A Modified Firefly Algorithm for Scheduling: Optimization Strategy with Objective Function Filtering

Arcely Perez-Napalit, Melvin A. Ballera

[1,2]College of Information and Technology Education, Technological Institute of the Philippines, Manila, Philippines
Email: manapalit@tip.edu.ph, melvin.ballera@tip.edu.ph

ABSTRACT

This study evaluates the Modified Firefly Algorithm (MFA), an advanced optimization technique that integrates processes like Constraint Satisfaction, Hamming Distance, objective function filtering, and positional adjustment. The MFAs were assessed based on solution quality, computational efficiency, and performance in complex optimization problems. A meta-optimization process fine-tuned its parameters, improving solution quality. Despite longer computation times, the MFA outperformed other Firefly Algorithm variants in solution quality and adaptability. The study demonstrated the MFA's effectiveness in a real-world staff scheduling scenario, making it promising for similar problems in various domains. Despite challenges in computation time, the study underscores the trade-off for enhanced solution quality and the potential for further refinement and highlights the benefits of enhanced solution quality and recommends exploring varying n values and comparing them with other advanced optimization algorithms for future research.

Keywords : Bio-inspired algorithms, Hybrid algorithms, Meta-optimization, Metaheuristic algorithms, Swarm Intelligence

1. INTRODUCTION

THE significance of optimization in various fields emphasizes the use of optimization algorithms to find optimal solutions to complex problems. These algorithms address real-world challenges by evaluating objective functions and seeking the best solutions. There are research papers focus on modelling real-world applications and innovative techniques for solving optimization problems using various approaches, including heuristic search, finite methods, intelligent algorithms (such as genetic algorithms, swarm intelligence, and artificial neural networks), operations research techniques, and agent-based procedures [1]. On the other hand, operations research techniques involve mathematical models applied to practical problems, while agent-based procedures utilize multiple agents interacting to solve problems effectively [2]. These intelligent algorithms leverage heuristics and machine learning to efficiently find optimal solutions [3]. Nature-inspired algorithms, such as Swarm Intelligence (SI) algorithms, have gained popularity for their ability to locate near-optimal solutions in communication networks, routing, and scheduling. SI algorithms, which include organism-based (e.g., fireflies, bees) and entity-based (e.g., birds, wolves) approaches, offer advantages like ease of implementation, high convergence rates, and scalability [4], [5]. SI algorithms effectively deal with uncertainty and imprecision in real-world optimization problems [6] and [7]. Among the SI algorithms, the Firefly Algorithm (FA), inspired by the attractive and synchronizing behaviour of fireflies, has succeeded in optimization tasks and has been widely utilized in various applications [8]. The Firefly Algorithm (FA) is an optimization technique introduced by Yang in 2008, inspired by fireflies' attractive and synchronizing behaviour. It utilizes the concept of firefly luminosity, where fireflies are attracted to brighter ones and move randomly when no brighter fireflies are nearby. This brightness influences the optimization's objective function, and the algorithm's process involves initializing a population, evaluating objective function values, updating attractiveness, and moving the fireflies based on this updated attractiveness to find optimal solutions. It has been successful and widely utilized in various optimization tasks [9]. The FA process is presented in Fig. 1.

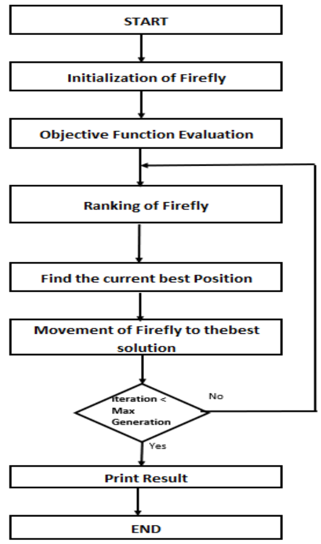


Fig. 1. Firefly Algorithm Process [36]

Each firefly represents a potential solution to the problem, and its brightness corresponds to the quality of that solution as determined by the objective function. Fireflies move and adjust their solutions based on the brightness of other fireflies, seeking optimal or near-optimal solutions to complex optimization problems. The attractiveness between fireflies is controlled by a parameter β, influenced by the distance between them. The algorithm's movement is determined by the level of attraction between fireflies, ensuring exploration and convergence toward better solutions. The algorithm incorporates various factors, including light intensity, movement, and distance between fireflies, to achieve comprehensive optimization results [8]. The Firefly Algorithm can be expressed in Eq. 1, which considers the fireflies' attractive light intensity, their movement based on attraction, and their distance from one another. The equation encompasses the factors of fireflies' attractiveness, movement, and distance and presents a comprehensive comparison of the Firefly Algorithm [8]. In the FA, the attractiveness of firefly *j* to firefly *i* is determined using Eq.2, which considers the brightness intensity of each firefly. The objective function *f(x)* represents the quality of a particular solution and influences the algorithm's behavior. Eq. 3 shows the relationship between the brightness of a firefly and its corresponding objective function value. The attractiveness between fireflies is controlled by the parameter β and is influenced by the distance between fireflies, represented by *r*. The equation that governs the attraction between fireflies can be expressed as Eq. 4. The distance between two fireflies *i* and *j* is important in determining their attractiveness in the algorithm. This distance is calculated using the Cartesian equation presented in Eq.5. The movement of fireflies is determined by their level of attraction, which is based on the brightness of firefly j relative to firefly i. If firefly j is brighter than firefly i, the movement of fireflies is given by Eq. 6.

$$X_{id}(t+1) = X_{id}(t) + \beta o^{e} - \gamma^{r_{ij}} (X_{jd}(t) - (X_{id}(t)) + \epsilon \quad (1)$$

$$I(r) = \frac{I_s}{r^s} \quad (2)$$

$$I_i = f(X_i) \quad (3)$$

$$\beta(r) = \beta o^{e} - \gamma^{r_{ij}} \quad (4)$$

$$r_{ij} = \sqrt{\sum_{k=1}^{D} (X_{ik} - X_{jk})^2} \quad (5)$$

$$\propto . S_k \left(rand_{ik} - \frac{1}{2} \right) \quad (6)$$

The Firefly Algorithm (FA) is renowned for its simplicity and versatility in solving diverse optimization problems. It balances exploration and exploitation effectively, making it proficient in discovering local and global optima [10], [11]. As a stochastic algorithm, FA uses randomization to avoid being trapped in local optima, making it a powerful tool for complex optimization problems [12]. However, FA faces challenges in high-dimensional problems, leading to extended computation times and a potential need for additional optimization techniques. Sensitivity to parameter selection and high computational cost are additional limitations [13], [14]. The randomization process in FA may limit solution space exploration and impact performance, particularly in high-dimensional problems [15]. Furthermore, its effectiveness depends on problem nature, and it may struggle with premature convergence and multiple optima in complex objective functions [10]. Various studies have proposed modifications to overcome FA's limitations, including adjusting the attractiveness function, using hybrid techniques, and incorporating domain-specific knowledge. These enhancements promise to improve FA's performance and overcome challenges [8]. The proposed modifications significantly benefit solving complex problems, including resource allocation and scheduling. By comparing the modified algorithm with standard and hybrid versions, potential improvements can be identified, advancing optimization strategies, and improving outcomes in critical applications [11].

Various researchers have successfully applied FA to address diverse optimization problems. [16] used FA for economic emissions and dispatch problems. [17], [18] and [19] applied FA to the Job Shop Scheduling Problem (JSSP) make span time and flowtime for different workloads. [20] utilized FA for cloud computing job scheduling. [21] for disjunctive graph for optimal solutions, [22] for work scheduling, FA has also proven successful in staffing and timetabling scheduling tasks. [23] for patient appointment scheduling, [24] and [25], [26] for optimized nurse scheduling, [27][28] for staff shift lists.

Various performance measures have been used to compare FA with other algorithms. Comparison of FA with Ant Colony (AC), Firefly Algorithm, Cuckoo Search Algorithm, and Bat Algorithm – regarding running time and success rate [29], convergence rate of FA versus Genetic Algorithm (GA) [25], comparison of FA, Genetic Algorithms and Particle Swarm Optimization for nurse scheduling [6], another nurse scheduling using FA, GA and Simulated Annealing (SA), comparison of AC, FA, potential bacterial field and PSO for route-finding [13], FA, Cuckoo Algorithm (CA) and Artificial Bee Colony (ABC) for multimodal optimization [30] and FA and SA for the nurses scheduling [19]. Overall, FA demonstrates potential in its performance with different case scenarios.

Hybrid meta-heuristic algorithms have gained popularity as practical approaches to enhance algorithm performance [31]. By combining multiple algorithms, these hybrid methods efficiently solve problems and improve the original algorithms' performance while retaining their desired features [23]. FA has been subject to various enhancements, including hybridization techniques, to obtain high-quality solutions surpassing previously proposed algorithms [32]. For instance, [33] proposed a hybrid FA and GA for nurse scheduling and FA with PSO [345], FA and CS for engineering optimization tasks [35], ABC and FA for course timetabling [36], FA and Variable Neighbourhood Search (VNS) for scheduling [37], and FA with SA for nurse scheduling [10] and [4] FA and TS method for hospital scheduling. These studies demonstrate the potential of hybridizing FA with other algorithms to address complex optimization challenges efficiently.

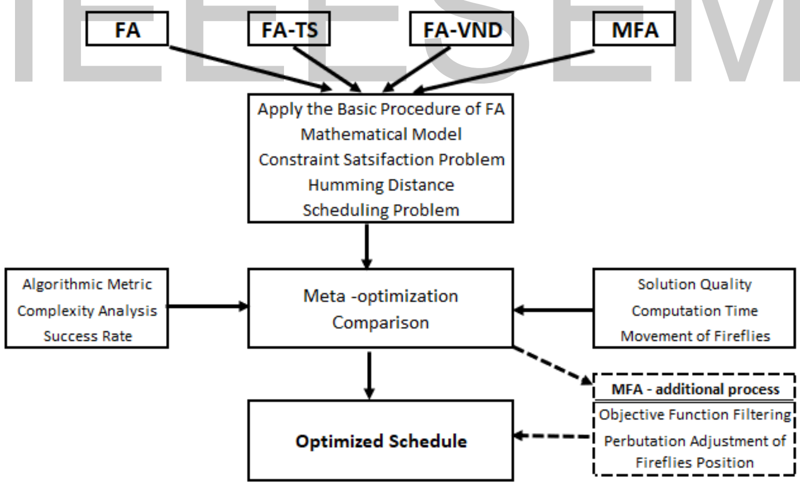


Fig. 2. Conceptual Framework of the Study

The study aimed to develop and evaluate a Modified Firefly Algorithm (MFA) tailored for solving scheduling problems. The MFA integrated optimization techniques to enhance search performance and discover optimal or near-optimal solutions. The objectives included evaluating the algorithm's acceptability regarding time and space complexity, readability, and generality. Additionally, the study measured the optimization process's performance in solution quality, success rate, performance rate, computation time, and parameter setting. Lastly, the research sought to create a proof of concept in the scheduling field, demonstrating the MFA's practical applicability for optimizing scheduling tasks in various domains.

2. METHODS

The study required two primary materials: a software tool or programming language and accurate datasets representing scheduling problems. MATLAB R2022a was chosen as the software tool due to its robust and efficient features for implementing complex algorithms and generating high-quality solutions. The hardware used was a computer system with Intel(R) Core™-i5-7400 PC CPU @3.00Ghz, 8.00GB RAM, x64-based processor, and Windows 10 Pro operating system to ensure practical implementation and efficient solution of the optimization model. Real-world scheduling data from the case study organization was used, offering valuable insights into actual scheduling practices and operations, enhancing the credibility and relevance of the research to real-world scenarios, and contributing to the field of scheduling optimization.

2.1. Procedure for Optimization

To ensure the accuracy of the experimental computation, the researcher followed the Constrained Optimization Good Practice Checklist by [38], which includes two main sections: modelling and optimization stages. The checklist helped structure the problem, formulate the mathematical model, select the best method, analyze algorithm performance, evaluate the solution, and use it for decision-making. Using this checklist, the researcher ensured the proper conduct of the optimization process and enhanced the reliability and validity of the study's findings.

2.2. Modelling Stage

The study chooses a model that fits the problem's nature, available data, and research goals. This involves transforming the real-world scheduling problem into a mathematical formulation with which optimization algorithms can work. The goal is to create a well-defined and accurate mathematical model representing the problem and its variable relationships, increasing the likelihood of finding an optimal solution.

2.2.1. Problem Structuring

In this study, it is necessary to determine if the policies can be quantified within the constraints of the decision variables to meet the optimization objective. For the study, the schedule dataset of the Renal Department's Technicians (T-staff) was used to create schedules, considering various shifts and day patterns. Defining the problem structure, including hard and soft constraints, is essential to managing a solution that satisfies the objective function. The study has identified the following policies for designing a schedule in the case study:

- All technician staff members are assigned to the same department.
- Technician staff must have either 0 (days off) or 1 (working) shift per day.
- A scheduling cycle consists of 30 days.
- Each day's schedule includes four shifts of eight hours: dawn, morning, afternoon, and evening.
- Technician staff members are allowed a maximum of 22 working shifts within one scheduling cycle.
- Each technician staff member must work one Sunday duty shift within the one-cycle schedule.
- Technicians can work at most two consecutive evening shifts.
- The department requires a minimum of six and seven technician staff members from Monday to Saturday, with a maximum of two on duty on Sunday.
- Technicians have the option to request two consecutive days of working shifts.
- Technicians can also request two consecutive days off.

2.2.2. Mathematical Formulation

The study utilized various indexes, data, and variables to facilitate the optimization process in creating schedules for the technician staff. Information about staff requirements, potential shifts, and daily patterns was used to formulate a mathematical model that generated schedules meeting the outlined policies while satisfying constraints.

Indexes: i referred to technicians (i=1,2,…,ntechnicians)., j referred to shift and k referred to the day (k=1,2,…,ndays).
Constants (data): S set of indexes corresponding to senior technicians, O set of indexes corresponding to orderly technicians and D_{sum} set of indexes (dates) corresponding to Sundays. Eq. 7 and Eq.8 presents the formulation of constant data.

$$CDO_{ik} = \begin{cases} 1, if\ technician\ i\ requested\ a\ Consecutive\ Day\ Off\ at\ day\ k \\ 0, oherwise \end{cases} \quad (7)$$

$$CS_{ijk} = \begin{cases} 1, if\ technician\ i\ requested\ a\ Consecutive\ Shift\ j\ at\ da \\ 0, oherwise \end{cases} \quad (8)$$

Decision variables:

$$X_{ijk} = \begin{cases} 1, if\ technician\ i\ will\ work\ on\ shift\ j\ on\ day\ k \\ 0, oherwise \end{cases} \quad (9)$$

A free day is a day which a technician wasn't assigned to any shift. (10)

$$Y_{ijk} = \begin{cases} 1, if\ technician\ i\ will\ have\ a\ free\ day\ on\ day\ k \\ 0, oherwise \end{cases}$$

Penalty variables:

A day-off penalization is given if a technician worked on a day that requested as a day off.

$$\in_{ik}=\begin{cases}1, if\ technician\ i\ requested\ a\ day-off\ at\ day\ k\ and\ it\ wasn't given \\ 0, oherwise\end{cases} \quad (11)$$

A requested shift penalization is given if a technician requested a shift, and he was not assigned to it (12)

$$X_{ijk}=\begin{cases}1, if\ technician\ i\ will\ work\ on\ shift\ j\ on\ day\ k\ and\ it\ wasn't given \\ 0, oherwise\end{cases}$$

2.2.3. Model Development

By integrating these constraints into the mathematical model, the study optimizes the scheduling problem and produces the best possible solution. The following are the constraints:
C1: *T*echnician works one shift at most per day.
C2: Only Senior technicians can be assigned to the dawn shift.
C3: Only Orderly technicians can be assigned to the morning shift.
C4: At most of 22 working shifts for technicians.
C5: Technician should have assigned a one Sunday shift in one cycle.
C6: Maximum of seven assigned technicians in a shift from except Sunday.
C7:no two consecutive evening shifts.
C8: determines that a day is either a working day or free day.
C9: determines that a penalization cannot be penalized and given as a free simultaneously.
C10: determines that a requested free day is either given as a free day or is penalized.

Based on the presented mathematical functions form the given hard and soft constraints, the mathematical representation of the problem is presented in Equation 13.

$$Minimize\left(\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} S_{ijk}\left(1-X_{ijk}\right)+\sum_{j\in J}\sum_{k\in K} DOX_{iik}+\sum_{j\in J}\sum_{k\in K} ND_{jk}S_{dt}^{min}+\sum_{j\in J}\sum_{k\in K} N_{jk}\,D_{dt}^{min}\right) \quad (13)$$

2.2.4. Model Validation

The model underwent rigorous testing to ensure its validity and accuracy for optimization. It underwent comprehensive validation under various conditions, assessing its performance using computable metrics. Additionally, statistical and sensitivity analyses were conducted to assess the model's stability and responsiveness to changes in input parameters**.**

2.3. Optimization Stage

The optimization process consists of three stages: rapidly optimizing initial values through random selection. Then, an optimization process refines the solutions further. Finally, the algorithm aims for convergence to find a meaningful and optimal solution while considering constraints and objectives.

2.3.1 Select Optimization Method

The primary objective of the CSP is to generate a feasible solution that does not violate any hard constraints while concurrently minimizing the total penalty incurred from any breached soft constraints. This is represented in Eq.14.

$$minimize\ Penaltyvalue(S)=\begin{pmatrix}\sum_i w_i\, k_i\,(S) \\ subjectto\ s\in\ S, Penaltyvalue(S)=0\end{pmatrix} \quad (14)$$

The Hamming distance HD(S_x, S_y) represents the count of variables in S_x and S_y that hold different values. The calculations for this are presented in Eq.15.

$$HD\left(S_x(i), S_y(i)\right)=\sum_{i=1}^{n} CD(S_x(i), S_y(i)) \quad (15)$$

The movement of fireflies in the search space is the progression of a less bright firefly towards a brighter one in a search space problem, where fireflies share more values with the brighter firefly. The β-induced movement of solution S_x towards solution Sy in a discrete problem space is governed by Eq.16.

$$\beta_{movement_{i(i=1,2,3,\dots,n)}}\left(S_x(i), S_y(i)\right) = S_x(i) \leftarrow S_y(i)), \quad \text{if } S_x(i) \neq S_y(i) \text{ and } rnd \leq \beta \qquad (16)$$

2.3.2 Optimization/Sensitivity Analysis

The experiments were conducted to test the impact of manipulating variables on the outcome of the optimization problem. Both computational and experimental experiments were employed to identify the best solution. Statistical analysis, including mean, standard deviation, maximum, and minimum functions, was used to evaluate algorithm metrics such as objective function values, running time, and the number of firefly movements. The success and performance rate metrics were used to assess the optimization algorithm's performance.

2.3.3 Report Result

The important metrics, such as computation time, iterations, and convergence level, must be presented clearly and concisely to aid in decision-making. The interpretation of the findings should be comprehensive and aligned with the defined goals and performance metrics to ensure the results are meaningful and relevant to the investigated problem.

2.3.4 Report Result

The optimal solution is presented and assessed for feasibility and practicality, which involves analyzing the effects of input parameters and decision variables on the outcome and evaluating the generated solutions. The methods section thoroughly justifies the results, detailing the mathematical model, techniques, tools, and procedures employed in the study to ensure the credibility and reliability of the findings.

3. Results and Discussion

3.1. Modified Firefly Algorithm

The Modified Firefly Algorithm (MFA) is an advanced version of the standard Firefly Algorithm, with additional processes that enhance its optimization capabilities. These new processes, such as objective function filtering and adjusting positions, are vital in improving the algorithm's efficiency. The objective function filtering helps focus the search on promising areas, while adjusting positions to fine-tune the fireflies' locations, ensuring a balance between exploration and exploitation to achieve optimal solutions. These innovative steps make MFA more effective in navigating the solution space and optimizing results effectively.

```
Begin
Step 1   Problem Constants -->α,γ,iteration, nmaxIter
Step 2   Set maximum number of generations
Step 3   Best Initital solution
Step 4   Loop i = 1 to nPopulation
Step 5     Generate iniital population of firefly randomly --> m randon solution
Step 6   End loop
Step 7   While (iteration <nmaxIter)
Step 8   Define light absorption coeffiicient -
Step 9   Loop i= 1 to nPopulation
Step 10   Loop j=1 to nPopulation
Step 11     Compare (Fitness(si) < Fitness(sj))
Step 12     Save the position of the ScndFfly in AuxFfly
Step 13     Set the position of AuxFfly to the new position
Step 14    Evaluate the objective Fucntion at the new positionof the ScndFFly
           and save the result in ScndFunval
Step 15  Compare the objective function values of ScndFfly and AuxFfly
         If AuxFfly > ScnfFFly, update the position of ScndFfly to the new
         position of AuxFfly -
Step 17  Increment movements by 1
Step 18  Compare cost <= cost of the current position
Step 19  True: Assign the neighbor position to current position -
Step 20  break
Step 21     Determine the distance r_ij
Step 22     Calculate attractiveness β -
Step 23     β Movements (move toward s_i to s_j) -
Step 24    Evaluate new solution and update light intensity-
Step 25   End loop j
Step 26  End loop j
Step 27  Rank the fireflies and find the current global best solution
Step 28  End while
```

Fig. 3. Pseudocode for the Modified Firefly Algorithm

3.2. Complexity Analysis

The study emphasizes the importance of time and space complexity in evaluating algorithm performance. Time complexity refers to the number of steps an algorithm takes to solve a problem based on the input size, affecting its execution speed. On the other hand, space complexity measures the total memory consumption during operation. Table 1 compares the time complexity for four algorithms, indicating their computational efficiency and suitability for solving optimization problems.

TABLE 1.
TIME COMPLEXITY OF FA VARIANTS

Algorithm	Time Complexity
FA	$O(n^\wedge 2 * m * iter * (f(n) + h(n,d)))$
FA-TS	$O(n * (n * m * iter\_FA * (f(n) + h(n,d)) + iter\_TS * c(n)))$
FA-VND	$O(n^\wedge 2 * d * f(n) * iter\_FA + n * h(n) * iter\_VND)$
MFA	$O\_modified\_FA(n^\wedge 2 * m * iter\_FA * H(n,d))$

3.3. Performance of the Optimization Process

The optimization process of the discussed algorithms utilizes various modifications and techniques to enhance their performance in finding optimal solutions for a given problem. The process begins by initializing the firefly population, where each firefly represents a potential solution characterized by its position in the search space and brightness, which denotes its quality.

The parameter description for the optimization process is as follows: There are eight technicians (*nTechnicians)* and four shifts (*nShifts*) in a scheduling cycle of 30 days (*NDays*). Technicians work for 22 days within the 30-day cycle (*TechWorkingDays*). The γ values range from 0.025 to 1 in increments of 0.025. The α value is set at 0.5. The optimization process is simulated 30 times (*NSimulation*) with a population size of 50 (*nPopulation*) and 30 iterations (*nIterations*). After approximately 6 hours of evaluating the objective γ function value, the results indicated that the ideal γ value for the average objective value was 0.675. This highlights the importance of meticulously selecting the γ parameter to optimize the algorithm's performance and achieve a suitable trade-off between exploration and exploitation. By striking this balance, the algorithm can efficiently explore the solution space and effectively converge toward the optimal solution.

3.3.1 Solution Quality

The statistical analysis of the computational experiment's best objective function values is presented in Table 2. FA and FA-VNS show similar mean values, while FA-TS has a slightly higher mean value, suggesting potential for further optimization. MFA exhibits the lowest standard deviation, indicating consistent and reliable performance in solving the optimization problem with tightly clustered output values around the mean.

The experiment results presented in Table 2 indicate that FA and FA-VNS have comparable and stable running times, while FA-TS shows a running time nearly twice as long as MFA. This suggests that FA-TS may struggle with initial population variations of fireflies, leading to a slower convergence toward an optimal solution. On the other hand, MFA demonstrates a faster convergence and shorter running time, making it more efficient in effectively addressing optimization problems.

The study compared the mean firefly movement of FA, FA-VND, and MFA algorithms. FA and FA-VND showed similar movements, implying comparable exploration and exploitation behavior. However, MFA exhibited significantly lower mean movement, indicating a more focused search strategy. The low standard deviation in MFA suggests consistent and tightly clustered firefly movement, leading to faster convergence and improved optimization performance. These findings highlight the modified algorithm's (MFA) effectiveness in guiding firefly movement toward optimal solutions. MFA's focused search strategy and consistent movement indicate its potential for efficient search space exploration and convergence to high-quality solutions as shown in Table 2.

TABLE 2.
ALGORITHM METRIC FOR OBJCETIVE FUCNTION USING RANDOM SIMULATION

Algorithm	Objective Function Value		Computation Time		Movement of Fireflies	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
FA	17.767	4.842	60.795	7.768	300,700.000	41,571.000
FA-TS	18.733	9.940	813.690	228.870	120,970.000	88,887.000
FA-VND	17.867	4.863	71.445	14.128	305,680.000	62,294.000
MFA	8.367	2.287	485.610	38.960	2,991.200	304.770

IEEE-SEM, Volume 11, Issue 9, September-2023
ISSN 2320-9151

3.4. Comparison of Parameter Settings

The study employed meta-optimization tuning to select optimal parameter values before optimization. The parameters that produced the best results were identified by running the algorithms with various parameter settings and evaluating their performance. These parameters, including general population size and specific ones like α, beta, and γ in the Firefly Algorithm, aimed to maximize performance and enable the algorithms to solve the optimization problem effectively.

The study compares four Firefly Algorithm (FA) variants presented in Fig.4 with γ=0.8 and different α values to analyse their balance between exploration and exploitation for solution values. Lower γ values in FA indicate a focus on exploration, while higher values achieve a better balance and improved convergence. FA-TS and MFA also exhibit exploration at lower parameter values, generating diverse solutions, but higher values prioritize exploitation, leading to potential suboptimal solutions. Overall, metaheuristic algorithms are most effective at lower parameter values, ensuring a broad search of the solution space and avoiding premature convergence to suboptimal solutions, which is particularly beneficial for complex and multimodal optimization problems.

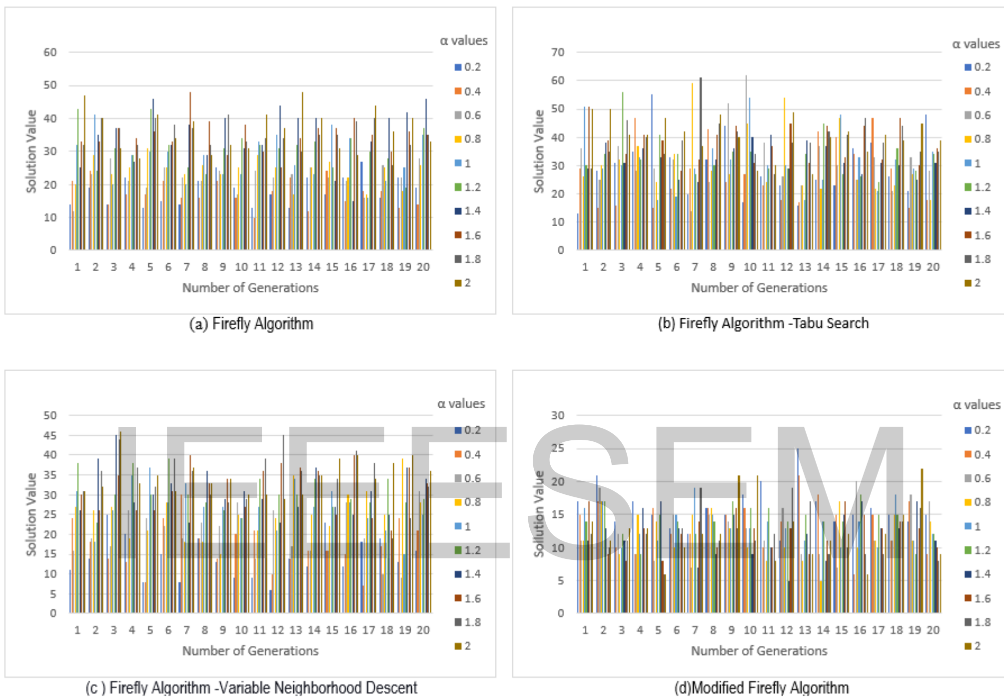


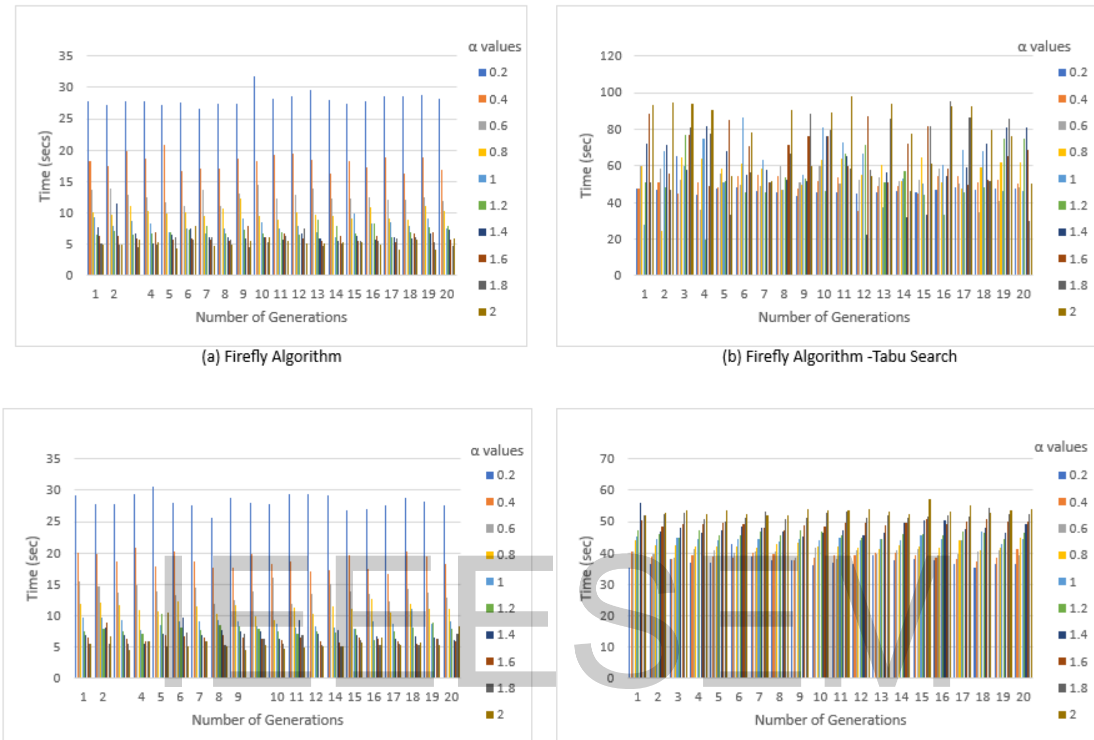
Fig. 4. Comparison of Best Solution Value for γ=0.8

Table 3 compares four algorithms' performance under different γ values. The results show that FA is moderately sensitive to changes in γ, with minor fluctuations in mean performance. FA-TS also demonstrates moderate sensitivity but consistently outperforms FA across all tested γ values. FA-VND exhibits less sensitivity to γ changes than FA and FA-TS, providing more consistent results. On the other hand, MFA is less affected by γ changes, with consistently lower mean performance, indicating a focused search process that may lead to higher-quality solutions. Table 7 summarizes the computational time for four algorithms. FA shows moderate computational efficiency compared to the others. FA-TS has the highest computation time, making it the least efficient. FA-VND follows a similar trend to FA, with computation time decreasing as the α value increases, indicating better efficiency with larger α values. However, MFA's computational time increases with larger α values, implying reduced efficiency. This divergent trend suggests that MFA's computational efficiency is more sensitive to changes in α values than FA and FA-VND.

TABLE 3.
COMPARATIVE SUMMARY FOR BEST SOLUTION VALUE AND COMPUTATION TIME

Algorithm	Mean – Best Solution Value			Mean – Computation Time		
	γ = 0.8	γ = 1.5	γ = 2.0	γ = 0.8	γ = 1.5	γ = 2.0
FA	28.45	29.35	28.73	10.91	10.78	11.19
FA-TS	33.52	34.54	33.58	62.80	59.03	56.46
FA-VND	26.56	27.11	27.16	11.27	11.41	12.11
MFA	13.07	13.41	13.34	47.65	45.23	47.84

Fig. 5 compares computational running times for four algorithms at different α values (with a fixed γ value of 0.8). FA and FA-TS exhibit faster search processes as α increases, indicating a focus on exploitation. Some exceptions imply complex problem instances or local optima traps. Surprisingly, shorter running times at α=1 may represent an optimal balance between exploration and exploitation. FA-VND's efficiency improves with higher α values, but variability exists due to problem complexities. MFA's running times do not follow a clear trend, and longer times at γ =1.5 suggest potential convergence challenges. These findings underscore the significance of α settings in achieving efficient optimization and the need to strike the right balance between exploration and exploitation for optimal results.



(a) Firefly Algorithm (b) Firefly Algorithm -Tabu Search

(c) Firefly Algorithm -Variable Neighborhood Descent (d)Modified Firefly Algorithm

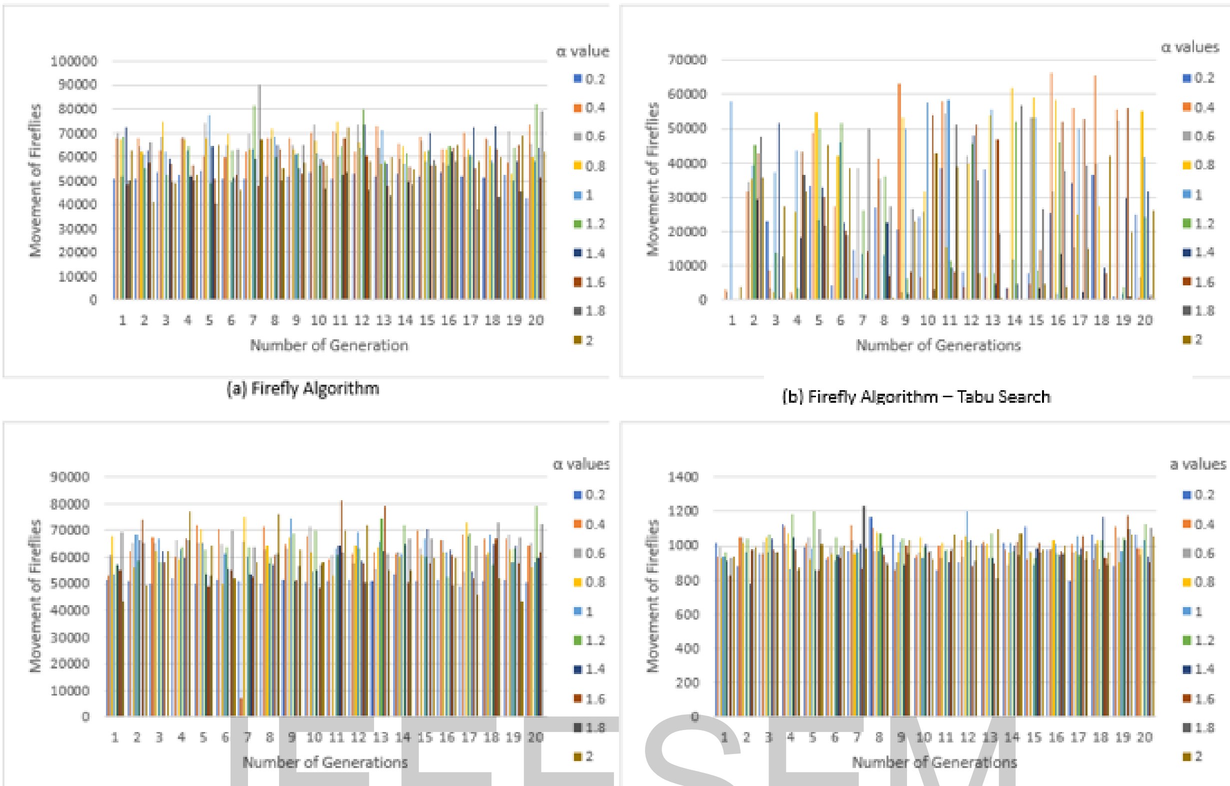
Fig.5. Comparison of Computation Time for γ=0.8

Table 4 compares firefly movements in four algorithms across different γ values. FA's movements remain relatively steady, indicating consistent exploration and convergence irrespective of γ changes. On the contrary, FA-TS's movements fluctuate, implying its performance might be γ-dependent. FA-VND shows a slight upward trend in movements as γ increases, suggesting γ value influences its exploration and convergence. Finally, MFA maintains consistent movements across all γ values, indicating its stable exploration and convergence are less affected by γ changes.

TABLE 4.
COMPARATIVE SUMMARY OF FIREFLIES MOVEMENT

Algorithm	Mean		
	γ = 0.8	γ = 1.5	γ = 0.8
FA	60,858	58,740	60,469
FA-TS	24,114	21,861	24,354
FA-VND	61,491	61,877	62,626
MFA	985	996	986

Fig. 6 compares firefly movement in four algorithms at γ value 2.0. FA shows many movements, some efficiently exploring, while others converge quickly or get trapped in suboptimal solutions. FA-TS has instances with no movements, suggesting possible firefly entrapment, while others explore efficiently or converge rapidly. FA-VND exhibits varied movements, indicating effective exploration for some and quick convergence or suboptimal entrapment for others. MFA displays a narrow range of movements, some exploring effectively, while others quickly converge or get trapped in suboptimal solutions.



(a) Firefly Algorithm

(b) Firefly Algorithm – Tabu Search

(c) Firefly Algorithm -Variable Neighborhood Descent

(d) Modified Firefly Algorithm

Fig.6. Comparison of Fireflies Movement for γ=2.0

4. CONCLUSIONS AND RECOMMENDATIONS

The study introduced a modified Firefly Algorithm (FA) for scheduling problems, incorporating Constraint Satisfaction Problems, Hamming Distance, objective function filtering, and positional adjustments. The Modified Firefly Algorithm (MFA) demonstrated improved convergence rates and solution quality compared to standard FA and FA-VND. Evaluating time and space complexity showed MFA's enhanced performance while maintaining acceptable complexity levels. Readability and generality were considered, ensuring clear comprehension and applicability to various optimization contexts. Testing MFA in a staff scheduling scenario validated its effectiveness in generating feasible and optimized schedules, highlighting its potential for real-world scheduling problems. Although MFA's computation time was longer due to the additional processes, it improved solution quality, proving a valuable trade-off. The study highlights the importance of γ and α parameter values in balancing exploration-exploitation trade-offs and emphasizes the adaptability of MFA for different problem scenarios. Overall, the study demonstrates the effectiveness of integrating meta-optimization and additional processes in enhancing the performance of metaheuristic algorithms, offering refined solutions to complex optimization problems. Moreover, using meta-optimization to fine-tune parameters enhances the MFA's efficiency and success rates. The study validates the practical applicability of the MFA by successfully generating feasible and optimal schedules in a real-world scenario, showcasing its potential for solving complex optimization problems. In conclusion, this study highlights the effectiveness of metaheuristic algorithms, particularly the MFA, and emphasizes the importance of suitable enhancements and adjustments to maximize performance.

To further develop the MFA, future research should focus on strategies to minimize computation time and explore varying 'n' values to assess its adaptability in different contexts. Comparing the MFA with other advanced algorithms could provide deeper insights into its relative performance. With continuous refinement, the MFA has the potential to become a robust and versatile tool in the field of optimization.

ACKNOWLEDGEMENT

The authors would like to thank the RLMH for the completion of the study.

REFERENCES

[1] H. H. Ali, H. Lamsali, and S. N. Othman, "Operating Rooms Scheduling for Elective Surgeries in a Hospital Affected by War-Related Incidents," *Journal of Medical System*s, vol. 43, no. 5, Apr. 2019, doi: https://doi.org/10.1007/s10916-019-1263-z.

[2] A.Y.A. Alfaki and A. YousifJob scheduling approaches based on firefly algorithm for computational grid 2013.(Doctoral dissertation, Universiti Teknologi Malaysia).

[3] A. Wasan and R. Dahiya, "Firefly algorithm for feature selection in intrusion detection system," *Soft Computing*, vol. 24, no. 7, pp. 4919–4932, 2020, doi: https://doi.org/10.1007/s00500-019-04463-x.

[4] J. Odili, A. Universiti, M. Pahang, R. Ambar, M. Helmy, and A. Wahab, "A Critical Review of Major Nature-Inspired Optimization Algorithms," *Technology, Engineering & Mathematics (EPSTEM*), vol. 2, pp. 376–394, 2018, Available: http://www.epstem.net/en/download/article-file/528356

[5] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm Intelligence Algorithms for Feature Selection: A Review," *Applied Sciences*, vol. 8, no. 9, p. 1521, Sep. 2018, doi: https://doi.org/10.3390/app8091521.

[6] W. A. Khan, N. N. Hamadneh, S. L. Tilahun, and J. M. T. Ngnotchouye, "A Review and Comparative Study of Firefly Algorithm and its Modified Versions," Optimization Algorithms - Methods and Applications, Sep. 2016, doi: https://doi.org/10.5772/62472.

[7] G. P. Georgiadis, A. P. Elekidis, and M. C. Georgiadis, "Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications," Processes, vol. 7, no. 7, p. 438, Jul. 2019, doi: https://doi.org/10.3390/pr7070438.

[8] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, p. 78, 2010, doi: https://doi.org/10.1504/ijbic.2010.032124

[9] N. Figueiredo, M. Macedo, H. Siqueira, C. J. Santana, A. Gokhale, and Carmelo, "Swarm intelligence for clustering — A systematic review with new perspectives on data mining," Engineering Applications of Artificial Intelligence, vol. 82, pp. 313–329, Jun. 2019, doi: https://doi.org/10.1016/j.engappai.2019.04.007.

[10] Z. Zhang, M. Li, and J. Hu, "A hybrid firefly algorithm for solving the maximum weight clique problem," *Soft Computing*, vol. 23, no. 12, pp. 4623–4633, 2019, doi: https://doi.org/10.1007/s00500-018-3461-7.

[11] Y. Liu, X. Zhang, and M. Li, "A hybrid firefly algorithm with orthogonal design for feature selection," Soft Computing, vol. 22, no. 19, pp. 6387–6396, 2018, doi: https://doi.org/10.1007/s00500-018-3149-y.

[12] M. Li, J. Hu, and H. Chen, "A hybrid firefly algorithm for the job shop scheduling problem," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 7, pp. 2475–2485, 2019, doi: https://doi.org/10.1007/s12652-019-01300-4.

[13] A. Yelghi and C. Köse, "A modified firefly algorithm for global minimum optimization," Applied Soft Computing, vol. 62, pp. 29–44, Jan. 2018, doi: https://doi.org/10.1016/j.asoc.2017.10.032.

[14] B. K. Patle, A. Pandey, A. Jagadeesh, and D. R. Parhi, "Path planning in uncertain environment by using firefly algorithm," Defence Technology, vol. 14, no. 6, pp. 691–701, Dec. 2018, doi: https://doi.org/10.1016/j.dt.2018.06.004.

[15] M. Sababha, M. Zohdy, and M. Kafafy, "The Enhanced Firefly Algorithm Based on Modified Exploitation and Exploration Mechanism," Electronics, vol. 7, no. 8, p. 132, Jul. 2018, doi: https://doi.org/10.3390/electronics7080132.

[16] G. E. Apostolopoulos, and D. S. Vlachos, "Particle swarm optimization in wireless sensor networks:A brief survey. ," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 122–130, 2010, doi: https://doi.org/10.1109/MCOM.2010.5393813.

[17] A. Khadwilard, S.Chansombat, T. Thepphakorn, W. Chainate, and P. Pongcharoen, "Application of Firefly Algorithm and Its Parameter Setting for Job Shop Scheduling," vol. 8, no. 1, pp. 49–58, Nov. 2012

[18] K. C. Udaiyakumar and M. Chandrasekaran, "Application of Firefly Algorithm in Job Shop Scheduling Problem for Minimization of Makespan," Procedia Engineering, vol. 97, pp. 1798–1807, 2014, doi: https://doi.org/10.1016/j.proeng.2014.12.333.

[19] P. Świtalski and A. Bolesta, "Firefly algorithm applied to the job-shop scheduling problem," Studia Informatica, no. 25, Dec. 2021, doi: https://doi.org/10.34739/si.2021.25.05.

[20] I. Ahmed Saleh, O. Ibrahim Alsaif, S. Abduttalib Muhamed, and E. Ibrahim Essa, "Task Scheduling for cloud computing Based on Firefly Algorithm," *Journal of Physics:* Conference Series, vol. 1294, p. 042004, Sep. 2019, doi: https://doi.org/10.1088/1742-6596/1294/4/042004.

[21] G. Mai, "Application of firefly algorithm for job shop scheduling," Jan. 2016, doi: https://doi.org/10.2991/icmmct-16.2016.330.

[22] F. Ebadifard, S. Doostali, and S-M Babamir, "A Firefly-based Task Scheduling Algorithm for the Cloud Computing Environment: Formal Verification and Simulation Analyses," Dec. 2018, doi: https://doi.org/10.1109/istel.2018.8661088.

[23] D.-P. Liu, T.-Y. Lin, and H.-H. Huang, "Improving the Computational Efficiency for Optimization of Offshore Wind Turbine Jacket Substructure by Hybrid Algorithms," Journal of Marine Science and Engineering, vol. 8, no. 8, p. 548, Jul. 2020, doi: https://doi.org/10.3390/jmse8080548.

[24] S. Karmakar, S. Chakraborty, T. Chatterjee, Arindam Baidya, and Sriyankar Acharyya, "Meta-heuristics for solving nurse scheduling problem: A comparative study," Sep. 2016, doi: https://doi.org/10.1109/icaccaf.2016.7748951

[25] G.-D. Zhou, T.-H. Yi, H. Zhang, and H.-N. Li, "A Comparative Study of Genetic and Firefly Algorithms for Sensor Placement in Structural Health Monitoring," Shock and Vibration, vol. 2015, pp. 1–10, 2015, doi: https://doi.org/10.1155/2015/518692.

[26] A. P. Napalit and M. A. Ballera, "Optimizing a schedule using firefly algorithm with Tabu search algorithm," Jan. 2023, doi: https://doi.org/10.1063/5.0121582.

[27] R. Sari and R-I Widianti, "Optimizing Employee Scheduling System with Firefly Algorithm (Case Study: MJ Store)," Sep. 2020, doi: https://doi.org/10.1109/ic2ie50715.2020.9274615.

[28] A. P. Napalit and M. A. Ballera, "Application of Firefly Algorithm in Scheduling," Nov. 2021, doi: https://doi.org/10.1109/icoco53166.2021.9673581.

[29] D. S and A. Ravikumar, "A Study from the Perspective of Nature-Inspired Metaheuristic Optimization Algorithms," International Journal of Computer Applications, vol. 113, no. 9, pp. 53–56, Mar. 2015, doi: https://doi.org/10.5120/19858-1810.

[30] V. Kulkarni, V. Desai, V and R. Kulkarni, Comparison of Firefly, Cultural and the Artificial Bee Colony Algorithms for Optimization. (2018) 16. 4.

[31] W. Tong, "A Hybrid Algorithm Framework with Learning and Complementary Fusion Features for Whale Optimization Algorithm," *Scientific Programming*, vol. 2020, pp. 1–25, Feb. 2020, doi: https://doi.org/10.1155/2020/5684939.

[32] A.M. Altabeeb, A. M., Mohsen and A. Ghallab, A. An improved hybrid Firefly algorithm for capacitated vehicle routing problem. Applied Soft Computing, 2019. 84, 105728. https://doi.org/10.1016/j.asoc.2019.105728

[33] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," Applied Soft Computing, vol. 11, no. 1, pp. 652–657, Jan. 2011, doi: https://doi.org/10.1016/j.asoc.2009.12.025.

[34] S. M. Farahani, A. A. Abshouri, B. Nasiri, and M. R. Meybodi, "Some Hybrid models to Improve Firefly Algorithm Performance," *International Journal of Artificial Intelligence*, vol. 8, no. S12, pp. 97–117, Jan. 2012, Accessed: Jul. 29, 2023. [Online]. Available: http://www.ceser.in/ceserp/index.php/ijai/article/view/2358/0

[35] M. A. Tawhid and A. F. Ali, “Direct Search Firefly Algorithm for Solving Global Optimization Problems,” *Applied Mathematics & Information Sciences*, vol. 10, no. 3, pp. 841–860, May 2016, doi: https://doi.org/10.18576/amis/100304.

[36] R. Sahoo, D. Ojha, & D. Mohapatra and M. Patra. Automatic generation and optimization of course timetable using a hybrid appraoch. *Journal of Theoretical and Applied Information Technology*. (2017). 95. 68-77.

[37] C. Wang and X. Chu, “An Improved Firefly Algorithm With Specific Probability and Its Engineering Application,” IEEE Access, vol. 7, pp. 57424–57439, 2019, doi: https://doi.org/10.1109/access.2019.2914534.

[38] W.H. Crown, N.C. Buyukkaramikli, W.S. Thayer, and M. TuncA novel hybrid metaheuristic algorithm for the multiple-choice multidimensional knapsack problem. *Journal of Heuristics*, 2018, 24(6), 941-963. https://doi.org/10.1007/s10732-018-9385-2