Al-Nahrain Journal of Science
ANJS, Vol. 26(4), December, 2023, pp. 88-94

Modified Firefly Algorithm using Iterated Descent Method to Solve Machine Scheduling Problems

Hafed M. Motair

Al Diwaniyah Secondary School for Distinguished Students, Ministry of Education, Iraq

Article's Information

Received: 23.09.2023
Accepted: 30.11.2023
Published: 10.12.2023

Keywords:
Single machine scheduling
Metaheuristics
Iterated descent method
Firefly algorithm
Genetic algorithm

Abstract

One of the most efficient metaheuristic algorithms that is used to solve hard optimization problems is the firefly algorithm (FFA). In this paper we use this algorithm to solve a single machine scheduling problem, we aim to minimize the sum of the two cost functions: the maximum tardiness and the maximum earliness. This problem (P) is NP-hard so we solve this problem using FFA as a metaheuristic algorithm. To explore the search space and get a good solution to a problem (Q), we hybridize FFA by Iterated Descent Method (IDM) in three ways and the results are FFA1, FFA2, and FFA3. In the computational test, we evaluate these algorithms (FFA, FFA1, FFA2, FFA3) compared with the genetic algorithm (GA) through a simulation process with job sizes from 10 jobs to 100 jobs. The results indicate that these modifications improve the performance of the original FFA and one of them (FFA3) gives better performance than others.

DOI: 10.22401/ANJS.26.4.13

*Corresponding author: hafedmotair@gmail.com

1. Introduction

One of the most important methods that used to solve several real-word problems are global optimization methods. Most of these methods that implementing to solve a hard optimization problem are metaheuristics such as GA [1], Particle Swarm Optimization (PSO) [2], Evolutionary Programming Algorithms (EPAs) [3]. However, metaheuristics often need to combined with some kinds of local search in order to get-away from local minima. In this paper we consider a single machine scheduling problem and aim to minimize the sum of two cost functions: maximum tardiness and maximum earliness, its shown that this problem is NP-hard problem, so we try to solve this problem using one of the most efficient metaheuristic algorithms that used to solve hard optimization problems which is firefly algorithm (FFA) [4], [5], but instead of using FFA alone, we combined the algorithm with IDM by three ways and the resulting are: FFA1, FFA2 and FFA3. Several approaches considering FFA and some of these approaches include hybridize FFA has been proposed in literature. [6] propose a discrete FFA metaheuristic, the objective is to minimize the makspan for the permutation flowshop scheduling problem (PFSP), the algorithm compared with ant colony optimization algorithm and the results showed the efficiency of the proposed method. [7] consider job shop scheduling problem (JSSP) and the objectives are to present the application of FFA for solving JSSP, explore the parameter setting of the proposed FFA and examine different parameter setting and compare the results. In [8] the authors applied and hybridized the FFA with local search algorithm to solve combinatorial optimization problems. The proposed algorithm compared with some evolutionary algorithms and the results showed the efficiency of the proposed algorithm. In [9] the authors hybridized FFA with simulated annealing (SA) algorithm to solve FSP with learning effects, the problem formulated by mixed integer linear programming. The results showed the effectiveness of the proposed algorithm. In [10] the authors extended the FFA to bi-objective hybrid FSPs and the objective is to minimize makspan and mean flow time cost functions and the results showed that the proposed algorithm outperforms many metaheuristics. In [11] the authors proposed a hybrid crow search algorithm to minimize a makspan in PFSP, they use the SPV rules to convert real values to job sequence. They use NEH algorithm to generate initial population and use variable neighborhood search (VNS) with SA algorithm to developed the proposed algorithm. The result of comparing the proposed algorithm showed the efficiency of the proposed algorithm. In [12] the authors solving a TSP problem using FFA and k-means clustering, the algorithm divides the nodes into sub-problems by using the k-means clustering and then use the FFA to find the best path in each cluster. The results showed that the proposed algorithm give better performance than the other algorithms. [13] hybridized FFA with GA to schedule the tasks and the objective is to minimize the execution time for all tasks. The results showed that the proposed algorithm outperform GA and FIFO algorithms. In [14] the authors combined the FFA with VNS for Data Clustering (FA-VNS), the results showed that the proposed algorithm performance better than other well-known clustering algorithms in literature. In [15] the authors propose a modified FFA to effectively observe the network by introducing a new health function for early detection of suspicious nodes, the results showed that the proposed algorithm reduces the number of suspicious nodes. In this paper, we proposed three modifications to firefly algorithm FFA and used IDM algorithm to improve the performance of the original algorithm.

The first improvement used the IDM to generate the first solution of the initial population and the second improvement is the use of IDM to generate all solutions of the initial population. The third improvement is the use of IDM algorithm to update some solutions rather than generating them randomly and this process done to the solutions that having same values of objective function, the results showed that thrse modifications improve the performance of the original algorithm. This paper is organized as follows: The problem definition is stated in Section 2, the Firefly algorithm is defined in Section 3. In Section 4 the proposed algorithms are described. The experimental results are outlined in Section 5, finally, the conclusions are discussed in Section 6.

2. Problem Definition

Let $\mathrm{N} = \{1,2,\dots,\mathrm{n}\}$ be the set of n jobs, a_{j} a processing time and t_j is a due date of a job j, for $\mathrm{j} = 1,2,\dots,\mathrm{n}$. Let $\pi = (\pi(1),\pi(2),\dots,\pi(\mathrm{n}))$ be a sequence of the jobs in N, where π(j) is the jth job to be processed by a machine. The completion time of job π(j) is given by $\mathrm{C_j} = \sum_{\mathrm{k}=1}^{j} \mathrm{a}_{\pi(\mathrm{k})}$, the tardiness of the job π(j) is givin by $\mathrm{T}_j = \max(\mathrm{C}_{\pi(\mathrm{i})} - \mathrm{t}_{\pi(\mathrm{i})}, 0)$, and the earliness of the job π(j) is givin by $\mathrm{E_j} = \max(\mathrm{t}_{\pi(\mathrm{i})} - \mathrm{C}_{\pi(\mathrm{i})}, 0)$. For a given schedule $\delta = (1,2,\dots,\mathrm{n})$, the mathematical form of the problem can be written as follows:

$$\left.\begin{array}{ll} \multicolumn{2}{l}{g(\pi) = \min(\mathrm{T_{max}} + \mathrm{E_{max}})} \\ \multicolumn{2}{c}{s.t} \\ C_j \geq a_j & j = 1,2,\dots,n \\ C_j = C_{(j-1)} + a_j & j = 2,3,\dots,n \\ T_j \geq C_i - t_j & j = 1,2,\dots,n \\ T_j \geq 0 & j = 1,2,\dots,n \\ E_j \geq t_j - C_j & j = 1,2,\dots,n \\ E_j \geq 0 & j = 1,2,\dots,n \end{array}\right\} \dots (1)$$

The objective is to find optimal permutation (sequence) that optimize problem (P).

3. Firefly Algorithm

Firefly algorithm (FFA) is one of the most powerful metaheuristics that inspired by flashing behavior of fireflies, it is a population search algorithm and is designed to be guided by three rules [5]: The first one is that the fireflies are unisex which means that each firefly can be attracted to any other one. The second rule is that the brightness affected by the value of the objective function. The third is the attraction of each firefly depend on its brightness and they are decreasing as their distance increase. There are two main parameters which control the behavior of FFA: light intensity (γ) and the attractiveness (β). The attractiveness is defined using the distance as follows: $\beta = \beta_0 e^{-\gamma r_{ij}^2}$, where β_0 initial attractiveness ($r_{ij} = 0$), r_{ij} is the distance between solution (i) and the neighborhood solution (j) ,

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{m=1}^{D} (x_{im} - x_{jm})^2} \quad \ldots (1)$$

Therefore, position of the solution was updated using this new attractiveness value as in the following equation (Attraction equation):

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha \varepsilon_i^t \quad \ldots (2)$$

where x_i^t is the ith solution in iteration t, $i = 1, \ldots, N$,and N is the population size. ε_i^t is Gaussian distribution vector of numbers at time t and α the randomization parameter can be reduced with the iteration process as follows $\alpha = \alpha_0 \theta^t$, $\theta \in (0,1)$ for some initial value α_0. We note that our problem (P) uses the solutions as integer sequences $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ such that $\pi(i)$ is integer value, so we need a method to convert the real values to integer values, this done by rounded these real values to the nearest integer values, this method is similar to that used in [16]. In the following steps we summarize the FA algorithm:

Standard FA:

1. (Initialization) Generate initial population randomly contains N solutions $X_i = (x_{im})$, $i = 1, \ldots, N,\ m = 1, \ldots, D$. Evaluate each firefly in the initial population using the objective function $f(X)$
2. (Attraction) Compare each solution X_i with other all solutions X_j in the population, where $i, j = 1, \ldots, N,\ i \neq j$. If $f(X_k) > f(X_h)$, then move X_h towards X_k and update the position using Attraction equation. The solutions are then evaluated using updated positions.
3. (Stopping criterion) Stop the algorithm if the stopping criterion is satisfied, otherwise go to step (2).

4. Proposed Algorithms

4.1. Solution presentation

FFA use continues number encoding, so we need a method to convert the real values to integer values, this done by rounded these real values to the nearest integer values (SPV), this method is similar to that used in [16] which convert the sequence of real values to integer. An example of this process in Table (1) where the first column is the sequence of 5 jobs, the second column is the sequence of real values to be converted to permutation sequence and the last column is the permutation sequence resulted by SPV procedure.

Table 1. Example of SPV procedure

Job sequence	Sequence of real values	Permutation sequence
1	0.5	4
2	-1.4	2
3	-0.12	3
4	-4.78	1
5	2.1	5

We note that we can use the SPV procedure to mapping the permutation sequence to real-valued sequence. An example of this process presenting in Table 2.

Table 2. Example of mapping job permutation to sequence of real values

Job sequence	Sequence of real values	Permutation sequence
3	-0.45	-0.45
5	1.42	1.42
1	2.67	-1.31
2	-3.31	2.67
4	-0.95	-0.95

4.2 Iterated Descent Method (IDM)

In IDM the initial solution (π) selected and the neighborhood solution $(\acute{\pi})$ generated then the algorithm evaluates the objective function values $f(\acute{\pi}), f(\pi)$ and calculates $\Delta = f(\acute{\pi}) - f(\pi)$. If $\Delta < 0$, then $\acute{\pi}$ is considered as the current solution. On the other hand, when $\Delta > 0$, then π is remained as the current solution. This process is repeated and the search continues with all neighborhoods of the current solution. The algorithm stops when the stopping criterion is satisfied.

IDM Algorithm

1. Choose a starting solution π
2. Calculate $F(\pi)$ (objective function value)
3. Repeat until a termination condition is satisfied:
   i. Generate randomly a solution $\acute{\pi}$ as a neighbor of π
   ii. Calculate $F(\acute{\pi})$
   iii. If $F(\acute{\pi})$is better than $F(\pi)$ then $\pi = \acute{\pi}$
4. Return solution π and a value $F(\pi)$.

4.3 Modifications of FFA

The first modification is simple such that one of the solutions of the initial population is generated using the IDM algorithm and the other solutions are generated randomly. The resulting modified algorithm denoted FFA1. The second modification is denoted FFA2 such that all solutions of the initial population are generated using the IDM algorithm. The third modification denoted FFA3 is shown in the original FAA that in the case where there is not any brighter one between any two solutions, the process goes to update the new solution to the random walk, we modified this part of the algorithm so that instead of going for a random walk we updated the solution using IDM algorithm. We hope this modification ensures better solutions. The performance of the proposed modification algorithms is presented in this section as follows:

5.1 Parameter setting

The parameter setting presented in table (3):

Table 3. Parameter setting of algorithms

N	GA	FFA	FFA1	FFA2	FFA3
10	172.7	172.7	172.7	172.7	172.7
20	298.6	298.6	298.6	298.6	298.6
30	356.2	356.2	356.2	356.2	356.2
40	442.1	443.7	448.8	443.0	442.1
50	520.0	530.5	525.8	522.8	519.5
60	628.3	646.8	633.0	629.2	628.3
70	656.2	769.1	708.2	690.4	656.6
80	709.7	931.2	817.6	793.7	709.5
90	712.0	1136.4	1086.3	1003.6	713.5
100	810.3	1549.44	1515.6	1244.1	808.4
Mean	530.6	683.5	656.3	615.4	530.5

For the IDM algorithm we use the insertion neighborhood for generating a new solution and the number of iterations is 2000, we use discrete uniform distribution to generate the processing times on interval [1,99] and the due dates of jobs are also generated using a uniform distribution on the interval $[(1 - TF - RDD/2)P, (1 - TF + RDD/2)P]$, such that RDD and TF are hardiness factors of the problem (P) taken from the sets $\{0.2,\ 0.6, 1.2\}$ and $\{0.2, 0.4, 0.8\}$ respectively, $P = \sum_{j=1}^{n} a_j$. We use LENOVO machine Intel (R) Core ™ (i7) CPU @ 2.50 GHz, and 8 GB of RAM. In this research, we focused on generating instances of sizes $n = 10, 20, \dots, 100$ and for each n we generate nine examples.

5.2 Comparison of the proposed modification algorithms.

In table (4) we present the average of nine examples of the objective function values of algorithms (GA, FFA1, FFA2, FFA3) for $n = 10, 20, \dots ,100$, and in table (5) we present the average times of these problems. We compare the performance of the original firefly algorithm (FFA) and its modifications (FFA1, FFA2, and FFA3) with the genetic algorithm (GA). The results in table (4) showed that all compared algorithms gave the same results for $n = 10{,}20$ and 30 which can be considered as moderate values. The rest of the results showed that the modified algorithm FFA3 and the genetic algorithm GA gave similar results, with a preference for the modified Algorithm FFA3. It is clear that the modifications made to the firefly algorithm (FFA), despite their simplicity, gave good results and greatly improved the performance of the algorithm. This is evident if we compare the results of the FFA with the modified algorithms FFA1, FFA2 and FFA3.

The results also showed that the effect of the IDM algorithm is clear in improving the performance of the three modifications FFA1, FFA2 and FFA3, and that using this algorithm to generate the initial population as in the FFA2 is better than generating only one solution in the initial population as in FFA1. And we can use the Figure (2) to illustrate these results. Table (5) shows the mean values of execution times of the considered algorithms where the obvious effect of using the local search algorithm (IDM) is shown.

Table 4. Comparison of the results (in mean values) of the considered algorithms.

	Number of Iterations	Size of Initial Population
GA	500	500
FFA	100	100
FFA1	100	100
FFA2	100	100
FFA3	100	100

Table 5. Comparison of the execution times of the considered algorithms (in mean values).

n	GA	FFA	FFA1	FFA2	FFA3
10	14.36	8.42	8.15	14.19	20.06
20	14.79	10.95	9.33	14.64	21.27
30	16.18	14.95	11.69	17.80	23.51
40	17.04	18.83	12.77	19.44	23.75
50	18.64	26.34	21.32	24.65	28.27
60	21.13	34.48	28.87	30.03	31.00
70	23.23	41.91	38.21	39.43	34.25
80	26.16	50.24	45.39	46.86	38.33
90	28.32	54.55	51.86	54.40	44.71
100	31.24	55.00	47.59	59.84	51.55
Mean	21.109	31.567	27.518	32.128	31.67

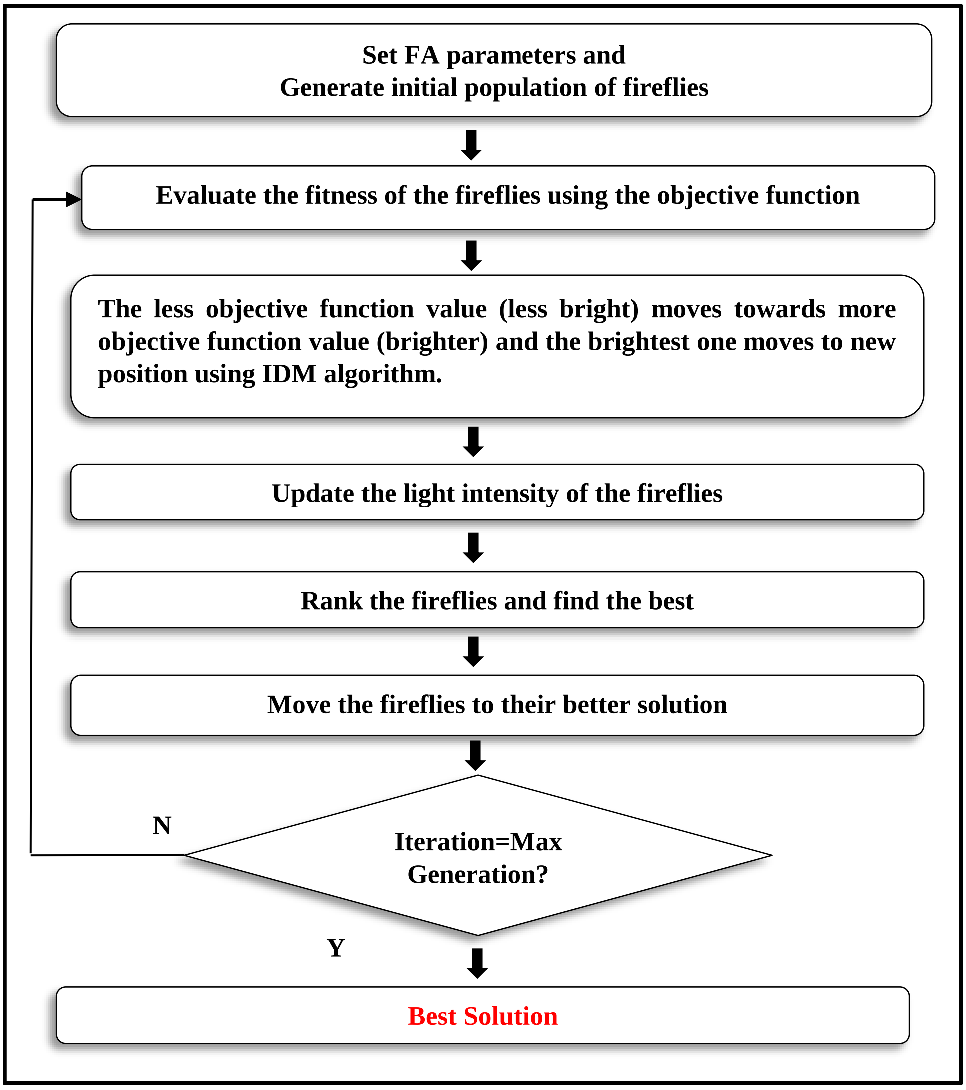

Figure 1. Flowchart of the proposed modification of FFA (FFA4)

Al-Nahrain Journal of Science
ANJS, Vol. 26(4), December, 2023, pp. XX-XX

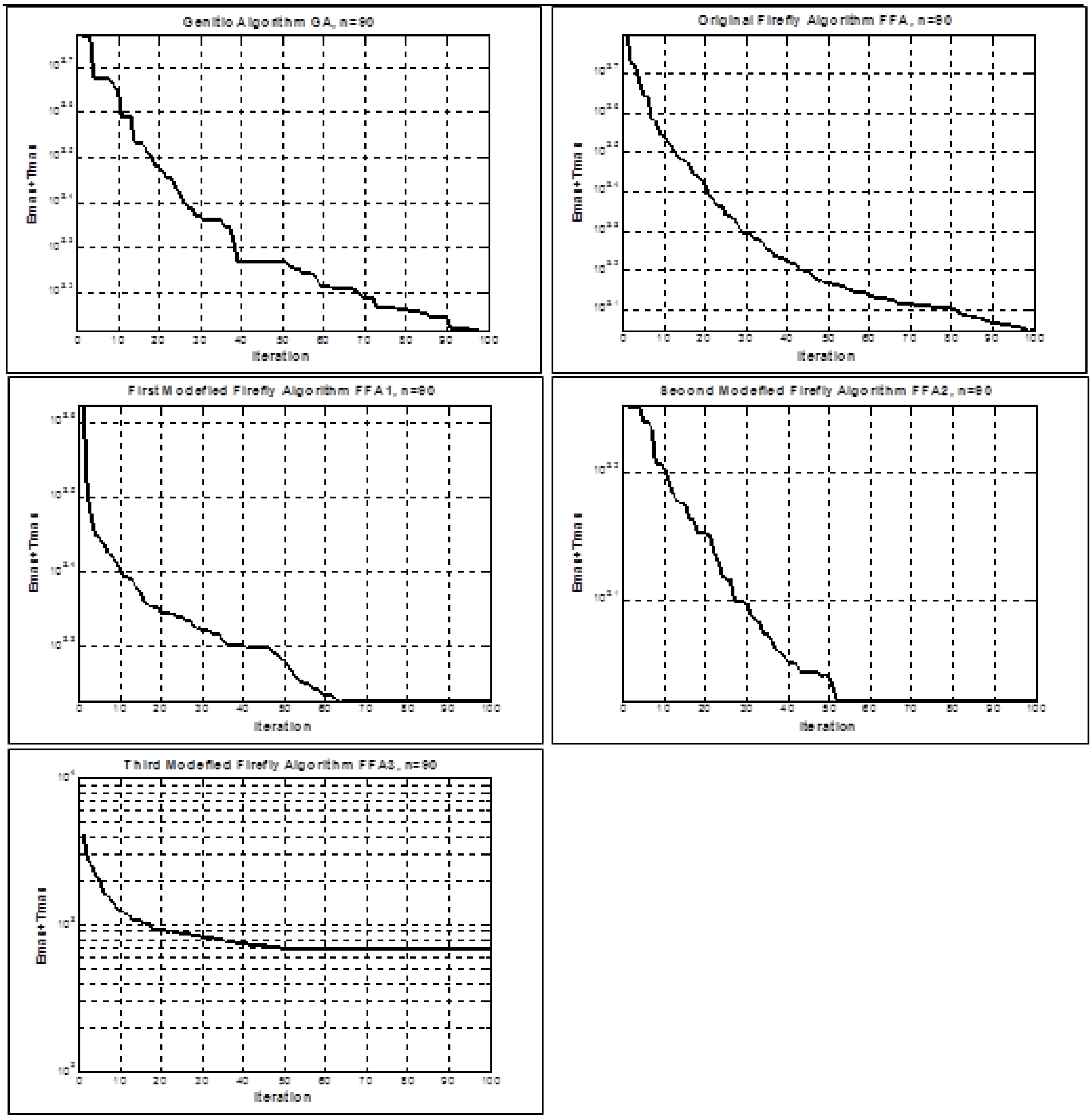

Figure 2. Comparison between GA, FFA, FFA1, FFA2, and FFA3, n=90.

5. Conclusions

In this paper, we proposed modifications to firefly algorithm FFA which are: FFA1, FFA2, and FFA3, we used the IDM algorithm to improve the performance of the original algorithm. The first improvement (FFA1) used the IDM to generate the first solution of the initial population which to some degree improved the performance of the original algorithm. The second improvement (FFA2) is the use of IDM to generate all solutions of the initial population which gave better results than those obtained by FFA1. The third improvement is the use of the IDM algorithm to update some solutions rather than going on a random walk to update these solutions and this process was done to the solutions that have the same values of the objective function, the results showed that this modification improves the performance of the original algorithm and gave better values (table (4)) than other modifications also this modification gave competitive results to those values obtained by the genetic algorithm.

References

[1] Grefenstette, J.J.; “Genetic algorithms and machine learning”, EM Proceedings of the sixth annual conference on Computational learning theory, 1993, p. 3–4.

[2] Poli, R.; Kennedy, J.; Blackwell, E.T.; “Particle swarm optimization”, Swarm Intelligence, 1(1): 33–57, 2007.

[3] Husbands, P.; Harvey, I.; Cliff, D.; Miller, E.G.; “Artificial evolution: A new path for artificial intelligence”. Brain Cogn., 34(1): 130–159, 1997.

[4] Yang, X.-S.; “Nature-inspired metaheuristic algorithms”. Luniver Press, 2010.

[5] Yang, X.-S.; “Firefly algorithms for multimodal optimization”, Em. International symposium on stochastic algorithms, Springer, 2009, p. 169–178.

[6] Sayadi, M.; Ramezanian, R.; Ghaffari-Nasab, E.N.; “A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems.”. Int. J. Indus. Eng. Comp., 1(1): 1–10, 2010.

[7] Khadwilard, A.; Chansombat, S.; Thepphakorn, T.; Chainate, W.; Pongcharoen, E.P.; “Application of firefly algorithm and its parameter setting for job shop scheduling.”. The J. Indus. Technol., 8(1): 49–58, 2012.

[8] Fister, I.; Yang, X.-S.; Fister, I.; Brest, E.J.; “Memetic firefly algorithm for combinatorial optimization”, arXiv preprint arXiv:1204.5165, 2012.

[9] Nouri, B.V.; Fattahi, P.; Ramezanian, E.R.; “Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities”. Int. J. Prod. Res., 51(12): 3501–3515, 2013.

[10] Marichelvam, M.K.; Prabaharan, T.; Yang, E.X.S.; “A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems”, IEEE Trans. Evol. Comp., 18(2): 301–305, 2013.

[11] Huang, K.-W.; Girsang, A.S.; Wu, Z.-X.; Chuang, E.Y.-W.; “A hybrid crow search algorithm for solving permutation flow shop scheduling problems”, Appl. Sci., 9(7): 1353, 2019.

[12] Jaradat, A.; Diabat, W.; “Solving traveling salesman problem using firefly algorithm and k-means clustering”, em 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), IEEE, 2019, p. 586–589.

[13] Rajagopalan, A.; Modale, D.R.; Senthilkumar, E.R.; “Optimal scheduling of tasks in cloud computing using hybrid firefly-genetic algorithm”, em Advances in decision sciences, image processing, security and computer vision, Springer, 2020, p. 678–687.

[14] Al-Behadili, H.N.K.; “Improved Firefly Algorithm with Variable Neighborhood Search for Data Clustering”, Baghdad Sci. J., 19(2), 2022.

[15] Shandilya, S.K.; Choi, B.J.; Kumar, A.; Upadhyay, E.S.; “Modified Firefly Optimization Algorithm-Based IDS for Nature-Inspired Cybersecurity”, Processes, 11(3): 715-720, 2023.

[16] Bacanin, N.; Brajevic, I.; Tuba, E.M.; “Firefly algorithm applied to integer programming problems”, Recent Adv. Math., 888: 999-1002, 2013.