and oscillations. Groups are dynamically allocated at each iteration, based on sizing parameter g, which must be defined before the algorithm begins. The pseudo code for the GBFA can be seen below, where the group size is set to 5 ($g = 5$).

```
/* Define objective function
f(x_i), x_i = (x_{1,1}, ..., x_{i,d})
/* Initialise population of fireflies
x_i (i = 1,2, ..., n)
/* Define group size
g = 5
/* Begin
 while (t < MaxIteration)
   for i = 1 to n do
     Create groupings based on size g
     Assign group lead as g_1
     /* Calculate brightness of g_1
     b_{g_1} = f(g_1)
     for j = 1 to g do
       /* Calculate brightness
       b_j = f(g_j)
       /* Determine movement
       if (b_{g_1} > b_j)
         Move g_j towards b_{g_1}
       end if
     end for j
   end for i
Rank fireflies and set current best
end while
/* End
```

Group sizes can be set to allow groups to operate independently of each other, Figures 1 and 2 show a visualisation of how groups are organised. With Figure 1 showing a group size (g) set to five ($g = 5$), with a population (n) of 50 ($n = 50$), where each colour represents a different group of fireflies within the swarm, and the visible group leaders are x_1, x_6, x_{41} and x_{46}. Figure 2 shows a group size of ten ($g = 10$), with a population of 80 ($n = 80$), again where each group within the swarm is represented by a different colour and the visible group leaders are x_1, x_{11}, x_{61} and x_{71}.

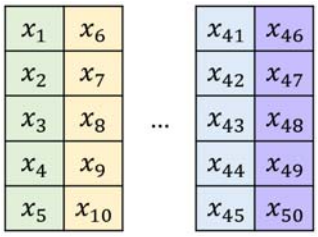

Figure 1: GBFA group example where $g = 5$ and $n = 50$.

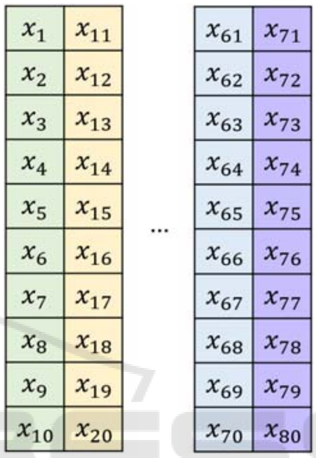

Figure 2: GBFA group example where $g = 10$ and $n = 80$.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

4 EXPERIMENT DESIGN

To review the performance of the proposed GBFA, a standard FA and the GBFA were implemented using Python 3.9, along with eight bound constrained global optimisation problems that are commonly used to test the performance of optimisation algorithms (Ackley, Easom, Griewank, Michalewicz, Rastrigin, Rosenbrock, Schwefel and Sphere). These were chosen based on the optimisation problems noted by (Fister et al., 2013), which have seen continued usage in modern optimisation algorithm testing in work such as (Wang et al., 2017) and (Zivkovic et al., 2022). Functions were implemented using the NumPy Python library and each of the benchmark functions are outlined within this section.

All experiments were conducted using 30 runs, each with 200 iterations and a population of $n = 40$ fireflies. The group size for the GBFA is $g = 5$ and the number of dimensions is $d = 2$.

4.1 Ackley

The Ackley function is shown in (4), it is highly multimodal and has a global minimum of $f = 0$ at $s = (0,0,\ldots,0)$, where $s_i \in [-32.768,32.768]$, $i = 1,2,\ldots,D$.

$$f_1(s)\sum_{i=1}^{D-1}(20 + e - 20e^{-0.2\sqrt{(0.5(s_{i+1}^2+s_i^2)}} - e^{0.5(\cos(2\pi s_{i+1})+\cos(2\pi s_i))} \quad (4)$$

4.2 Easom

The Easom function is shown in (5) has several local minimum and global minimum, $f = -1$ at $s = (\pi,\pi,\ldots,\pi)$, where $s_i = [-2\pi, 2\pi]$.

$$f(s) = (-1)^D\left(\prod_{i=1}^{D}\cos^2(s_i)\right) \exp\left[-\sum_{i=1}^{D}(s_i-\pi)^2\right] \quad (5)$$

4.3 Griewank

The Griewank function, shown in (6), has a global minimum of $f = 0$ at $x = (0,0,\ldots,0)$, where $s_i \in [-600,600], i = 1,2,\ldots,D$. It is also important to note that when the number of variables is higher than 30, this function is highly multimodal.

$$f(s) = -\prod_{i=1}^{D}\cos\left(\frac{s_i}{\sqrt{i}}\right) + \sum_{i=1}^{D}\frac{s_i^2}{4000} + 1 \quad (6)$$

4.4 Michalewicz

The Michalewicz function, in two-dimensional parameter space, has the global minimum of $f = -1.8013$ at $s = (2.20319,1.57049)$ and is shown in (7).

$$f(s) = -\sum_{i=1}^{D}\sin(s_i)\left[\sin\left(\frac{is_i^2}{\pi}\right)\right]^{2\cdot 10} \quad (7)$$

4.5 Rastrigin

The Rastrigin function is shown in (8), where $s_i \in [-15,15]$, $i = 1,2,\ldots,D$. It has a global minimum of $f = 0$ at $x = (0,0,\ldots,0)$ and is highly multimodal.

$$f(s) = D * 10 + \sum_{i=1}^{D}(s_i^2 - 10\cos(2\pi s_i)) \quad (8)$$

4.6 Rosenbrock

The Rosenbrock function is also commonly known as the ‘banana function’. It has several local optima and is shown in (9), where $s_i \in [-15,15]$, $i = 1,2,\ldots,D$. The function has a global minimum of $f = 0$ at $s = (1,1,\ldots,1)$.

$$f(s) = \sum_{i=1}^{D}100(s_{i+1} - s_i^2)^2 + (s_i - 1)^2 \quad (9)$$

4.7 Schwefel

The Schwefel function is shown in (10), where $s_i \in [-500,500]$, $i = 1,2,\ldots,D$. This is a highly multimodal function and has a global minimum of $f = 0$ at $s = (1,1,\ldots,1)$.

$$f(s) = 418.9829 * D - \sum_{i=1}^{D}s_i\sin\sqrt{|s_i|} \quad (10)$$

4.8 Sphere

De Jong’s Sphere function is shown in (11), where $s_i \in [-600,600], i = 1,2,\ldots,D$. This is a unimodal and convex function and has a global minimum of $f = 0$ at $s = (0,0,\ldots,0)$.

$$f(s) = \sum_{i=1}^{D}s_i^2 \quad (11)$$

5 RESULTS

The primary goal of the experiments conducted was to show that the GBFA could outperform a standard FA implementation through the incorporation of grouping behaviours into the movement of the fireflies. The data shown in Table 1 is the average best results for the FA and GBFA when benchmarked using each of the eight optimisation functions described in Section 4. The experiments were conducted over 30 runs at 200 iterations each, in 2 dimensions, with a population of 40. Based upon the population size used, GBFA was configured to run with a group size defined as 5, meaning there would be a total of eight groups dynamically allocated at each iteration of execution.

Table 1: Comparison of the average best results of FA and GBFA, with the best result in **bold**.

Function	FA	GBFA
Ackley	1.59E-01	**1.32E-01**
Easom	**-1.00E+00**	**-1.00E+00**
Griewank	2.31E-02	**2.12E-02**
Michalewicz	**-1.80E+00**	**-1.80E+00**
Rastrigin	5.78E-02	**4.86E-02**
Rosenbrock	2.93E-03	**1.36E-03**
Schwefel	8.60E-01	**6.05E-01**
Sphere	5.83E-01	**4.36E-01**

The results shown in Table 1 are promising, with the GBFA outperforming the standard FA in six of the eight benchmark functions: Ackley, Griewank, Rastrigin, Rosenbrock, Schewefel and Sphere, and performing equally as well as the standard FA in the remaining benchmark functions: Easom and Michalewicz. This shows us that the increased search diversity offered by the GBFA is capable of addressing issues with FA that lead to suboptimal performance, such as premature convergence or oscillations within the swarm. Table 2 shows a comparison of results with a recent study (Wahid et al. 2018), which presents a hybrid FA. The standard FA is combined with a Genetic Algorithm (GA) and an embedded search pattern and is referred to as GA-FA-PS. The GA is used to modify the positions of the fireflies within the swarm after the fireflies have been randomly placed within the search domain, before the first execution of the FA. At the end of each iteration, the embedded search pattern is used to increase search diversity, through the introduction of further exploitation and exploration.

Table 2: Comparison of the average best of the GBFA and a modified FA presented by (Wahid et al. 2018), with the best results in **bold**.

Function	GBFA	GA-FA-PS
Ackley	**1.32E-01**	2.52E-01
Rosenbrock	**1.36E-03**	-1.92E+00
Sphere	**4.36E-01**	-3.01E+00

The work presented by Wahid et al. attempts to address the premature convergence issues within the swarm by modifying the functionality of the FA, much like the algorithm presented in this paper. Wahid et al. conducted their research study using only three optimisation benchmark functions: Ackley, Rosenbrock and Sphere. The data is again quite positive, as while the hybrid FA proposed by Wahid et al. showed the capability to outperform the standard FA, the GBFA presented in this paper was able to significantly outperform the hybrid FA, particularly in the Rosenbrock and Sphere optimisation benchmark functions.

Table 3 shows a comparison of results with another recent study (Gamao et al., 2019), which presents a modified mutated FA, to attempt to address the premature convergence issue of the standard FA and improve results. In this study three FA variants have been proposed: Mutated Firefly Algorithm (MFA), Modified Mutated Firefly Algorithm-Las Vegas (MMFA-LV) and Modified Mutated Firefly Algorithm-Monte Carlo (MMFA-MC). These algorithms have been implemented in an attempt to increase search diversity of the swarm through mutation of the lower performing fireflies and higher performing fireflies.

Table 3: Comparison of the average best results of GBFA and (Gamao et al., 2019), with the best result in **bold**.

Function	GBFA	MFA	MMFA-LV	MMFA-MC
Ackley	**1.32E-01**	5.97E+00	5.59E+00	3.87E+00
Rosenbrock	**1.36E-03**	7.54E+01	7.21E+01	4.69E+01
Sphere	**4.36E-01**	3.98E+00	1.80E+00	1.77E+00

The proposed mutation process enhances features and attractiveness of the bottom forty percent of the swarm, by mutating them with the top forty percent of the swarm. The algorithms presented by Gamao et al. also attempt to improve the search capabilities of FA by combining the mutation principles with a Las Vegas (LV) search algorithm and a Monte Carlo (MC) search algorithm. Gamao et al. used only three optimisation benchmark functions: Ackley, Rosenbrock and Sphere to test their algorithms. Again, while the algorithm variants proposed by Gamao et al. were able to show improvements on the standard FA, the GBFA presented in this paper shows significantly improved results.

6 CONCLUSIONS

The attraction behaviour of FA has an extremely important role within the search process of the FA and controls how the swarm moves and finds candidate solutions. As previously noted, modification of the attraction behaviour within the FA is an important area to research when trying to alleviate issues such as premature convergence or oscillations within the swarm, as it can result in having the firefly and swarm move in different ways to the standard FA, ultimately increasing search diversity also. The GBFA presented in this study has shown that positive results can be achieved through modification of this attractiveness relationship and can allow for fireflies to move toward positions that they would normally not, allowing for greater search capabilities and enhanced performance.

While the results observed in this study are extremely positive, further experimentation with the GBFA is required. The next stage of research for this algorithm is to tune the group size parameter and the number of groups within a swarm, to evaluate the performance when using larger or smaller group sizes. Additionally, concepts such as the cross-group communication behaviour seen in other research within the area, such as (Cao et al., 2022), that allows individual fireflies to exist across multiple groups can be incorporated into the GBFA to investigate the impact that this has on the performance of the algorithm.

REFERENCES

Altherwi, A. (2020). Application of the Firefly algorithm for optimal production and demand forecasting at selected Industrial Plant. *Open Journal of Business and Management, 08*(06), 2451-2459. doi:10.4236/ojbm.2020.86151

Ariyaratne, M., Fernando, T., & Weerakoon, S. (2019). Solving systems of nonlinear equations using a modified Firefly Algorithm (MODFA). Swarm and Evolutionary Computation, 48, 72-92. doi:10.1016/j.swevo.2019.03.010

Bacanin, N., Zivkovic, M., Bezdan, T., Venkatachalam, K., and Abouhawwash, M. (2022). Modified firefly algorithm for workflow scheduling in cloud-edge environment. *Neural Computing and Applications, 34*(11), 9043-9068. doi:10.1007/s00521-022-06925-y

Cao, L., Ben, K., Peng, H., and Zhang, X. (2022). Enhancing Firefly algorithm with adaptive multi-group mechanism. *Applied Intelligence, 52*(9), 9795-9815. doi:10.1007/s10489-021-02766-9

Chandrawati, T. B., and Sari, R. F. (2018). A review of Firefly algorithms for path planning, vehicle routing and traveling salesman problems. *2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI)*. doi:10.1109/icon-eei.2018.8784312

Chou, J., and Ngo, N. (2017). Modified Firefly algorithm for multidimensional optimization in Structural Design Problems. *Structural and Multidisciplinary Optimization, 55*(6), 2013-2028. doi:10.1007/s00158-016-1624-x

Fister, I., Yang, X., Brest, J., and Fister, I. (2013). Modified Firefly algorithm using quaternion representation. *Expert Systems with Applications, 40*(18), 7220-7230. doi:10.1016/j.eswa.2013.06.070

Gamao, A. O., Gerardo, B. D., and Medina, R. P. (2019). Modified mutated Firefly algorithm. *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. doi:10.1109/icetas48360.2019.9117417

Jain, A., Sharma, S., and Sharma, S. (2021). Firefly algorithm. *Nature-Inspired Algorithms Applications,* 157-180. doi:10.1002/9781119681984.ch6

Napalit, A. P., and Ballera, M. A. (2021). Application of firefly algorithm in scheduling. *2021 IEEE International Conference on Computing (ICOCO)*. doi:10.1109/icoco53166.2021.9673581

Nayak, J., Naik, B., Dinesh, P., Vakula, K., and Dash, P. B. (2020). Firefly algorithm in biomedical and health care: Advances, issues and challenges. *SN Computer Science, 1*(6). doi:10.1007/s42979-020-00320-x

Parwanti, A., Wahyudi, S. I., Ni'Am, M. F., Ali, M., Iswinarti, and Haikal, M. A. (2021). Modified Firefly algorithm for optimization of the water level in the tank. *2021 3rd International Conference on Research and Academic Community Services (ICRACOS)*. doi:10.1109/icracos53680.2021.9701981

Qi, X., Zhu, S., and Zhang, H. (2017). A hybrid Firefly algorithm. *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. doi:10.1109/iaeac.2017.8054023

Siemiatkowska, B., and Stecz, W. (2021). A framework for planning and execution of drone swarm missions in a hostile environment. *Sensors, 21*(12), 4150. doi:10.3390/s21124150

Suganya, T. S., and Murugavalli, S. (2019). A hybrid group search optimization: Firefly Algorithm-based Big Data Framework for ancient script recognition. *Soft Computing, 24*(14), 10933-10941. doi:10.1007/s00500-019-04596-x

Tong, N., Fu, Q., Zhong, C., and Wang, P. (2017). A multi-group Firefly algorithm for numerical optimization. *Journal of Physics: Conference Series, 887*(1), 012060. doi:10.1088/1742-6596/887/1/012060

Wahid, F., Ghazali, R., and Shah, H. (2018). An improved hybrid firefly algorithm for solving optimization problems. *Advances in Intelligent Systems and Computing,* 14-23. doi:10.1007/978-3-319-72550-5_2

Wang, C., and Liu, K. (2019). A randomly guided Firefly algorithm based on elitist strategy and its applications. *IEEE Access, 7*, 130373-130387. doi:10.1109/access.2019.2940582

Wang, H., Wang, W., Zhou, X., Sun, H., Zhao, J., Yu, X., and Cui, Z. (2017). Firefly algorithm with neighborhood attraction. *Information Sciences, 382-383*, 374-387. doi:10.1016/j.ins.2016.12.024

Wang, W., Xu, L., Chau, K., and Xu, D. (2020). Yin-Yang Firefly algorithm based on dimensionally cauchy mutation. *Expert Systems with Applications, 150*, 113216. doi:10.1016/j.eswa.2020.113216

Yang, X. S., and He, X. (2013). Firefly Algorithm: Recent advances and applications. *International Journal of Swarm Intelligence, 1*(1), 36. doi:10.1504/ijsi.2013.055801

Zivkovic, M., Tair, M., K, V., Bacanin, N., Hubálovský, Š, and Trojovský, P. (2022). Novel hybrid firefly algorithm: An application to enhance XGBoost tuning for intrusion detection classification. *PeerJ Computer Science, 8*. doi:10.7717/peerj-cs.956.