Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

Enhancing Optimization Efficiency: The Modified Flower Pollination Algorithm with Spiral Renewal Mechanism

Shifaa N. Sahi[1], and Nazar K. Hussein[2]

[1,2] Dept. of Mathematics, College of Computer Science and Mathematics, Tikrit University, Iraq
Email: shafaanamh@gmail.com and *nazar.dikhil@tu.edu.iq*

Article information

Article history:
Received :26/7/2023
Accepted :22/9/2023
Available online :

Abstract

The Flower Pollination Algorithm (FPA) is a well-known method for solving optimization problems using swarm intelligence. But, like other swarm-based algorithms, FPA has some problems, such as being prone to local optima and being slow to converge in high-dimensional global optimization problems. The Modified Flower Pollination Algorithm (MFPA) was made to deal with these problems. MFPA tries to improve exploitation capabilities and accuracy of solutions by using a balanced approach to exploitation and exploration. It uses the spiral renewal mechanism from the Moth Flame Optimization (MFO) algorithm to avoid local optimums. These strategies help FPA work better when they are used together. Benchmark test functions, including those from the CEC 2017 test suite, were tested experimentally. When MFPA's performance was compared to that of other swarm intelligence algorithms and more complex methods, the results were very positive and encouraging.

Keywords:
Modified Flower Pollination Algorithm (MFPA), Optimization problems, Swarm intelligence, Spiral renewal mechanism and Benchmark test functions.

Correspondence:
Author : Nazar K. Hussei
Email: *nazar.dikhil@tu.edu.iq*

I. INTRODUCTION

An "optimization problem" is a type of math problem that involves changing a set of variables to decrease or increase an objective function while meeting a set of constraints. In other words, the challenge is to figure out which of the options that meet the constraints is the best. There are two main types of optimization problems: those that are linear and those that are not. In linear optimization problems, both the goal function and the rules are linear functions. In nonlinear optimization problems, the goal function and/or the rules are nonlinear functions. There are problems that need to be fixed in many fields, such as physics, computer science, economics, and engineering. [1]. Some examples of optimization problems are:
Find the lowest cost to produce a specific quantity of products. maximize firm profits by observing production limitations. Locate the graph's shortest path between two spots. Reduce the regression model's error to the classification model's highest level of accuracy. The Newton's method, the genetic algorithm, the gradient ratio, and many other optimization algorithms are available in Collision and can be utilized to solve optimization problems. The particular challenge, as well as the traits and constraints of the target function, influence the algorithm choice. There are numerous different kinds of optimization algorithms, and each has unique procedures and strategies. However, the following are some broad groups of optimization algorithms: [2] Genetic algorithms simulate natural evolution to produce new solutions through crossover and mutation.

The best solutions are chosen for the following generation after being reviewed for applicability [3]. These techniques, known as Particle Packing Optimization (PSO), are based on how particle swarms behave. Each particle moves across the search region as a potential solution and modifies its position based on its ideal location and swarm location. [4] These techniques, called ant colony optimization (ACO), use ant behavior to determine the shortest path between an ant colony and a food supply. These algorithms are based on a physical cooling process, in which the material is gradually heated and chilled to obtain the lowest energy state. As the ants walk across the search area, each one represents a potential solution and emits pheromones that entice other ants to choose the same path. To traverse the search space and avoid the local optima, the algorithm begins at a high temperature and gradually reduces it. [6], Tapu Search (TS) These algorithms are predicated on the notion of keeping lists of blocked solutions or recently visited blocks. While moving through the search space, the algorithm avoids blocking solutions and discovers new regions.

Yang Shishi, the researcher who also came up with the Firefly algorithm, first presented the Flower Pollination Algorithm (FPA), one of the metaphysical optimization techniques. The flower pollination algorithm, or FPA, was introduced by the FPA in their work "Floral Pollination Algorithm for Global Improvement," which was published in the journal "Unconventional Arithmetic and Natural Arithmetic" in 2012 [7]. It is a method of pollinating plants. The method applies a cutting-edge model to the answer put forth in each cycle of optimization [8]. Since pollination of plants occurs at random, the algorithm makes use of this concept to generate a random pollinator, aiding in the creation of diversity and encouraging the investigation of prospective areas for finding solutions. Numerous issues, including those involving numerical approaches, performance optimization, data categorization, image analysis, artificial intelligence, and deep learning, can be solved using the flower pollination algorithm. The algorithm for pollinating flowers (FPA) is a recently developed optimization technique that significantly improves upon Yang's initial algorithm from 2012. The following is a list of recent FPA publications and references:

The Flower Binary Pollination (BFPA) algorithm was jointly developed and is designed to solve binary optimization problems, particularly for feature selection tasks [9]. By incorporating economic and environmental goals into the optimization process, the FPA algorithm provides a new approach to solving the CEED problem. use of an algorithm for flower pollination in order to address the issues of economical load transfer (ELD) and universal economic discharge (CEED) in power systems, a new optimization approach based on the flower pollination algorithm (FPA) was proposed in this study. The suggested FPA is a meta-thinking algorithm that draws inspiration from how flowers pollinate themselves in nature [11]. The floral pollination algorithm was modified, and a novel research approach was used to enhance the FPA algorithm's effectiveness in resolving design issues in mechanical engineering. Using the DECD-FPA algorithm, a potential strategy to solve engineering design issues with numerous goals and restrictions was proposed [12]. Algorithm for adaptive flower pollination in order to improve the FPA method's performance in addressing the issue of decreasing the software test suite, this paper's contribution is to create an adaptive version of the algorithm. A viable method for enhancing software testing issues for intricate and sizable software systems is the AFPA algorithm [13]. Synthesis of a linear antenna matrix The AFPA algorithm offers the suggested method has promise for creating antenna arrays with better performance and less complexity and expense. This strategy could be applied to many kinds of antenna arrays and additional antenna design and optimization problems [14]. algorithm for pollinating flowers, a thorough analysis of the FPA algorithm, its modifications, and its uses in many disciplines in order to help researchers and practitioners select the best optimization algorithm for their particular situation, this study offers insight into the benefits and drawbacks of the FPA algorithm. The evaluation also identifies areas that require more study to enhance the effectiveness and performance of the FPA algorithm [15]. In order to increase the performance of an innovative multiple FPA method for the problem of continuous improvement, provide an innovative floral pollination algorithm. To get the greatest optimization outcomes, the suggested IFPA algorithm makes better use of the search space while maintaining population variety. Its performance is thoroughly examined. Second, it suggests a brand-new technique for parameter adjustment that could result in higher optimization outcomes [17], and the upgraded flower pollination algorithm boosts its efficiency in resolving complex optimization issues. The suggested algorithm has a new mutation factor, an updated flower set updating mechanism, and a flower pollination algorithm with appealing pollination. The flower pollination algorithm has been strengthened by a thorough investigation of the suggested method, which included comparison with other contemporary metaphysical algorithms in many comparisons on a standard function [19]. Adaptive learning and new pollination mechanisms by integrating vaccinations from random flowers and nearby flowers, the improved pollination process enhances the ability to seek, and the adaptive learning mechanism updates the step size and aids in balancing exploration and exploitation. An enhanced flower pollination method was tested on several datasets [20], and the findings reveal that it performs better in terms of solution quality and convergence speed than previous algorithms [21]. The algorithm is straightforward and simple to use, and in some situations, it performs better than other optimization algorithms. However, it may not always be the best; hence, it is advised to test out a few different optimization methods to acquire the best outcomes for a specific optimization problem.

Although the FPA algorithm offers numerous benefits, it also has several drawbacks, such as: can take a while: This method could take a while longer than others to get the best solution. The size of the issue can have a big impact; algorithms may struggle to handle large-scale or high-dimensional issues, and they may require modifications to perform better in these situations. Sometimes the best answer is not found. In certain circumstances, the algorithm may have trouble locating the best solution; other times, it may get close but fall short. It can be necessary to make modifications to boost their performance. By making some changes to the algorithm, such as altering the parameter values or how the solution is updated, the algorithm's performance can be enhanced. You might experience issues with local solutions that get you stuck: Particularly if the start is bad, algorithms may have trouble stumbling over local solutions and failing to reach the ideal answer.

2. Pollination algorithm for original flowers

Abiotic and biotic pollination are the two basic forms. About 90% of blooming plants are biophilapes, meaning that pollen is dispersed by pollinators like insects and animals. The remaining 10% of pollination occurs in an abiotic form, which implies that it is not required. Any pollinators are welcome, as wind and water dispersal aid in pollinating these flowering plants. There are at least 200,000 distinct pollinator species, including insects, bats, and birds. Pollinators, or pollinators, as they are frequently known, can be highly diverse. Pollinators like bees are a fantastic example because they support floral stability. This indicates that these pollinators favor visiting some kinds of blooms while avoiding others. Due to the increased pollen transmission to the same plant or plants and increased flower reproduction, this floral stability offers an evolutionary benefit. Since pollinators can ensure a useful supply of nectar with little effort and low learning or exploration costs, rather than concentrating on something novel, unpredictable, and perhaps more advantageous than flower species, pollinators may also benefit from flowers' stability. Pollination can occur from pollen from a different plant, whereas self-pollination is the fertilization of the flowers of one plant, such as a peach blossom, from pollen from the same flower or different flowers from the same plant, usually in the absence of nectar. Flowering stability will require minimal investment costs and will likely need a guaranteed amount of nectar to achieve pollination. Additionally, floral stability can be used as a stepping stone upward using similarities or differences between two blooms, and bees and birds may demonstrate fiber-like flight behavior in which jumping steps or flight distances are controlled by the distribution of fibers.

The following four rules can be used to explain how flowers remain stable and how pollinators behave when pollinating: Abiotic self-pollination is local pollination, whereas biopollination is universal pollination in which pollinators travel along the fiber. As an alternative, flower wandering can be viewed as a reproductive likelihood that increases in direct proportion to how similar two blooms are to one another. Additionally, it should be highlighted that a potential key to controlling the switching or interaction between local and universal pollination the following four rules can be used to explain how flowers remain stable and how pollinators behave when pollinating: Abiotic self-pollination is local pollination, whereas biopollination is universal pollination in which pollinators travel along the fiber. As an alternative, flower wandering can be viewed as a reproductive likelihood that increases in direct proportion to how similar two blooms are to one another. Additionally, it should be highlighted that a potential key to controlling the switching or interaction between local and universal pollination $p \in [0,1]$.

2.1 Mathematical representation of mass vaccination and local vaccination

Global and local vaccination models in mathematics The two primary phases of the flower pollination algorithm are mass pollination and local pollination. Pollinators like insects spread pollen during mass pollination. Pollen can travel long distances because insects can fly and move quickly, which can be stated mathematically as follows:

$$y_i^{t+1} = y_i^t + l(\beta) * (y_i^t - y_{best}). \qquad \dots(1)$$

While the solution in the iteration t is y_i^t and y_best the best solution obtained in the iteration is the scaling agent to control the step size follow Levy distribution,
For local pollination, it can be self-pollinating, that is, between flowers in the same plant
Its mathematical expression is as follows:

$$y_i^{t+1} = y_i^t + \epsilon * \left(y_j^t - y_k^t\right). \qquad \dots(2)$$

While the current solution at y_i^t iteration t and y_j^t, y_k^t are two different solutions randomly selected, ϵ is a parameter follow the uniform distribution

Pseudo-code of Flower pollination algorithm

```
Objective min of, $f(X),\ X = (X_1, X_2, \dots, X_N)$
Initialize a population of flowers /pollen gametes with random solutionsN
Find the best solution in the initial population $g^*$
Define a switch probability $p \in [0,1]$
While ()$t < Maxiter$
    For (all flowers in the population)N
      If rand<p
         Calculate L which obeys a Levy distribution
          Global pollination  $y_i^{t+1} = y_i^t + l(\beta) * (y_i^t - y_{best})$
       Else
          Generate from a uniform distribution in $\epsilon$ [0,1]
          Randomly choose and from j and k form N
          Local pollination  $y_i^{t+1} = y_i^t + \epsilon * (y_j^t - y_k^t$
      End if
         Evaluate new solutions
         Update the best solution
    End for
Find the current best solution $g_*$
End while
```

3. Proposed method

In the original blossoming pollination method, the region to be searched is determined by the fiber distribution ratio, and the optimal solution is found by adding a percentage of the difference between the current and optimal solutions. The upgrade procedure might include superior but less varied options that are far from the current ones. In order to enhance the search, step size was included in the original equation as a third factor. The using the factor v which enables it to produce new solutions in regions that are very far from the current and best solution. This increases the likelihood that the algorithm will approach the global solution and enhances the search for the best, as shown by the following equation:

$X_{New} = x^t_{r_1} + v * (g^* - x^t_i).$ …(3)

where v is a random value in the interval $(-c, c)$ where p calculate as follow:

$c = 1 - \left(\frac{t}{maxiter}\right).$ …(4)

Despite having parallels to WOA and MFO was only recently proposed [27] [26]. An algorithm is used to imitate moths' nocturnal flight patterns. During the night, the moth will switch to a straight-line positioning flight mode in order to maintain a straight line of flight. The moth is able to fly far at night by maintaining a specific angle with the moon. When the moth and the simulated moon are too close together, the moth will spiral around the simulated moon at a precise angle, and the simulated moon will then become what is known as a flame in the algorithm. Due to its superior precision and speed of convergence, the MFO method has been used in numerous disciplines. The spiral updating mechanism, which is the main component of MFO, makes sure the moth may locate the ideal solution without encountering the local optimum. The spiral renewal mechanism is described by the following equation:

$M(P_i, S_j) = |S_j - P_i|. e^{bk}. \cos(2\pi K) + S_j,$ …(5)

where P_i is the ith moth, $|S_j - P|$ is the distance between moth and flame, S_j is the jth flame, b represents the logarithmic helix constant, $k \in [-1,1]$ is the distance parameter. The MFO algorithm inspires the MS mechanism and its core update formula is as follows:

$X_{New} = |\, g^* - X_{r1}|. e^{k}. \cos(2\pi k) + X_{r2},$ …(6)

where X_{ms} is a new species group updated by MS mechanism X_{new} is the population of the original algorithm $|\, g^* - X_{r1}|$ is the distance between g^* and X_{r1}. $k \in [-1,1]$ is the distance parameter.

Pseudo-code of modified Flower pollination algorithm

```
Objective min or max , $f(x) X = (X_1, X_2, \dots, X_N)$
Initialize a population of flowers /pollen gametes with random solutionsN
Find the best solution in the initial population $g^*$
Define a switch probability $p \in [0,1]$
While ()$t < Maxiter$
        Compute the value of $c$ by eq (4)
        Evaluate the objective function of each solution $x^t_i$ as $Fitness(i)$
        For (all flowers in the population)N
        $Xnew = zeros(dim\quad)$
        If rand<p
             Generate the random value $v$ in the interval $(-c, c)$
            Global pollination via eq(3)
        Else
             Generate from a uniform distribution in $\epsilon$ [0,1]
             Randomly choose $r_1, r_2$ from N
            local pollination via eq(6)
        End if
  Evaluate the objective function at the solution $Xnew$ as $F_{new}$
           If $F_{new} < Fitness(i)$
              $x^{t+1}_i = Xnew$
           End for
           Update the best solution
        End for
Find the current best solution $g_*$
End while.
```

4. Experimental results and analysis

To replicate actual difficulties in the real world, it is highly beneficial to use a range of test functions. Researchers continuously evaluate the performance of stochastic algorithms using these functions, contrasting MFPA's results with those of the CEC2017 test functions [22]. These groups were employed in a number of measuring functions in the 1950s to assess how well the suggested algorithm performed in comparison to a number of sophisticated metaphysical algorithms. BAT [23], PSO [4], CSO [24], SCA [25], WOA [26], MFO [27], HHO [28], and SSA [29] are a few examples.

An even better algorithm is MFPA. The study's next paragraph will go into further information about the test outcomes. For practical sectors like electrical engineering, water analysis, and health, where proficiency tests are one of the most crucial instruments for developing systems and improving algorithms, the existence of this improved method is crucial. The CEC2017 test includes a number of unique functions that are used to evaluate the effectiveness of algorithms.
Since metaphysical algorithms are unpredictable, it is best to assess and contrast them objectively to provide fair test circumstances. These algorithms were all created in Python 3 and tested on an Intel i5-7300U core CPU running at 2.50 GHz with 8 GB of RAM. Testing functions includes testing monotonous, multimodal, hybrid, and composite functions with 50 dimension, the test function and issue from CEC 2017 were employed. We repeat the process 30 times, paying attention to each job each time, to make sure that all of our tests are continuous and correct. We select the maximum number of repetitions for each experiment, and each experiment has a unique population size (N).

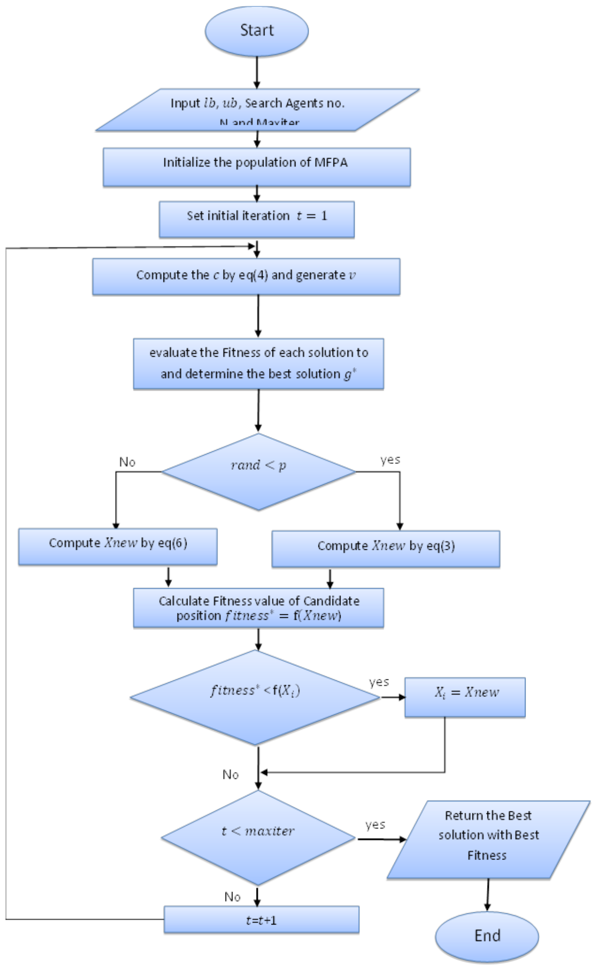

Start
Input lb, ub, Search Agents no. N and Maxiter
Initialize the population of MFPA
Set initial iteration $t = 1$
Compute the c by eq(4) and generate v
evaluate the Fitness of each solution to and determine the best solution g^*
$rand < p$
No
yes
Compute $Xnew$ by eq(6)
Compute $Xnew$ by eq(3)
Calculate Fitness value of Candidate position $fitness^* = f(Xnew)$
$fitness^* < f(X_i)$
yes
$X_i = Xnew$
No
$t < maxiter$
yes
Return the Best solution with Best Fitness
No
t=t+1
End

Figure. 1 (Flowchart of MFPA)

4.1. Performance measurement and parameters

Three criteria will be utilized to assess the performance of the proposed technique in order to verify and demonstrate the efficacy of the MFPA algorithm. The average percentage of improvement in the outcomes makes up the first criterion, which is computed as follows:

$$\mathrm{Mean} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Any advancements will be represented by this statement. The statistic used to calculate the second measure, which is the rate of continuous improvement and the standard deviation of the outcomes of those improvements, is determined as follows:

$$\mathrm{Std} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mathrm{Mean})^2}.$$

Along with the third scale, the mean (MED) crosses in a variety of ways to show how well an algorithm can stabilize. On the scientific front, a stronger, more elastic algorithm performs better when the standard deviation is smaller. The parameter values for MFPA and other algorithms are shown in Appendices A.1 and Appendix A.2 provides details of the CEC2017 benchmark functions. Since each of these algorithms is created to perform optimally in a particular environment given its characteristics, accurate comparisons are possible. Bold is used to highlight the greatest outcomes.

4.2 Numerical Performance Assessment

This displays the findings in tables, emphasizing the best outcomes in bold. The victory (W), tie (T), and loss (L) numbers for each processing unit are displayed in the last row of each table. The exploration and exploitation phases, as well as the capacity to avoid local optimal solutions, were the main areas of comparison between the MFPA algorithm's effectiveness and that of several optimizers in CEC2017 with 50 dimensions. This is done by comparing the MFPA algorithm's overall efficiency against that of other algorithms.

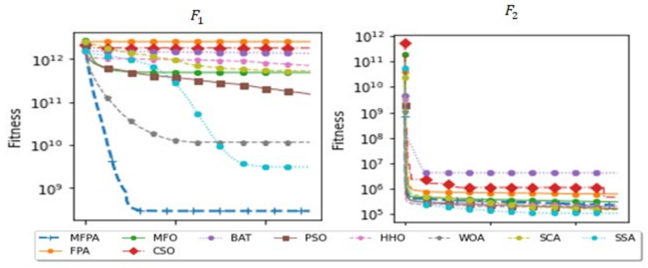


Fig. 2 MFPA for F1 and F2 Convergence curves and other traditional algorithms during 2500 iterations

4.3 Exploitation analyses and exploration capabilities

Single-minimum value functions (F1 and F2) and multimodal optimization functions (F3 and F9), which contain multiple minimum values, were used to evaluate the performance of the proposed algorithm, MFPA. MFPA's astounding exploitation efficiency was demonstrated by the test results on single-minimum-value functions. The results shown in Appendices A.3 show that the algorithm consistently beats competing methods. Additionally, Figure 2 amply demonstrates the convergence of MFPA, demonstrating its superiority to competing strategies.

Turning to the multimodal optimization functions, MFPA showcased its strength in exploration and the ability to handle problems with multiple minimum values. The algorithm's performance on functions F3 and F9 was exceptionally robust, as indicated in Appendices A.4. Furthermore, Figure 3 visually exhibits the convergence of MFPA compared to other algorithms, highlighting its capability to explore the search space and discover new solutions.

MFPA's effectiveness in both exploitation and exploration has been confirmed by evaluations of single-minimum and multimodal optimization functions. The algorithm's ability to reach optimal solutions for single-minimum value functions and its success in tackling multimodal optimization problems underline its strong performance and promising potential for solving a wide range of optimization tasks.

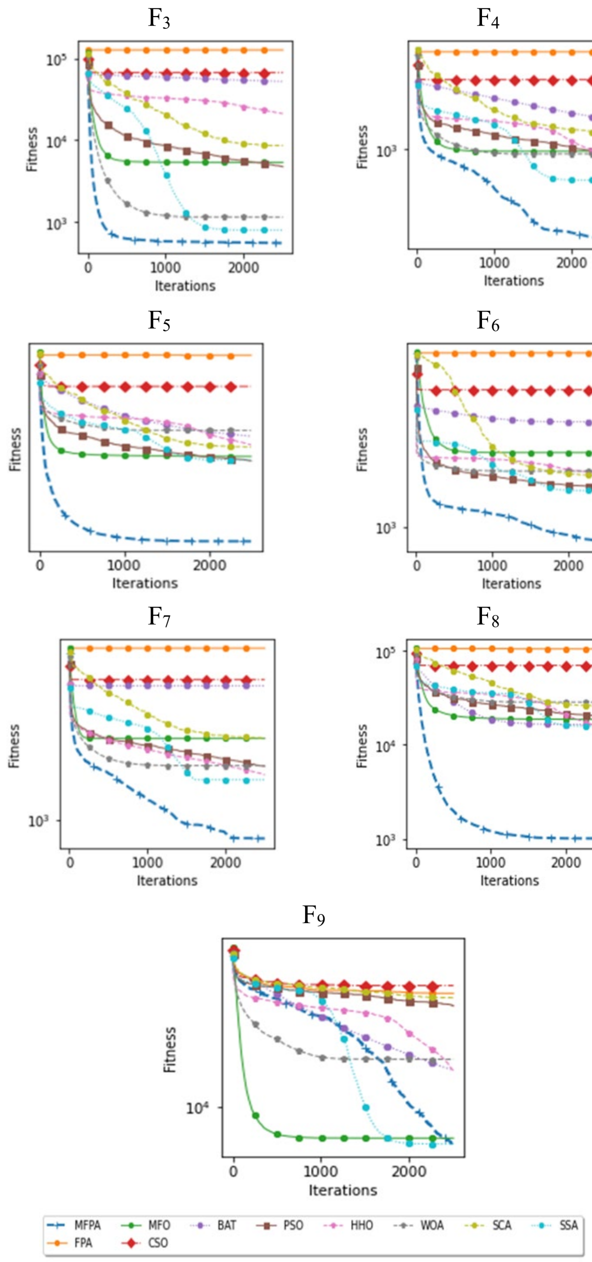


Fig. 3 MFPA for F3 – F9 Convergence curves and other traditional algorithms during 2500 iterations

The suggested algorithm, MFPA, underwent evaluation to determine its capacity to achieve a balance between exploitation and exploration in addition to excelling at handling multimodal optimization issues. This balance is essential because it allows the algorithm to go through the search space quickly and avoid becoming stuck in local optima. A group of functions, F10–F19, created especially for this purpose, were used to analyze this aspect. As shown in Appendices A.5 and Figure 4, the outcomes of testing MFPA on these functions showed that it performed superiorly to other algorithms. These results clearly demonstrate the algorithm's skill in achieving a delicate balance between exploitation and exploration, allowing it to successfully investigate novel solutions while still exploiting the most promising areas of the search field. The fact that MFPA was able to achieve this balance successfully emphasizes its strength and potential as a strong optimization method.

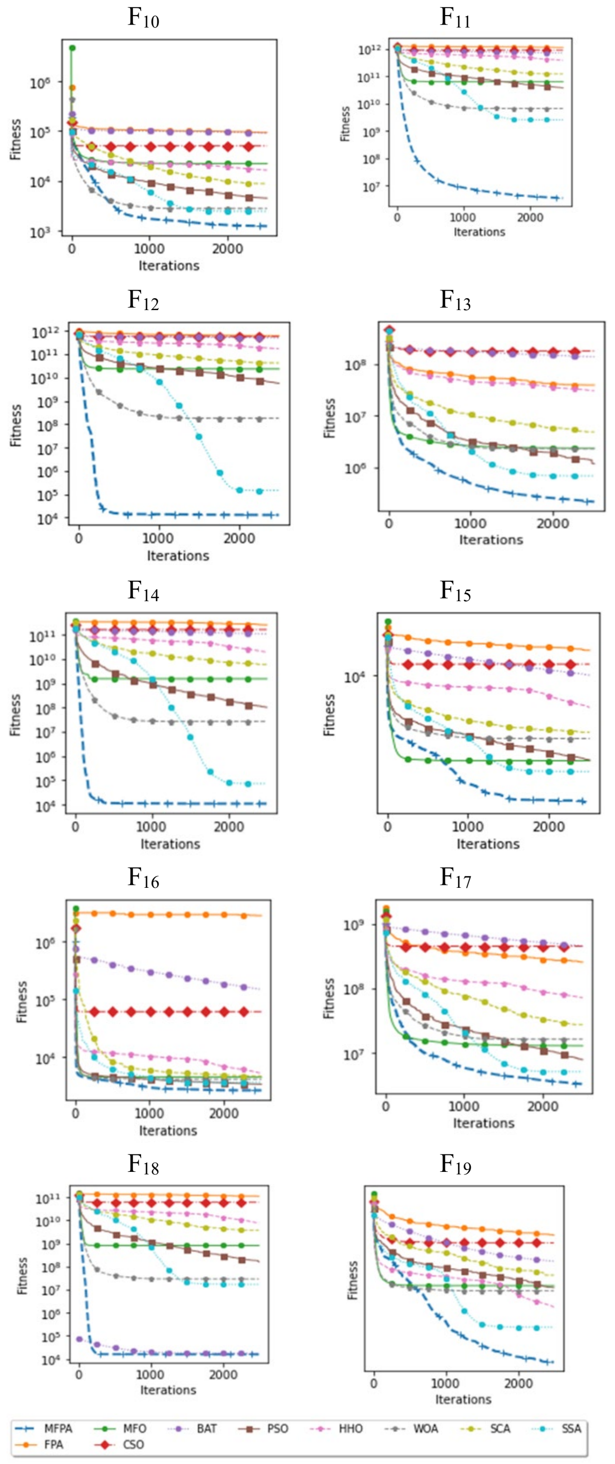
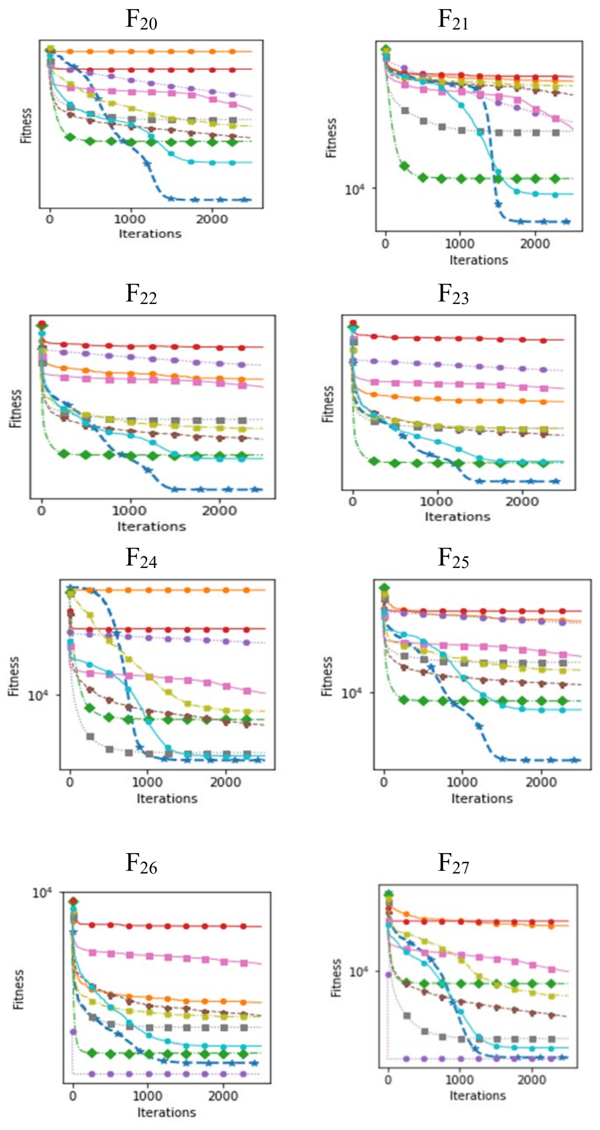


Fig. 4 MFPA for F10 – F19 Convergence curves and other traditional algorithms during 2500 iterations

The effectiveness of the suggested algorithm, MFPA, in handling complex functions must also be evaluated. In order to do this, a set of difficult functions, F20–F29, were carefully picked for assessment. These functions reflect challenging optimization issues that frequently include complicated networks and layered environments. The outcomes of testing MFPA on these functions showed how much more effective it is than competing algorithms, thus enhancing its reputation as a potent optimization method. Appendices A.6 and Figure 5 show the algorithm's affinity and convergence in comparison to other algorithms, providing evidence of its extraordinary performance. The effective handling of complex functions by MFPA demonstrates its capability to deal with practical optimization problems that call for reliable and effective solutions.



F28 F29

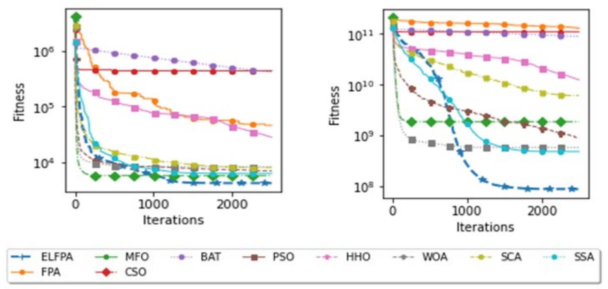


Fig. 5 MFPA for F20 – F29 Convergence curves and other traditional algorithms during 2500 iterations

5. Conclusion and future works

A modified version of the Flower Pollination Algorithm (FPA) has been introduced, known as the Modified Flower Pollination Algorithm (MFPA), which incorporates the Spiral Renewal Mechanism from the Moth Flame Optimization (MFO) algorithm. The MFPA addresses the limitations observed in the original FPA, such as premature convergence and suboptimal exploration-exploitation balance.
By integrating the Spiral Renewal Mechanism from MFO, the MFPA algorithm introduces a novel approach to diversify the search process and enhance exploration capabilities. This mechanism allows the algorithm to dynamically adjust the search trajectory, enabling it to escape local optima more effectively. Moreover, the MFPA algorithm maintains a balance between exploration and exploitation by leveraging the efficient local search mechanisms present in FPA.
To evaluate the performance of the MFPA algorithm, extensive experimentation has been conducted on various benchmark functions, including the widely recognized CEC 2017 test suite. The results of these experiments have been compared against other cutting-edge metaheuristic algorithms. The findings unequivocally demonstrate that the MFPA algorithm outperforms not only the original FPA but also other state-of-the-art methods in terms of convergence speed, ability to avoid local optima, and overall solution efficiency. These results highlight the effectiveness of the Spiral Renewal Mechanism in enhancing the algorithm's exploration capabilities and its potential for handling complex global optimization problems.

The successful application of the MFPA algorithm as a feature selection technique further underscores its adaptability and efficacy in addressing a wide range of optimization problems. As such, the MFPA algorithm holds promise for practical applications and may be further tested and utilized in various real-world scenarios. The encouraging results obtained from this study emphasize the need for continued research and potential enhancements to the MFPA algorithm, fostering advancements in the field of swarm intelligence optimization.

Acknowledgement

The authors would like to express their thanks to the college of computer science and mathematics, Tikrit University, for supporting this research.

References

[1] J. Nocedal, S. J. Wright, J. Nocedal, and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[2] O. Özkaraca, J. Güç, Ü. Optimizasyon, Y. Kullanimina, and İ. İnceleme Öz, "A REVIEW ON USAGE OF OPTIMIZATION METHODS IN GEOTHERMAL POWER GENERATION," *Mugla Journal of Science and Technology*, vol. 130, doi: 10.22531/muglajsci.437340.

[3] J. H. Holland, "Genetic algorithms," *Sci Am*, vol. 267, no. 1, pp. 66–72, 1992, doi: 10.1038/SCIENTIFICAMERICAN0792-66.

[4] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," *Proceedings of the International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995, doi: 10.1109/MHS.1995.494215.

[5] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput Intell Mag*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[6] E. Aarts, J. Korst, and W. Michiels, "Simulated annealing," *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pp. 187–210, 2005, doi: 10.1007/0-387-28356-0_7/COVER.

[7] X. Huang, T. H. Klinge, J. I. L. B, X. Li, and J. H. Lutz, "Unconventional Computation and Natural Computation," Lecture Notes in *Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 7445, pp. 29–40, 2012, doi: 10.1007/978-3-642-32894-7.

[8] Z. A. A. ALYASSERI, A. T. KHADER, M. A. AL-BETAR, M. A. Awadallah, and X. S. Yang, "Variants of the flower pollination algorithm: A review," *Studies in Computational Intelligence*, vol. 744, pp. 91–118, 2018, doi: 10.1007/978-3-319-67669-2_5/COVER.

[9] D. Rodrigues, X. S. Yang, A. N. De Souza, and J. P. Papa, "Binary flower pollination algorithm and its application to feature selection," *Studies in Computational Intelligence*, vol. 585, pp. 85–100, 2015, doi: 10.1007/978-3-319-13826-8_5/COVER.

[10] A. Y. Abdelaziz, E. S. Ali, and S. M. Abd Elazim, "Combined economic and emission dispatch solution using Flower Pollination Algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 264–274, Sep. 2016, doi: 10.1016/J.IJEPES.2015.11.093.

[11] A. Y. Abdelaziz, E. S. Ali, and S. M. Abd Elazim, "Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems," *Energy*, vol. 101, pp. 506–518, Apr. 2016, doi: 10.1016/J.ENERGY.2016.02.041.

[12] O. K. Meng, O. Pauline, S. C. Kiong, H. A. Wahab, and N. Jafferi, "Application of Modified Flower Pollination Algorithm on Mechanical Engineering Design Problem," *IOP conf ser mater sci Eng*, vol. 165, no. 1, p. 012032, Jan. 2017, doi: 10.1088/1757-899X/165/1/012032.

[13] M. N. Kabir, J. Ali, A. R. A. Alsewari, and K. Z. Zamli, "An adaptive flower pollination algorithm for software test suite minimization," *3rd International Conference on Electrical Information and Communication Technology*, EICT 2017, vol. 2018-January, pp. 1–5, Jan. 2018, doi: 10.1109/EICT.2017.8275215.

[14]U. Singh and R. Salgotra, "Synthesis of linear antenna array using flower pollination algorithm," *Neural Comput Appl*, vol. 29, no. 2, pp. 435–445, Jan. 2018, doi: 10.1007/S00521-016-2457-7/METRICS.

[15]M. Abdel-Basset and L. A. Shawwky, "Flower pollination algorithm: a comprehensive review," *Artif Intell Rev*, vol. 52, no. 4, pp. 2533–2557, Dec. 2019, doi: 10.1007/S10462-018-9624-4/METRICS.

[16]Y. Chen and D. Pi, "An innovative flower pollination algorithm for continuous optimization problem," *Appl Math Model*, vol. 83, pp. 237–265, Jul. 2020, doi: 10.1016/J.APM.2020.02.023.

[17]P. E. Mergos and X. S. Yang, "Flower pollination algorithm parameters tuning," *Soft comput*, vol. 25, no. 22, pp. 14429–14447, Nov. 2021, doi: 10.1007/S00500-021-06230-1/TABLES/10.

[18] M. Abdel-Basset, R. Mohamed, S. Saber, S. S. Askar, and M. Abouhawwash, "Modified Flower Pollination Algorithm for Global Optimization," Mathematics 2021, Vol. 9, Page 1661, vol. 9, no. 14, p. 1661, Jul. 2021, doi: 10.3390/MATH9141661.

[19] P. E. Mergos and X. S. Yang, "Flower pollination algorithm with pollinator attraction," *Evol Intell*, pp. 1–17, Jan. 2022, doi: 10.1007/S12065-022-00700-7/METRICS.

[20] K. M. Ong, P. Ong, and C. K. Sia, "A new flower pollination algorithm with improved convergence and its application to engineering optimization," *Decision Analytics Journal*, vol. 5, p. 100144, Dec. 2022, doi: 10.1016/J.DAJOUR.2022.100144.

[21] F. Seghir and G. Khababa, "An improved discrete flower pollination algorithm for fuzzy QoS-aware IoT services composition based on skyline operator," *Journal of Supercomputing*, pp. 1–32, Feb. 2023, doi: 10.1007/S11227-023-05074-W/METRICS.

[22] R. Mallipeddi, P. S.-N. T. University, and undefined 2010, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," *al-roomi.org*, 2010, Accessed: May 28, 2023. [Online]. Available: http://www.al-roomi.org/multimedia/CEC_Database/CEC2010/RealParameterOptimization/CEC2010_RealParameterOptimization_TechnicalReport.pdf

[23] X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," *Studies in Computational Intelligence*, vol. 284, pp. 65–74, 2010, doi: 10.1007/978-3-642-12538-6_6.

[24] R. Cheng, Y. J.-I. transactions on cybernetics, and undefined 2014, "A competitive swarm optimizer for large scale optimization," *ieeexplore.ieee.org*, Accessed: May 28, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6819057/

[25] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowl Based Syst*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/J.KNOSYS.2015.12.022.

[26] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/J.ADVENGSOFT.2016.01.008.

[27] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl Based Syst*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/J.KNOSYS.2015.07.006.

[28] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris Hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/J.FUTURE.2019.02.028.

[29] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/J.ADVENGSOFT.2017.07.002

Appendix A

Appendix A.1 Hyper Parameter Settings

Algorithm	**Parameter**	**Range\Value**
FPA	p Coefficient c_1	[0, 1] [2/e, 2]
FPA	p	[0, 1]
MFO	T	[-1, 1]
CSO	Phi	0
.BAT	Loudness, Pulse rate Frequency minimum Frequency maximum	0.5, 0.5 0 2
PSO	Inertia weight (wmin, wmax) Cognitive coefficient	0.04, 0.09 2
HHO	Beta	1.5
WOA	Convergence constant (a) [0, 2] Coefficient (b) 1	2
SCA	Convergence constant r_1	[0, 2]
SSA	Coefficient c_1	[2/e, 2]

Appendix A.2 CEC 2017 Benchmark Function.

Type	**Fun**	**Function name**	**fmin**
U	F1	Shifted and rotated bent cigar function	100
U	F2	Shifted and rotated Zakharov	300
M	F3	Shifted and rotated ROSENBROCKS	400
M	F4	Shifted and rotated restrains	500
M	F5	Shifted and rotated expanded scoffers f6	600
M	F6	Shifted and rotated Lunacek bi -restrains	700
M	F7	Shifted and rotated non-continuous restrains	800
M	F8	Shifted and rotated levy	900
M	F9	Shifted and rotated Schwefels	1000
H	F10	Hybrid function 1 (n=3)	1100
H	F11	Hybrid function 2 (n=3)	1200
H	F12	Hybrid function 3 (n=3)	1300
H	F13	Hybrid function 4 (n=4)	1400
H	F14	Hybrid function 5 (n=4)	1500
H	F15	Hybrid function 6 (n=4)	1600
H	F16	Hybrid function 6 (n=5)	1700
H	F17	Hybrid function 6(n=5)	1800
H	F18	Hybrid function 6 (n=5)	1900
H	F19	Hybrid function 6(n=6)	2000
C	F20	Composition function 1 (n=3)	2100
C	F21	Composition function 2 (n=3)	2200
C	F22	Composition function 3 (n=4)	2300
C	F23	Composition function 4 (n=4)	2400
C	F24	Composition function 5 (n=5)	2500
C	F25	Composition function 6 (n=5)	2600
C	F26	Composition function 7 (n=6)	2700
C	F27	Composition function 8 (n=6)	2800
C	F28	Composition function9 (n=3)	2900
C	F29	Composition function 10 (n=3)	3000

Appendix A.3 Results of Unimodal functions for the MFPA vs some recent techniques.

Fun	Criteria	CSO	SSA	PSO	WOA	BAT	HHO	SCA	MFO	FPA	MFPA
F1	ave	1.81E+12	3.08E+09	1.52E+11	1.16E+10	1.36E+12	7.10E+11	5.28E+11	4.88E+11	2.55E+12	**2.91E+08**
	std	2.192E+11	2.537E+09	4.952E+10	4.472E+09	3.237E+10	7.922E+10	5.719E+10	1.969E+11	3.180E+11	1596452184
	med	1.779E+12	3.430E+09	1.549E+11	1.010E+10	7.765E+10	7.318E+11	5.219E+11	5.181E+11	2.589E+12	8480.589518
F2	ave	469217.6	**109430.5**	163000.6	193137.4	4181579	203536.7	145726.5	300850.9	615757.4	224611.5
	Std	1.585E+06	2.506E+04	2.967E+04	6.001E+04	1.505E+04	2.757E+04	2.341E+04	1.047E+05	7.499E+04	37263.57566
	med	3.806E+05	1.095E+05	1.616E+05	1.825E+05	8.602E+04	2.003E+05	1.420E+05	2.793E+05	6.229E+05	223758.7345
rank	W/T/L	0/0/2	1/0/1	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2	1/0/1

Appendix A.4 Results of Multi-unimodal functions for the MFPA vs some recent techniques.

Fun	Criteria	CSO	SSA	PSO	WOA	BAT	HHO	SCA	MFO	FPA	MFPA
F3	ave	66998.4	800.18	4753.17	1153.63	52324.62	21138.93	8630.45	5358.94	126831.3	**554.69**
	std	1.402E+04	9.771E+01	1.296E+03	1.460E+02	4.641E+02	4.230E+03	1.756E+03	3.475E+03	2.009E+04	44.26465708
	med	6.602E+04	7.918E+02	4.930E+03	1.123E+03	1.190E+03	2.077E+04	8.607E+03	4.342E+03	1.303E+05	547.1299557
F4	ave	1464.14	846.29	973.99	976.4	1176.15	951.15	1091.82	990.8	1704.29	**614.62**
	std	6.610E+01	7.137E+01	6.156E+01	7.472E+01	4.057E+01	3.603E+01	3.351E+01	9.151E+01	7.674E+01	72.35156842
	med	1.486E+03	8.457E+02	9.729E+02	9.589E+02	7.339E+02	9.484E+02	1.095E+03	9.703E+03	1.713E+03	592.533494
F5	ave	769.98	685.59	684.8	718.45	711.76	702.1	699.78	689.39	808.64	**603**
	std	1.194E+01	1.113E+01	8.197E+00	1.568E+01	1.041E+01	6.522E+00	5.907E+00	1.282E+01	1.147E+01	2.117938714
	med	7.734E+02	6.834E+02	6.856E+02	7.190E+02	6.340E+02	7.025E+02	6.981E+02	6.871E+02	8.092E+02	602.4210022
F6	ave	4193.63	1461.41	1514.89	1791.32	3002.68	1749.65	1707.13	2173.39	6170.34	**855.13**
	std	2.826E+02	1.626E+02	7.016E+01	9.950E+01	1.004E+02	5.799E+01	7.866E+01	5.200E+02	3.970E+02	61.58220351
	med	4.111E+03	1.435E+03	1.496E+03	1.780E+03	1.098E+03	1.773E+03	1.714E+03	2.121E+03	6.237E+03	841.4950511
F7	ave	1790.8	1182.11	1250.4	1253.28	1745.96	1205.97	1405.86	1403.36	2040.57	**926.42**
	std	7.072E+01	7.845E+01	5.276E+01	5.448E+01	8.739E+01	3.710E+01	3.010E+01	7.913E+01	8.556E+01	47.8696701
	med	1.774E+03	1.179E+03	1.246E+03	1.255E+03	1.037E+03	1.200E+03	1.411E+03	1.373E+03	2.051E+03	919.7508851
F8	ave	70048.41	15780.9	19896.09	28484.07	16486.89	14882.34	25728.7	18673.84	104988	**1007.98**
	std	1.002E+04	3.043E+03	3.964E+03	8.005E+03	4.102E+03	1.171E+03	4.033E+03	5.450E+03	1.333E+04	110.9974115
	med	7.268E+04	1.459E+04	2.026E+04	2.641E+04	1.189E+04	1.463E+04	2.542E+04	1.782E+04	1.043E+05	961.4466083
F9	ave	15720.22	8708.89	14571.83	11940.56	11469.7	11395.55	15019.39	8900.26	15272.34	**8632.6**
	std	5.936E+02	8.483E+02	7.082E+02	1.400E+03	2.588E+03	1.303E+03	3.663E+02	1.228E+03	5.108E+02	2745.167926
	med	1.578E+04	8.008E+03	1.469E+04	1.196E+04	7.076E+03	1.111E+04	1.506E+04	8.895E+03	1.529E+04	7624.239994
rank	W/T/L	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7	0/0/7	7/0/0

Appendix A.5 Results of Hybrid functions for the MFPA vs some recent techniques.

Fun	Criteria	CSO	SSA	PSO	WOA	BAT	HHO	SCA	MFO	FPA	MFPA
F10	ave	50469.2	2449.69	4504.52	2782.58	92218.77	16365.74	8828.7	22276.62	93327.88	**1232.74**
	std	1.426E+04	5.261E+02	1.191E+03	6.232E+02	1.923E+03	2.679E+03	1.956E+03	1.646E+04	1.596E+04	65.95750678
	med	4.817E+04	2.402E+03	4.120E+03	2.677E+03	4.662E+03	1.719E+04	8.579E+03	1.725E+04	9.404E+04	1218.666124
F11	ave	9.00E+11	2.56E+09	3.76E+10	6.61E+09	7.07E+11	3.84E+11	1.22E+11	6.28E+10	1.13E+12	**3503324**
	std	2.132E+11	2.641E+09	1.675E+10	3.649E+09	1.301E+10	1.112E+11	2.931E+10	4.455E+10	1.478E+11	1919440.203
	Med	8.373E+11	1.834E+09	3.448E+10	5.755E+09	5.699E+09	3.967E+11	1.243E+11	4.834E+10	1.160E+12	3217230.612
F12	ave	5.67E+11	147799.2	5.64E+09	1.8E+08	4.99E+11	1.70E+11	4.21E+10	2.33E+10	6.13E+11	**12939.3**
	std	2.047E+11	6.688E+04	3.459E+09	2.499E+08	1.131E+09	1.173E+11	1.563E+10	2.875E+10	7.107E+10	10867.26593
	med	5.895E+11	9.423E+04	4.518E+09	1.125E+08	1.139E+09	1.409E+11	3.750E+10	7.490E+09	6.323E+08	10903.50253
F13	ave	1.8E+08	687403	1178547	2291624	1.39E+08	30408952	4859968	2359651	39310857	**216426.3**
	std	9.520E+07	7.672E+05	1.201E+06	1.707E+06	8.484E+05	3.166E+07	3.373E+06	4.014E+06	1.400E+07	185022.8454
	med	1.040E+08	5.727E+05	6.766E+05	2.127E+06	6.438E+05	1.999E+07	3.737E+06	1.033E+06	3.675E+07	162895.4925
F14	ave	1.64E+11	74358.9	1.03E+08	26692289	1.08E+11	1.94E+10	6E+09	1.53E+09	2.55E+11	**11012.08**
	std	6.306E+10	3.138E+04	9.247E+07	4.638E+07	1.330E+09	1.468E+10	2.984E+09	2.910E+09	6.767E+10	7145.801918
	med	1.464E+11	5.368E+04	6.167E+07	6.974E+06	5.678E+06	1.708E+10	5.575E+09	3.652E+05	2.625E+11	9059.879487
F15	ave	11158.38	4037.96	4501.68	5515.57	10049.51	7402.1	5858.66	4471.31	12723.78	**2977.82**
	std	1.748E+03	5.624E+02	6.527E+02	8.221E+02	5.171E+02	1.753E+03	4.389E+02	5.127E+02	8.793E+02	523.5996841
	med	1.087E+04	3.842E+03	4.574E+03	5.421E+03	3.145E+03	7.033E+03	5.916E+03	4.468E+03	1.281E+04	2804.373496
F16	ave	61252.86	3683.91	3426.7	4192.94	148244.1	5216.22	4709.87	4534.96	2791807	**2679.87**
	std	1.634E+05	3.799E+02	3.704E+02	4.726E+02	2.495E+02	1.085E+03	2.911E+02	1.526E+03	1.943E+06	330.496663
	med	7.325E+04	3.568E+03	3.375E+03	4.113E+03	2.855E+03	4.952E+03	4.716E+03	4.182E+03	2.403E+06	2683.409635
F17	ave	4.47E+08	5141403	7889465	16328707	4.53E+08	71212865	27543217	12940008	2.54E+08	**3296114**
	std	2.342E+08	4.167E+06	5.373E+06	1.274E+07	1.130E+07	4.117E+07	1.397E+07	1.501E+07	9.834E+07	2600313.618
	med	3.408E+08	4.547E+06	6.203E+06	1.167E+07	3.500E+06	5.392E+07	2.351E+07	9.159E+06	2.455E+08	2633811.571
F18	ave	6.1E+10	17053613	1.57E+08	29073092	17754.6	7.66E+09	3.74E+09	8.19E+08	1.12E+11	**15838.77**
	std	2.695E+10	1.984E+07	2.194E+08	6.049E+07	8.186E+07	8.203E+09	1.666E+09	2.304E+09	2.759E+10	12871.70277
	med	7.234E+10	8.922E+06	6.868E+07	1.058E+07	5.009E+06	6.083E+09	3.407E+09	3.988E+07	1.187E+11	11354.11146
F19	ave	4560.53	3255.36	3812.37	3765.37	4237.13	3521.72	4007.32	3843.33	4703.76	**2831.21**
	std	2.392E+02	2.869E+02	3.241E+02	3.512E+02	4.889E+02	2.806E+02	1.611E+02	2.922E+02	1.384E+02	319.6565146
	med	4.499E+03	3.235E+03	3.884E+03	3.813E+03	3.144E+03	3.596E+03	4.007E+03	3.797E+03	4.734E+03	2814.648489
rank	w/t/l	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	10/0/0

Appendix A.6 Results of Composition functions for the MFPA vs some recent techniques.

Fun	**Criteria**	**CSO**	**SSA**	**PSO**	**WOA**	**BAT**	**HHO**	**SCA**	**MFO**	**FPA**	**MFPA**
F20	ave	3394.65	2631.47	2813.24	2959.14	3150.5	3035.39	2908.03	2787.96	3561.62	**2412.75**
	std	1.339E+02	6.514E+01	4.596E+01	1.013E+02	5.125E+01	8.516E+01	4.460E+01	7.420E+01	1.004E+02	17.29685119
	Med	3.413E+03	2.646E+03	2.819E+03	2.953E+03	2.530E+03	3.014E+03	2.903E+03	2.781E+03	3.576E+03	2397.808003
F21	ave	17427.72	9703.5	15929.13	13273.57	13899.39	13398.6	16654.28	10489.39	17047.9	**9690.21**
	std	6.786E+02	1.850E+03	1.745E+03	1.324E+03	2.528E+03	1.208E+03	4.319E+02	1.010E+03	3.105E+04	2811.272655
	med	1.746E+04	1.031E+04	1.639E+04	1.348E+04	8.790E+03	1.333E+04	1.674E+04	1.068E+04	1.704E+04	9105.600679
F22	ave	4962.48	3182.71	3442.9	3721.23	4617.8	4233.77	3590	3230.58	4369.76	**2844.28**
	std	3.582E+02	9.997E+01	8.097E+01	1.883E+02	9.548E+01	2.132E+02	7.492E+01	7.564E+01	7.429E+01	24.64406587
	med	4.972E+03	3.147E+03	3.451E+03	3.751E+03	2.989E+03	4.207E+03	3.578E+03	3.219E+03	4.366E+03	2849.354676
F23	ave	5568.49	3264.89	3663.53	3776.15	4863.85	4496.28	3774.79	3245.59	4246.09	**3029.14**
	std	5.384E+02	9.187E+01	7.984E+01	1.723E+02	1.376E+02	2.405E+02	6.064E+01	5.091E+01	5.547E+01	61.79476527
	med	5.434E+03	3.270E+03	3.653E+03	3.773E+03	3.165E+03	4.486E+03	3.768E+03	3.243E+03	4.253E+03	3014.997016
F24	ave	32360.55	3308.2	5747.74	3486.85	25086.01	10165.09	7382.41	6354.31	64870.05	**3034.77**
	std	6.108E+03	8.815E+01	6.977E+02	1.335E+02	4.286E+02	9.735E+02	7.762E+02	3.849E+03	1.022E+04	31.45844895
	med	7.154E+04	6.000E+03	1.169E+04	5.534E+03	4.604E+04	1.981E+04	1.746E+04	1.161E+04	6.396E+04	3037.865467
F25	ave	24868.14	8193.6	10853.3	13984.83	21653.95	14922.57	12815.13	9048.23	22207.85	**5136.13**
	std	2.820E+0+	2.675E+03	7.221E+02	1.215E+03	7.765E+02	7.011E+02	4.847E+02	8.458E+02	9.439E+02	629.7388736
	med	2.562E+04	8.475E+03	1.081E+04	1.417E+04	6.629E+03	1.479E+04	1.277E+04	8.941E+03	2.235E+04	4998.271063
F26	ave	8032.7	3809.11	4606.11	4285.97	**3200.01**	6350.56	4578.91	3646.24	5001.54	3355.8
	std	1.113E+03	1.688E+02	1.867E+02	4.691E+02	1.018E+02	9.094E+02	1.743E+02	1.230E+02	2.435E+02	61.15253205
	med	8.130E+03	3.864E+03	4.684E+03	4.151E+03	3.640E+03	6.339E+03	4.608E+03	3.658E+03	4.992E+03	3355.716763
F27	ave	18662.53	3790.64	5610.76	4264.04	**3300.01**	9890.64	7309.72	8507.42	17611.1	3309.48
	std	2.195E+03	2.585E+02	5.386E+02	2.741E+02	5.211E+02	8.839E+02	6.693E+02	1.117E+03	1.165E+03	16.88030016
	med	1.847E+04	3.751E+03	5.717E+03	4.245E+03	40404E+03	9.849E+03	7.295E+03	8.865E+03	1.764E+04	3307.970002
F28	ave	437524.5	6309.82	7034.7	8313.3	412148.1	28187.87	8006.84	5752.27	46046.67	**3804.21**
	std	1.062E+06	7.372E+02	8.103E+02	9.872E+02	3.547E+02	2.448E+04	8.454E+02	6.151E+02	3.324E+04	358.8508676
	Med	2.641E+05	6.215E+03	7.157E+03	8.264E+03	4.812E+03	1.969E+04	7.943E+03	5.622E+03	3.178E+04	3791.541901
F29	ave	1.09E+11	4.84E+08	8.97E+08	5.83E+08	8.86E+10	1.25E+10	6.1E+09	1.87E+09	1.29E+11	**1332612**
	std	4.397E+10	2.303E+08	5.753E+08	3.097E+08	2.399E+08	7.920E+09	1.983E+09	3.894E+09	2.765E+10	453765.7117
	med	9.897E+10	5.117E+08	7.147E+08	5.299E+08	3.399E+08	9.554E+09	6.234E+09	1.181E+08	1.350E+11	1259587.299
rank	w/t/l	0/0/10	0/0/10	0/0/10	0/0/10	2/0/8	0/0/10	0/0/10	1/0/9	0/0/10	8/0/2