

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage: www.joiv.org/index.php/joiv



An Improved Flower Pollination Algorithm for Global and Local Optimization

M. Iqbal Kamboh^{a,*}, Nazri Bin Mohd Nawi^a, Azizul Azhar Ramli^a, Fanni Sukma^b

^a Faculty of Computer Science & Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia ^b Department of Information Technology, Politeknik Negeri Padang, West Sumatera, Indonesia

Corresponding author: ^{*}iqbalmoeez@gmail.com

Abstract— Meta-heuristic algorithms have emerged as a powerful optimization tool for handling non-smooth complex optimization problems and also to address engineering and medical issues. However, the traditional methods face difficulty in tackling the multimodal non-linear optimization problems within the vast search space. In this paper, the Flower Pollination Algorithm has been improved using Dynamic switch probability to enhance the balance between exploitation and exploration for increasing its search ability, and the swap operator is used to diversify the population, which will increase the exploitation in getting the optimum solution. The performance of the improved algorithm has investigated on benchmark mathematical functions, and the results have been compared with the Standard Flower pollination Algorithm (SFPA), Genetic Algorithm, Bat Algorithm, Simulated annealing, Firefly Algorithm and Modified flower pollination algorithm. The ranking of the algorithms proves that our proposed algorithm IFPDSO has outperformed the above-discussed nature-inspired heuristic algorithms.

Keywords- Dynamic switch probability; meta-heuristic; benchmark function; multimodal; exploitation; exploration.

Manuscript received 10 Feb. 2021; revised 12 Apr. 2021; accepted 19 Oct. 2021. Date of publication 31 Dec. 2021. International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.

I. INTRODUCTION

Solving multifaceted optimization problems can be challenging when multiple and inconsistent design goals are considered. An emerging trend in using meta-heuristic algorithms to answer complex optimization problems, these algorithms have revealed great success in maintaining balance among inconsistent design goals. Most of the meta-heuristic algorithms have been established in recent decades. Many of these algorithms require certain parameters to show their best performance. For example, the Genetic Algorithm (GA) [1] requires considerable adjustment for population size, crossover rate and mutation. In the circumstance of Particle Swarm Optimization (PSO) [2], the same issue also appears, which depends on population size, weight of inertia and social parameters. Similarly, Harmony Search (HS) [3] requires adjustment of harmony memory deliberation rate, harmony size, and tuning of pitch. As for Ant Colony Optimization (ACO) [4], choosing the correct evaporation rate, pheromone effect and heuristic function are essential. Some other successful stochastic algorithms are Bee Colony Algorithm

[5], Artificial bee colony algorithm (ABC) [6], Cuckoo Search Algorithm (CS) [7], Bat Inspired Algorithm (BA)[8], Firefly algorithm (FA) [9], Simulated Annealing(SA)[10], Differential Evolution (DE)[11] and Flower Pollination Algorithm (FPA)[12]. Literature shows that meta-heuristic algorithms cannot perform optimally for both exploration and exploitation simultaneously [13]. Therefore, hybrid techniques are more trendy among practitioners where one algorithm is used for exploration and another for exploitation to enhance the performance of algorithms [14]-[17].

In some cases, parameters are adjusted, and operators are changed to improve the efficiency of algorithms [18]. Flower Pollination Algorithm is one of the best algorithms in terms of minimal numbers of parameters, and it can be easily implemented and is also highly efficient [19]. The mentioned properties of the algorithm had motivated us to select it for enhancement.

In the literature, Flower Pollination Algorithm has been improved by practitioners to address the Switch probability, global pollination and local pollination. The Local Neighborhood Search Strategy (LNSS) [20] is used to increase the local search-ability of FPA by diversifying the local neighborhoods. Colonel search is introduced to control the local search space, in such a way that initially more global search as compared to the end of the search process. In [21], a static scaling factor is applied to control the mutation through the local pollination process and to enhance the convergence rate of the algorithm. In [16], switch probability is replaced by two dynamic weights to guide for fast convergence and to increase the stability. Differential Evolution and Flower Pollination Algorithms are hybridized to escape the FPA from local minima. Chaos theory and Flower Pollination algorithm are hybridized to enhance the convergence rate and accuracy of the optimal solution [22].

In this paper, the proposed Flower Pollination Algorithm has two improvements over the Standard Flower Pollination Algorithm. The one is Dynamic switch probability and the second one is swap operator [23]. The Switch probability is an operator that controls the balance between diversification and intensification. The swap operator is used for modification in local search to enhance its efficiency and escape from trapping in multi-local minima [24]. The proposed algorithm is tested on benchmark optimization functions of multimodal and unimodal. It is evident from experimental results that the proposed algorithm, Improved Flower Pollination with Dynamic switching probability, and Swap operator (IFPDSO) relatively outperformed the typical Flower-Pollination Algorithm (FPA) and the other wellknown algorithms like Simulated Annealing (SA), Genetic Algorithm (GA), Firefly Algorithm (FF), Bat Inspired Algorithm (BA) and Modified flower pollination algorithm (MFPA). In Section II, we will discuss the Flower Pollination Algorithm in detail with its characteristics and the modifications of the proposed algorithm IFPDSO. In section III, the improved algorithm performance is examined and evaluated on well-known benchmark mathematical functions. In section IV, the conclusion is discussed.

II. MATERIALS AND METHOD

This section describes the Standard Flower Pollination algorithm and the improvements have been done in Flower Pollination algorithm.

A. Standard Flower Pollination Algorithm

In 2012, Xing-She Yan developed the flower pollination algorithm (FPA) [12]. It is a nature-inspired metaheuristic, stochastic technique. The practitioner was inspired by the pollination conduct of various flowers to pollinate for reproduction. Some flowers only try to attract insects which help to gear up the pollination process. These pests are the main pollinators during the global pollination process. The practitioner focused on complex nonlinear problems that are not appropriately handled by conventional optimization methods. There are basically two types of pollination:

- *Biotic*: this is a type of elongated distance or cross pollination which requires pollinators to execute it, such birds, pests, bees. Thus, the levy's flight performs global search that covers 90% of overall pollination.
- *Abiotic*: it is a self-pollination or local pollination which happens within a flower that does not need any pollinators. In this case, flowers pollinate through wind and diffusion. It is only 10% of pollination.

- Constancy: It is the reproduction capacity of two similar flowers which can increase insects.
- Switch Probability: It is helpful for controlling local pollination and global pollination. The switch probability p ε [0 1].

Other features in the pollination process include flower constancy which can be measured as the reproduction probability and switch probability p ε [0,1] which is helpful to control the global and local pollination. The fixed value p= 0.8 is slightly biased for exploitation. Global pollination is guaranteed with the help of creatures that ensure the best reproduction. It is denoted by "g*" and the mathematical representation for global pollination can be expressed as:

$$x_i^{t+1} = x_i^t + \gamma L(\Lambda) (x_i^t - g^*)$$
(1)

In Equation (1) ' x_i ' represents the solution vector in the iteration "t", while "g*" is the finest existing solution among the overall iterations. where ' γ ' represents a scaling factor which controls step size. In Equation (2) the 'L' is a levy's distribution parameter that resembles the strong point of pollination.

$$L \sim \left[\Lambda T(\Lambda) \; \frac{\operatorname{sine}\left(\frac{\Lambda \pi}{2}\right)}{\pi s(\Lambda + 1)} \right] \qquad s \gg s_0 > 0 \tag{2}$$

L>0 for an extensive random walk

Where local pollination is illustrated in Equation (3).

$$x_i^{t+1} = x_i^t + \in (x_i^t - x_k^t)$$
(3)

In Equation (3) where ' x_j ' and ' x_k ' represent the pollen from various flowers of identical plant species, which fundamentally imitate the flower constancy in a partial neighborhood. On the other hand, if x_j^t and x_k^t are selected from the identical species, it consistently becomes a local random walk where \in is carefully chosen from a uniform distribution in the range [0, 1]. The Pseudo-code of the SFPA algorithm has been discussed below.

1) The Pseudo-code of SFPA

The pseudo-code is separated into three portions. The first portion represents the initialization of the population and its parameters. The second portion decides that either global pollination should occur or local pollination. In the third section, the solution is updated to display.

- 1. Fitness function min or max f(x) with dimension, d. $x = (x_1, x_2, x_3, \dots, x_d)$
- Initialize the population of `n' flowers with its random solutions
- Calculate the finest result g* from the preliminary random solutions
- 4. Initialize the Switching probability p ϵ [0,1]
- 5. While (t < Maximum-Iterations)
- 6. For i = 1: n (total number of
- solutions)
- 7. if rand < p
- Induce 'd' dimensional stepping Vector is L that follows the Levy's distribution.



- 18. end while
- 19. output (best Minimum or Maximum)



Fig. 1 The proposed algorithm's flowchart.

B. Proposed Algorithm

To address the problem of balancing between the exploration and exploitation, a number of practitioners introduced various modifications in SFPA to enhance its convergence rate and efficiency. In the above discussion, some modified versions of SFPA have been submitted by the authors in [20][16][19] [25][26][18]. All these practitioners have addressed one or more issues stated earlier but failed to handle all these problems simultaneously that have been identified in SFPA. Literature shows that the standard flower

pollination algorithm is slightly weak in local optima for solving multimodal optimization problems [24], which may lead to trapping in local minima.

In this research paper, An Improved Flower Pollination algorithm (IFPDSO) has been proposed by introducing Dynamic Switch probability and Swap Operator to tackle the identified issues. In Fig 1, flow of the proposed algorithm has been illustrated. The Dynamic switch probability is applied to control the exploration and exploitation while the Swap operator is introduced to enhance the diversification of the population in the local search process to escape from being trapped in a local optimum. The random number epsilon ε of the uniform distribution is replaced with $\beta \in [0 \ 0.5]$ to enhance its exploitation capability.

The Swap operator has been defined in the given below equation.

$$S_{i}^{t+1} = \begin{cases} y_{i,k} & rand_{i,k} \ [0\ 1] < S_{r} \\ S_{i,k}^{t} & else & i=1,2,3....n \\ k & 1,2,3.....d \end{cases}$$
(4)

In Equation (4), s_i^k shows 'kth' dimension of the solution vector 'i' where 'Sr' represents the swap over rate which is 0.4 to diversify the population to get the best local optimum and evade the premature convergence.

However, the static switch probability (p = 0.8) has been used in the original Flower Pollination Algorithm, which creates partiality between exploitation and exploration. The proposed algorithm will apply dynamic switch probability to adjust the balancing issue between global and local pollination to get an optimal solution. Dynamic switching probability can be illustrated in Equation (5).

$$P = 0.8 + 0.1(Max_{iter} - t)/Max_{iter}$$
(5)

In Equation (5), Max_{iter} represents the maximum iterations while 't' denotes current iteration.

1) The proposed Algorithm's Pseudocode

The proposed algorithm begins with the initial population, switch probability which is followed by global pollination and local pollination, while in local pollination a swap operator has been presented. After completing each iteration, the optimal solution is updated hence dynamic switch probability is applied for the next iteration.

- 1. Start
- 2. Fitness function f(x) (min or max) with dimension $dx = (x_1, x_2, x_3, \dots, x_d)$
- Initialization: population is 'n' number 3. of flowers, swap over rate, random solutions
- 4. Evaluate optimal solution, g* from the preliminary random results
- 5. Initialize switch probability which lies in p ε [0,1]
- 6. While (t < Max-Iterations)
- 7. For i = 1: n (Maximum solutions)
- 8. if rand < p
- 9. Find 'd' is dimensional step vector L that obeys Levy's distribution
- 10. Global pollination is implemented to achieve new solution S_i^{t+1}

^{11.}else

12.	Loc	cal j	pollinati	on is	s search	n pr	cocess	
	to	Get	solution	S_i^{t+1}	vector	by	using	
	swap operator							
13.	enc	l if						

- 14. Find the new fitness S_i^{t+1} using a fitness function
- 15.if fitness $(S_i^{t+1}) < \text{fitness} (S_i^t)$
- (Minimization of function)
- 16.Update (Initial solution) flower S_i^t
- 17.end if
- Implement the dynamic switch probability, p
- 19. end for
- 20. end while
- 21. Output (Min or Max) optimal solution

III. RESULTS AND DISCUSSION

In this section experimental results have been discussed and have proved the outstanding performance of the proposed algorithm. The primary parameter setting is used to validate the performance of IFPDSO as compared to the six wellknown optimization algorithms in Table I.

 TABLE I

 THE PARAMETERS USED IN ALGORITHMS.

Initial setting of parameters
n = 60, switch probability static $P = 0.8$, scaling factor
$\gamma = 0.01$, step Levy flight is $\lambda = 1.5$
n = 60, switch probability initially P = 0.8, $\gamma = 0.01$,
step Levy flight is λ = 1.5, Swapping rate is Sr = 0.4,
range Beta 0.1< β <0.9
n = 60, switch probability $P = 0.8$, scaling factors ist
$\gamma 1 = 1$, second $\gamma 2 = 3$, step Levy flight is $\lambda = 1.5$,
and cloning array = [8 7 6 5 4 3 2 1 1 1 1 1 1]
n= 60, Cross over =0.8 and mutation function with
Scale = 1 (Gaussian) and shrink = 1
n=60, pulse rate = 0.5 and minimum f = 0, Loudness
= 0.5, maximum f $= 2$
n=60, Randomness (alpha = 0.25), Absorption
efficient (gamma = 1, minimum attractiveness: firefly
is (heta = 0.2)) $\gamma = 0.01$ Levy flight step $\lambda = 1.5$
Annealing Fnc: Fast annealing, Initial temperature =
100 \mathbf{P}_{2} suggestive intermed = 100

The proposed algorithm is evaluated on the 18 multimodal and unimodal complex functions. The results of the Modified Flower Pollination Algorithm (MFPA) are selected from [18], for a fair comparison, the number of generations for each algorithm N= 1500, population size n= 50 and n= 30 independent runs are executed. The performance will be analyzed by ranking the proposed algorithm IFPDSO and five other algorithms. The mean absolute error (MAE) is the first statistical analysis.

$$MAE = \frac{\sum_{i=1}^{N} |m_i - b_i|}{N}$$
(6)

Where m_i , indicate the mean value, bi is the best optimal solution and N is the number of independent runs. In Table: IV. The maximum cost (Max), minimum cost (Min), the average cost (Mean) and the standard deviation (Std.) are the results of random runs of every algorithm for the benchmark functions in Table V. The results of the proposed algorithm are highlighted in bold font.

TABLE II
RANKING OF THE ALGORITHMS (UNIMODAL)

Algorithm	MAE	Rank
IFPDSO	3.375E-57	1
MFPA	6.3858E-33	2
FPA	4.9812E-12	3
GA	6.5606E-12	4
BAT	2.1499E-10	5
FF	3.62159E-9	6
SA	4.3 2669E-4	7

 TABLE III

 RANKING OF THE ALGORITHMS (MULTIMODAL)

Algorithm	MAE	Rank
IFPDSO	9.87E-17	1
MFPA	0.067145	2
FPA	0.0671478	3
FF	12.84937	4
BAT	15.125612	5
SA	45.140746	6
GA	109.5325	7

 TABLE IV

 Ranking of seven algorithms (ALL functions)

Algorithm	MAE	Rank
IFPDSO	5.565E-17	1
MFPA	0.0524615	2
FPA	0.0559426	3
FF	10.055946	4
BAT	11.837543	5
SA	35.327635	6
GA	85.721286	7

Table II shows the ranking of the algorithms in solving the unimodal functions while Table III and Table IV represent the ranking for multi-modal benchmark function and overall average ranking using Mean Absolute error. The ranking shows that the proposed IFPDSO algorithm performs better than GA, FF, SA, BAT, MFPA and standard Flower Pollination algorithm.

Secondly, the comparison on convergence rate and stability between the proposed algorithm, SFPA and other well-known meta-heuristic algorithms which have been mentioned has been carried out. All these figures which include Fig.2 to Fig.7 have been plotted against the number of generations and minimum cost. In each graph, the solid red line shows the presentation of the proposed algorithm. The analysis proves remarkable convergence of the proposed algorithm in each plot of benchmark functions. All the graphs have been generated in Mat lab software R2018 by using the system Lenovo i3 ThinkPad.

No.	Functions	Equation of Functions	Range
1	Sphere function	$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{d} \mathbf{x}^2$	[-5,5]
2	3- hump camel function	$F(x,y) = 2x^2 - 10.5x^4 + \frac{x^6}{6} + xy + y^2$	[-5,5]
3	Powell function	$F(x) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	[-4,5]
ŀ	Matyas function	$F(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	[-10,10]
;	Griewank function	$F(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	[-600,600]
5	Ackley function	$\mathbf{F}(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}cx_i^2\right) + a + \exp(1)$	[-32,32]
,	Easom's function	$F(x) = (-1)^{d+1} \prod_{i=1}^{d} \cos(x_i) \exp\left[-\sum_{i=1}^{d} x_i - (x_i - \pi)^2\right]$	[-100 100]
	Rastrigin,s function	$F(x) = \pi * 10 + \sum_{i=1}^{d} (x_i^2 - 10 * \cos(2xx_i))$	[-5.1 5.1]
	Zaharov's function	$F(x) = \sum_{i=2}^{d} x_i^2 + \left(\frac{1}{2} \sum_{i=1}^{d} i x_i\right)^2 + \left(\frac{1}{2} \sum_{i=1}^{d} i x_i\right)^4 - 5$	[-5 10]
0	Rosenbrock's function	$F(x) = \sum_{i=1}^{d} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	[-5 5]
1	Cross-in-tray function	$F(x) = -0.0001 \left \left(\sin(x_1) \sin(x_2) \exp\left(\left 100 - \frac{\sqrt{x_1^2 + x_2^2}}{x} \right \right) \right + 1 \right)^{0.1}$	[-10 10]
2	Drop- wave function	$F(x) = -\frac{1 + \cos(\frac{12}{\sqrt{x_1^2 + x_2^2}})}{25(\sqrt{x_1^2 + x_2^2})}$	[-5.1 5.1]
3	Eggholder function	$F(x) = -(x_2 + 47) + \sin\left(\sqrt{\left x_2 + \frac{x_1}{2} + 47\right }\right) - x_1 \sin\left(\sqrt{\left x_1 + \frac{x_2}{2} + 47\right }\right)$	[-5.1 5.1]
4	Holder table function	$F(x) = - \sin(x_1)\cos(x_2)\exp\left(\left 1 - \frac{\sqrt{x_1^2 + x_2^2}}{x}\right \right) $	[-10 10]
5	Schaffer function N2	$F(x) = 0.5 + \frac{\sin^2(x_1^2 + x_2^2) - 0.5}{ 1 + 0.001(x_1^2 + x_2^2) ^2}$	[-100 100]
6	Shubert function	$F(x) = (\sum_{i=1}^{5} i cos(i+1)x_i + i))(\sum_{i=1}^{5} i cos(i+1)x_i + i))$	[-5.1 5.1]
7	Schwefel function	$F(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{ x_i })$	[-500 500]
8	Beale function	$F(x) = (1.5 + x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	[-4.5 4.5]

TABLE VI
STATISTICAL ANALYSIS OF ALGORITHMS

Function	Algo	Min	Max	Mean	Std	
Sphere	GA	5.992e-16	1.582e-13	1.362e-13	2.413e-13	
Function	BAT	8.973e-13	1.293e-10	1.923e-11	2.142-11	
	SA	5.524e-37	5.524e-36	6.403e-36	1.354e-36	
	FF	1.107e-12	1.606e-10	3.742e-11	3.726e-11	
	FPA	8.543e-33	2.543e-25	7.923e-27	3.753e-26	
	MFPA	2.989e-70	6.011e-60	2.005e-61	1.097e-60	

	IFPDSO	4.14e-186	1.91e-180	1.97e-181	0	
Three	GA	7.87E-09	2.47E-07	6.19E-08	7.60E-08	
Hump	ABC	1.64E-26	5.17E-24	1.43E-24	1.81E-24	
Camel	SA	3.94E-27	1.12E-24	2.95E-25	4.54E-25	
Function	FPA 1.32E-27		1.98E-20	2.02E-21	6.24E-21	
/	IFPDSO	2.73e-110	5.62e-100	5.62e-101	1.77e-100	
Powell	GA	5.523e-14	1.883e-12	1.032e-11	1.192e-11	
Sum	ABC	1.96E-34	6.58E-33	7.80E-33	1.64E-32	
Function	SA	7.41E-31	1.36E-27	1.70E-28	4.22E-28	
	IFPDSO	4.69E-156	1.74E-148	2.89E-149	5.38E-149	

Cross-in- tray	GA	-2.06408	2.03473	-2.0386	5.660E-03						
	IFPDSO	0	3.28E-29	5.61E-30	1.34E-29						
	MFPA	0	9.66E-30	3.30E-31	1.76E-30						
	FPA	1.60E-22	2.91E-19	8.73E-20	8.43E-19						
	SA	2.54E-03	6.7234	4.5673	3.5642		IFPDSO	0	0	0	0
	FF	3.575e-09	7.396e-07	6.196e-07	1.883e-07		MEPA	0	0	0	0
Function	BAT	2.31E-8	1.4056	1.96E-01	1.32E-01		FPA	7.06E-27	2.76E-20	1.46E-21	8.58E-21
Rosenbrock	GA	3.72E-06	8.30E-03	2.49E-03	1.27E-04		FF SA	3.52E-13 3.164E-6	2.06E-10 4.73E-02	6.25E-11 3.30E-3	1.40E-11 1.7E-3
	IFPDSO	6.52e-158	1.15e-154	1.49e-155	0	Function	BAT	2.26E-12	6.26E-01	1.17E-01	2.70E-01
	MFPA	1.24E-71	5.57E-42	2.18E-44	1.03E-43	Beal	GA	-2.0548	2.0373	-2.0346	5.87E-03
	FPA	2.64E-31	6.20E-25	6.22E-26	1.952e-26		MFPA IFPDSO	2.55E-05	2.560E-05	2.551E-05	0
	FF	7.501e-13	5.753e-10	2.563e-10	1.354e-10		FPA	2.63E-05	2.56E-05	2.54E-05	0
	SA	1.07E-09	7.40E-03	3.64E-05	3.35E-04		FF SA	2.55E-05 2.65E-05	7.396e-01 2.55E-05	6.196e-02 2.54E-05	1.883e-07 0
Function	BAT	1.05E-12	6.70E-10	2.13E-11	2.07E-11	Function	EE	2.55E-05	1.40E-01	1.96E-02	1.32E-06
Zakharov	GA	8.932e-16	1.142e-12	2.684e-13	3.952e-13		IFPDSO BAT				
	IFPDSO	8.880E-16	8.880E-16	8.880E-16	0		MFPA	-186.730 -186.731	-186.730 -186.731	-186.730 -186.731	5.38E-15 1.66E-13
	MFPA	8.87E-16	8.88E-16	8.88E-16	0		FPA	-186.730	-186.730	-186.730	1.56E-13
	FPA	3.22E-14	2.12E-12	6.01E-12	8.07E-12	T unotion	SA	-186.731	-186.731	-186.731	9.25E-07
	SA	3.26E-13	3.39E-12	1.60E-12	1.07E-12	Shubert Function	GA BAT	-69.9381 - 186.731	-16.611 -71.4109	-24;3817 -173.261	13.1135 31.9251
	FF CA	9.278E-6	1.334E-4	8.17/E-5	3.157E-5		IFPDSO	U	U	0	0
Function	BAT	5.79E-06	6.2578	2.46432	2.20356		MFPA	0	0	0	0
Ackley	GA DAT	1.2/E-07	7.20E-06	1.96E-06	2.12E-06		SA FPA	9.61E-03 0	0	0	0
A .11	IFPDSO	0	0		0		S A	0.615.02	1.50E-01	8.22E-01	6.07E-01
	MFPA	0	0	0	0		FF	1.01E-12	5.54E-11	1.72E-11	1.48E-11
	rpa Meda	0	2.91E-12	8./3E-13	6.43E-13	Function	BAT	1.82E-14	1.48E-01	3.40E-01	3.79E-01
	SA	3.82E-8	8.84E-01	5./4E-01	0.84E-01	Scheffer N2	GA	1.89E-11	1.50E-04	1.14E-05	1.49E-
	rf S A	3.3/3e-10	/.396e-8	0.190e-9	1.8836-9		MFPA IFPDSO	-19.2085 -19.2085	-19.2085 -19.2085	-19.2085 -19.2085	7.81E-15 3.61E-15
Function	BAI	2.51E-10	1.42E-01	1.86E-01	1.42E-01		FPA	-19.2084	-19.2084	-19.2084	3.62E-15
Rastrigin's	GA DAT	3.72E-13	8.30E-02	2.49E-02	1.27E-01		SA	2.54E-03	6.7234	4.5673	3.5642
Dest. ¹	IFPDSO	-1	-1	-1	1.275.01	Function	BAT FF	2.31E-8	1.4056 7.396a 07	1.96E-01	1.32E-01
	MFPA	-1	-1	-1	0	Holder Table	GA	3.72E-06	8.30E-03	2.49E-03	1.27E-04
	FPA	-1	-1	-1	0		IFPDSO	-959.608	-959.608	-959.608	1.15E-13
	SA	-1	0	-0.6068	0.5432		MFPA	-959.608	-959.608	-959.608	1.15e-13
Function	BAT	-1	0	-0.3756	0.3795		FF FPA	-959.608 -959.608	-954.125 -959.608	-954.634 -959.608	186.503 1.15e-13
Easom	GA	-1	-1	-1	2.12E-13	Function	BAT SA	-955.608 -959.608	-943.305 -886.432	-947.152 -895.342	179.056 166.542
	IFPDSO	0	0	0	0	holder	GA	-951.963	-940.342	-941.642	109.785
	MFPA	0	0	0	0	Faa	IFPDSO	-1	-1	-1	0
	FPA	1.60E-08	2.91E-05	8.73E-06	8.43E-05		MFPA	-1	-1	-1	0
	SA	0	0	0	0		FPA	-1	-1	-1	8.07E-10
	FF	3.575e-8	7.396e-3	6.196e-3	1.883e-3		SA	-0.9999	-0.78579	-0.9298	3.41E-01
Function		2.31E-12	1.40E-01	1.86E-01	1.42E-01		FF	-1	-1	-1	1.61E-09
Griewank	GA BAT	3.72E-06	8.30E-02	2.49E-02	1.27E-02	Function	BAT	-1	-0.7875	-0.93335	3.41E-02
<u>.</u>	IFPDSO	6.52E-63	1.15E-52	1.49E-56	3.64E-56	wave	GA	-0.99996	-0.96324	-0.95722	3.01E-02
	MFPA	1.44E-68	5.66E-50	2.08E-51	1.12E-50	Dron-	IFPDSO	-2.06261	-2.06261	-2.06261	1.35E-15
	FPA	2.64E-36	6.20E-25	6.22E-28	1.85e-27		MFPA	-2.06260	-2.06260	-2.06260	9.04E-16
	FF	7.501e-13	5.753e-11	2.563e-11	1.354e-11		FPA	-2.06251	-2.06251	-2.06261	8.59E-11
	SA	1.07E-25	7.40E-24	3.64E-24	3.35E-24		SA	-2.06261	-2.05231	-2.06103	1.07E-03
Function	BAT	1.05E-17	6.70E-16	2.13E-16	2.07E-16		FF	-2.06251	-2.06251	-2.06251	1.50E-11
Matyas,s	GA	8.932e-14	1.142e-11	2.684e-11	3.952e-11	Function	BAT	-2.16261	-2.16261	-2.16261	1.70E-11







Fig. 3 The Min cost for unimodal Three Hump Camel function



Fig. 4 The cost for unimodal Powell Function

The performance of IFPDSO is better because it inherits proper exploitation in the form of scaling factor ' γ ' and Levy flight 'L' in the global pollination. The exploitation capability is improved by limiting the parameter epsilon ' ϵ ' (uniform distribution) and swap operator to avoid premature convergence.



Fig. 5 The Cost for Multimodal Matyas Function



Fig. 6 The Cost for Multimodal Griewank Function



Fig. 7 The Cost for Multimodal Ackley Function

The right balance between exploration and exploitation is produced by the dynamic switch probability in order to increase its search ability. Another advantage of the IFPDSO algorithm is that it has a smaller number of parameters which will enhance its efficiency and speed up the processing for solving complex optimization problems. Moreover, its complexity is low as compared to other versions of the standard Flower Pollination Algorithm.

IV. CONCLUSION

In the last few decades, we have seen many applications of nature-inspired meta-heuristic techniques in solving various types of non-polynomial problems. The complex optimization problems have drawn the attraction of researchers due to a wide variety of issues that possess the nature of optimization. Therefore, new optimization algorithms have been introduced to achieve better results, for example gradient-based and stochastic techniques, but swarm intelligence has become the most useful tool among evolutionary algorithms. The standard flower pollination is also one of the best nature-inspired metaheuristic algorithms. It has many advantages yet there are a few drawbacks over other swarm intelligence techniques. Various types of modifications are introduced to overcome these drawbacks, but most of these approaches fail in obtaining the most optimum solution for some complex problems. In this study, the flower pollination algorithm is improved by modification in local pollination using the swap operator and dynamic switch probability. It has proved to be a robust optimization method with fewer parameters. The above results prove that the proposed algorithm comparatively performed better than the flower pollination algorithm, Genetic Algorithm, Simulated Annealing and Bat Algorithm, Firefly Algorithm and Modified Flower Pollination Algorithm. But the research shows that in optimization problems there is a room for further improvement.

ACKNOWLEDGMENT

This research is the work under the Trans-disciplinary Research Grant Scheme by the Ministry of Education Malaysia.

REFERENCES

- [1] J. Holland, "Adaptation in natural and artificial systems: an introductory analysis with application to biology," Control Artif. Intell., 1975.
- H. A. Firpi and R. J. Vogelstein, "Particle swarm optimization-based [2] feature selection for cognitive state detection," Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, pp. 6556-6559, 2011.
- [3] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," Simulation, vol. 76, no. 2, pp. 60-68, 2001.
- M. Dorigo, "Optimization, learning and natural algorithms," PhD [4]
- *Thesis, Politec. di Milano*, 1992. D. Karaboga, "Artificial bee colony algorithm," *scholarpedia*, vol. 5, no. 3, p. 6915, 2010. [5]

- Y. Halim and C. E. Nugraheni, "A Bee Colony Algorithm based [6] Solver for Flow Shop Scheduling Problem," JOIV Int. J. Informatics Vis., vol. 5, no. 2, 2021.
- [7] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," 2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc., pp. 210-214, 2009
- [8] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, 2010, pp. 65-74.
- X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," arXiv Prepr. arXiv1003.1409, 2010. [9]
- [10] B. Dimitris and S. John, "Simulated annealing.pdf," Statistical Science, vol. 8, no. 1. pp. 10-15, 1993.
- A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution [11] algorithm with strategy adaptation for global numerical optimization," IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 398-417, 2008.
- [12] X. S. Yang, "Flower pollination algorithm for global optimization," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7445 LNCS, pp. 240-249, 2012.
- M. I. Kamboh, N. Mohd Nawi, and R. Bt. Mohamad, "An improved [13] flower pollination solution for economic dispatch with valve point effect," Indones. J. Electr. Eng. Comput. Sci., vol. 22, no. 2, p. 629, 2021.
- [14] M. M. Sidhu and S. Mehta, "Hybrid Gravitational Search Flower Pollination Algorithm for Combined Economic Emission Load Dispatch," Int. Res. J. Eng. Technol., vol. 4, no. 7, pp. 1485-1493, 2017.
- [15] M. Abdel-Baset and I. Hezam, "A Hybrid Flower Pollination Algorithm for Engineering Optimization Problems," Int. J. Comput. Appl., vol. 140, no. 12, pp. 10-23, 2016.
- [16] D. Chakraborty, S. Saha, and O. Dutta, "DE-FPA: A hybrid differential evolution-flower pollination algorithm for function minimization," 2014 Int. Conf. High Perform. Comput. Appl. ICHPCA 2014.2015.
- G. E. Yuliastuti and A. M. Rizki, "Optimization of Multi-Product [17] Aggregate Production Planning using Hybrid Simulated Annealing and Adaptive Genetic Algorithm," vol. 10, no. 11, pp. 484-489, 2019.
- E. Nabil, "A Modified Flower Pollination Algorithm for Global [18] Optimization," Expert Syst. Appl., vol. 57, pp. 192-203, 2016.
- [19] S. Abdel-Fattah Sayed, E. Nabil, and A. Badr, "A binary clonal flower pollination algorithm for feature selection," Pattern Recognit. Lett., vol. 77, pp. 21-27, 2016.
- R. Wang and Y. Zhou, "Flower pollination algorithm with dimension [20] by dimension improvement," Math. Probl. Eng., vol. 2014, 2014.
- [21] H. M. Dubey, M. Pandit, and B. K. Panigrahi, "Hybrid flower pollination algorithm with time-varying fuzzy selection mechanism for wind integrated multi-objective dynamic economic dispatch," Renew. Energy, vol. 83, pp. 188-202, 2015.
- W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, and X. Fan, "A combined [22] model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting," Energy Convers. Manag., vol. 136, pp. 439–451, 2017.
- R. Salgotra and U. Singh, "Application of mutation operators to flower [23] pollination algorithm," Expert Syst. Appl., vol. 79, pp. 112-129, 2017.
- [24] R. Wang, Y. Zhou, C. Zhao, and H. Wu, "A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation," Biomed. Mater. Eng., vol. 26, pp. S1345-S1351, 2015.
- L. Zhang, L. Zhang, S. Liu, J. Zhou, and C. Papavassiliou, "Three-[25] Dimensional Underwater Path Planning Based on Modified Wolf Pack Algorithm," IEEE Access, vol. 5, pp. 22783-22795, 2017.
- A. Mishra and S. Deb, "Assembly sequence optimization using a [26] flower pollination algorithm-based approach," J. Intell. Manuf., vol. 30, no. 2, pp. 461-482, 2019.