ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Impact Factor: 8.206

Volume 9, Issue 4, April 2026

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

An Envisioned Approach for Discovering the Frequent Path Detection by using FPA and BBA Algorithm

Karthik S

Assistant Professor, Department of Information Technology, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamil Nadu, India

ABSTRACT : In optimization techniques, many numbers of popularized natural algorithms provide best performance which handling in various problems. In this paper Flower pollination algorithm (FPA) is compared with Binary Bat algorithm (BBA). These two algorithms are scrutinize based on fmin elapsed time. Simulation results suggest Flower pollination algorithm (FPA) can execute enhanced than the Binary Bat algorithm (BBA) for discovering the frequent path detection. This paper presents the comparative analysis of FPA and BBA predicts the discovering the frequent path detection.

KEYWORDS: Frequent path detection, Flower Pollination Algorithm(FPA), Binary Bat Algorithm(BBA), Particle Swarm Optimization(PSO)

I. INTRODUCTION

Clustering is an important tool for variety of applications in data mining, data compression, statistical data analysis and vector quantization which is a process of grouping objects with similar properties. Any cluster should exhibit two main properties mainly low inter-class similarity and high intra-class similarity. it is an important method of the unsupervised learning. Optimization is used to improve the performance and competence. It is the process of searching to locate the best solution to a problem. The search for optimal solutions is complex for real world problem. The development of nature inspired algorithm works efficiency in problem solving for optimization to solve difficult problems. Most recently biologically inspired algorithms are developed. Applications of the FPA, BBA helps to solve complex and nonlinear problems in several areas. Optimization is the process of searching to locate the best solution to a problem.

Nature has been solving many problems for billions of years, and many kinds of biological systems have shown fascinating and remarkable efficiency in problem solving [12] [13] [14]. Recently two new nature inspired algorithm, namely the Flower pollination algorithm (FPA) and the Bat algorithm (BA), have been developed [15].

The Bat Algorithm is a swarm based algorithm which is established on the echolocation behaviour of bats. The bats use echolocation to find its prey [16]. The Bat algorithm is a very recent swarm intelligence algorithm, introduced in 2010 [17]. After that a few improvements of Bat algorithm have been proposed in the literature, e.g., [18] [19] [20] [21]. In this paper, we compare the performance of the flower pollination algorithm with the basic Bat algorithm [22] [17].

Particle Swarm Optimization algorithm (PSO) is proposed by James Kennedy and Russell Eberhart in 1995 [8]. PSO is one of the most used EAs. It is motivated by social behaviour of organisms such as bird flocking and fish schooling [9]. The PSO algorithm, while making adjustment towards "local" and "global" best particles, is similar to the crossover operation used by genetic algorithms.

II. LITERATURE SURVEY

The main purpose of a FPA is ultimately reproduction through pollination. Flower pollination is typically related with the transfer of pollen, and such transfer is linked with pollinators such as insects, birds, bats and other animals [1]. In [2] FPA was applied successively for different economic load dispatch problems. In [3] FPA was applied in nonlinear

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

algebraic systems with multiple solutions. In [4] binary FPA was applied to feature selection application. In [5] FPA algorithm was used for multiobjective optimization. In [6] Study of FPA algorithm for continuous optimization is presented. In [7] FPA algorithm with dimension improvement is introduced. BBA was a well-known nature-inspired optimization techniques like FPA, ACO, and PSO algorithms [9]. BBA is applied in continuous optimization in the context of engineering design optimization. BBA can deal with highly nonlinear problem efficiently and can find the optimal solutions accurately [10].

1. Overview of Nature-Inspired Metaheuristics

The field of optimization has been revolutionized by biologically inspired algorithms that mimic the efficiency of natural systems. According to Yang (2015) and Deb (2014), these metaheuristics are designed to solve complex, non-linear, and multi-objective problems that traditional mathematical methods struggle to handle. The paper highlights that nature has spent billions of years refining problem-solving mechanisms, leading to the development of popular algorithms like Particle Swarm Optimization (PSO), which mimics the social behavior of bird flocking or fish schooling to find global optima.

2. The Flower Pollination Algorithm (FPA)

Introduced by Yang in 2012, the FPA is inspired by the reproductive processes of flowering plants. The literature characterizes its efficiency through four key rules, primarily focusing on the distinction between global pollination (biotic/cross-pollination via pollinators moving in Le’vy flights) and local pollination (abiotic/self-pollination).
Global vs. Local: The algorithm uses a switch probability (p) to balance exploration and exploitation. Applications: Previous studies have applied FPA to economic load dispatch problems (Prathiba et al., 2014), non-linear algebraic systems (Platt), and feature selection in binary formats (Rodrigues et al., 2015).
Performance: FPA is noted for its simplicity and the effective use of Le'vy flights, which allow for significant jumps in the search space, preventing the algorithm from getting stuck in local optima.

3. The Binary Bat Algorithm (BBA)

The Bat Algorithm (BA), introduced by Yang in 2010, is based on the echolocation behavior of micro-bats. Bats emit sound pulses and listen for echoes to distinguish between prey and obstacles, even in complete darkness.
Mechanics: The algorithm updates the position and velocity of "bats" in a d-dimensional space while adjusting frequency, loudness, and pulse emission rates. Binary Adaptation: For discrete problems like frequent path detection, the Binary Bat Algorithm (BBA) adapts these continuous movements into binary values [0, 1].
Refinements: Studies by Khan and Sahai (2016) and Rekaby (2015) have compared BA with Genetic Algorithms and Neural Network training, noting its prowess in global numerical optimization.

4. Frequent Path Detection and Data Mining

The paper situates these algorithms within the context of Data Mining and Clustering. Clustering is defined as the unsupervised process of grouping similar objects to ensure:
Low inter-class similarity: Distinct groups are as different as possible.
High intra-class similarity: Objects within a group are as similar as possible.
In the context of frequent path detection, optimization algorithms are used to search the vast space of possible paths to find the most efficient or recurring sequences.

III. METHODOLOGY

The Original Flower Pollination Algorithm

Flower Pollination Algorithm (FPA) was founded by Yang in the year 2012. Inspired by the flow pollination process of flowering plants are the following rules:
Rule 1: Biotic and cross-pollination can be considered as a process of global pollination process, and pollen-carrying pollinators move in a way that obeys Le'vy flights.
Rule 2: For local pollination, a biotic and self-pollination are used.
Rule 3: Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.
Rule 4: The interaction or switching of local pollination and global pollination can be controlled by a switch probability p□[0,1], with a slight bias toward local pollination.

In order to formulate updating formulas, we have to convert the aforementioned rules into updating equations. For example, in the global pollination step, flower pollen gametes are carried by pollinators such as insects, and pollen can travel over a long distance because insects can often fly and move in a much longer range[20]. Therefore, Rule 1 and flower constancy can be represented mathematically as:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - B)$$

Where is the pollen i or solution vector xi at iteration t, and B is the current best solution found among all solutions at the current generation/iteration. Here γ is a scaling factor to control the step size. In addition, L(λ) is the parameter that corresponds to the strength of the pollination, which essentially is also the step size. Since insects may move over a long distance with various distance steps, we can use a Le'vy flight to imitate this characteristic efficiently. That is, we draw L > 0 from a Levy distribution:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{S^{1+\lambda}}, (S >> S_0 > 0)$$

Here, Γ(λ) is the standard gamma function, and this distribution is valid for large steps s > 0.
Then, to model the local pollination, both Rule 2 and Rule Flower pollination algorithm
Define Objective function f (x), x = (x1, x2, ..., xd)
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution in the initial population
Define a switch probability p ∈ [0, 1]
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
while (t <MaxGeneration)
for i = 1 : n (all n flowers in the population)
if rand <p,
Draw a (d-dimensional) step vector L which obeys a L´evy distribution
Global pollination via) $x_i^{t+1} = x_i^t + L(B - x_i^t)$
else
Draw U from a uniform distribution in [0,1]
Do local pollination via) $x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$
end if
Evaluate new solutions
If new solutions are better, update them in the population
end for
Find the current best solution B
end while
Output the best solution found e 3 can be represented as:

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$$

Where and are pollen from different flowers of the same plant species. This essentially imitates the flower constancy in a limited neighborhood. Mathematically, if and comes from the same species or selected from the same population, this equivalently becomes a local random walk if we draw U from a uniform distribution in [0, 1].Though Flower pollination activities can occur at all scales, both local and global, adjacent flower patches or flowers in the not-so-far-away neighborhood are more likely to be pollinated by local flower pollen than those faraway. In order to imitate this, we can effectively use the switch probability like in Rule 4 or the proximity probability p to switch between common global pollination to intensive local pollination. To begin with, we can use a naive value of p = 0.5as an initially value. A preliminary parametric showed that p = 0.8 might work better for most applications[20].

© 2026 IJMRSET | Volume 9, Issue 4, April 2026| DOI:10.15680/IJMRSET.2026.0904428

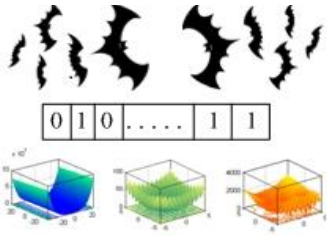
ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Binary Bat Algorithm.

Bats use echolocation to sense distance. The Echolocation works as a type of sonar, bats, mainly micro-bats, emit a loud and short pulse of sound, it hits into an object, after a fraction of time, the echo returns back to their ears. bats can compute from an object mechanism which makes bats being able to distinguish between a hurdle and a prey even in complete darkness. it flies arbitrarily with velocity v at point x with a frequency fmin, changing wavelength and noise A0 to search for victim. They can automatically adjust the wavelength (or frequency) adjust the rate of pulse with binary values [0, 1], depending on the nearness of their target, the loudness varies from a large (positive) A0 to a minimum constant value Amin.



Initial population is generated randomly for n number of bats. Each individual of the population consists of real valued vectors with d dimensions.

For the bats in simulations, we have to define the rules how their positions xi and velocities vi in ad-dimensional search space are updated. The new solutions xti at time step t are given by

$$fi = fmin + (fmax - fmin) \quad (1)$$
$$vt+1i = vti + (xti - x\_)fi \quad (2)$$
$$xt+1i = xti + vti, \quad (3)$$

where _ ∈ [0, 1] is a random vector drawn from a uniform distribution. Here x_ is the current global best location (solution) which is located after comparing all the solutions among all the n bats at each iteration t. As the product _ifi is the velocity increment, we can use fi (or _i) to adjust the velocity change while fixing the other factor _i (or fi), depending on the type of the problem of interest. In our implementation, we will use fmin = 0 and fmax = O(1), depending on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from [fmin, fmax].

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$xnew = xold + ϱ At, \quad (4)$$

where ϱ is a random number vector drawn from [−1, 1], while At =<Ati> is the average loudness of all the bats at this time step.

The update of the velocities and positions of bats have some similarity to the procedure in the standard particle swarm optimization, as fi essentially controls the pace and range of the movement of the swarming particles. To a degree, BA can be considered as a balanced combination of the standard particle swarm optimization and the intensive local search controlled by the loudness and pulse rate.

Loudness and Pulse Emission

The loudness Ai and the rate ri of pulse emission have to be updated accordingly as the iterations proceed. As the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. For example, we can use A0 = 100 and Amin = 1. For simplicity, we can also use A0 = 1 and Amin = 0, assuming Amin = 0 means that a bat has just found the prey and temporarily stop emitting any sound.

At+1

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

i = _At
i , rt
i = r0
i [1 − exp(−t)], (5)
where _ and are constants. In fact, _ is similar to the cooling factor of a cooling schedule in the simulated annealing (Kirkpatrick et al., 1983). For any 0 < _ < 1 and > 0, we have
At
i → 0, rt
i → r0
i , as t → ∞. (6)
In the simplest case, we can use _ = , and we have used _ = = 0.9 in our simulations.
Preliminary studies by Yang (2010a) suggested that bat algorithm is very promising for solving nonlinear global optimization problems. Now we extend it to solve multi objective optimization

Objective function:
1. F(x)x=(x1,x2,x3…x4)^t
2. Initialize bat population and velocity
3. Define pulse frequency
4. Initialize pulse rate and loudness
5. while (< maximum number of iterations)
6. Generate new solutions by adjusting frequency,
7. and updating velocities and locations/solutions
8. If (rand >) 9. Select a solution among the best solutions
9. Generate a local solution around the selected best solution
10. end if
11. If (rand < and
12. Accept new solutions
13. Increase , reduce
14. end if
15. Ranks the bats and find current best
16. end while
17. Display results.

Comparison between FPA and BBA Algorithm

BBA	FPA

Max_iteration=2000; % Maximum number of iterations
noP=30; % Number of particles
noV=100;
A=.25; % Loudness (constant or decreasing)
r=.1; % Pulse rate (constant or decreasing)
%n is the population size, typically 10 to 25
%Max_iter is the maximum number of iteration
% This frequency range determines the scalings
Qmin=0; % Frequency minimum
Qmax=2; % Frequency maximum
%Iteration parameters
N_iter=0; % Total number of function evaluations
% Initial arrays
	%Default parameters
	if nargin<1,
	para=[20 0.8];
	end
	
	n=para(1);% Population size, typically 10 to 25
	p=para(2);% probabibility switch
	
	%Iteration parameters
	N_iter=2000;
	% Total number of iterations
	
	% Dimension of the search variables
	d=3;
	Lb=-2*ones(1,d);
	Ub=2*ones(1,d);

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Q=zeros(n,1); % Frequency v=zeros(n,d); % Velocities Sol=zeros(n,d); cg_curve=zeros(1,Max_iter);	
% Initialize the population/solutions for j=1:d % For dimension	% Initialize the population/ solutions
% Find the current best [fmin,I]=min(Fitness); best=Sol(I,:);	% Find the current best [fmin,I]=min(Fitness); best=Sol(I,:); S=Sol;
% Start the iterations -- Binary Bat Algorithm	% Start the iterations -- Flower Algorithm
% Evaluate new solutions Fnew=CostFunction(Sol(i,:)); % If the solution improves or not too loudness if (Fnew<=Fitness(i)) && (rand<A) Sol(i,:)=Sol(i,:); Fitness(i)=Fnew;	% Evaluate new solutions Fnew=Fun(S(i,:)); % If fitness improves (better solutions found), update then if (Fnew<=Fitness(i)), Sol(i,:)=S(i,:); Fitness(i)=Fnew;

% Update the current best
if Fnew<=fmin,
best=Sol(i,:);
fmin=Fnew;
	% Update the current global best
	if Fnew<=fmin,
	best=S(i,:) ;
	fmin=Fnew ;

function [o] = MyCost(x) o=sum(x); % Modify or replace here according to your cost funciton end	% Objective function and here we used Rosenbrock's 3D function function z=Fun(u) % Draw n Levy flight sample function L=Levy(d) % Levy exponent and coefficient

Table 1: Comparison between FPA and BBA

IV. EXPERIMENTAL RESULTS

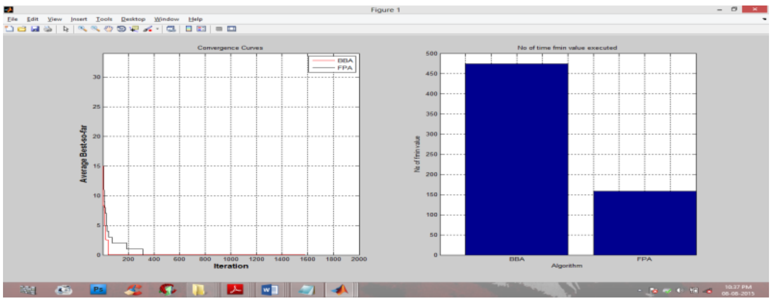


Figure 1. Performance Comparison using Line chart and bar chart

The results of elapsed time 6.003174 for BBA and Elapsed time are 2.472175 seconds for FPA with same kind of variable values. While compares the reaching best average in every iteration, FPA detects short length current best value (nearly 50), BBA doing some later performance (more than 200) other than FPA. Simulation results are done using matlab R2011a. Identifying the fmin values FPA having very less count more than thrice of BBA. FPA use to write best, fmin values for every 100 iteration only as per the way of algorithm. BBA provide the results for evaluations, In fmin values, FPA has chosen of 178 times and BBA having chosen of 474 times. FPA provides the best optimal solution in terms of fmin iterations and elapsed time as shown in the table 1.algorithm than BBA .From the above results FPA algorithm achieved lowest duration more advance over BBA algorithm.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. CONCLUSION

FPA and BBA algorithm evaluates in optimizing frequent path detection. These models were trained and tested with most detected values and same code executed more than 200 times. From the above results it found FPA algorithm achieved lowest duration more advance over BBA algorithm. The main purpose for determine frequent path detection using Flower Pollination algorithm and Binary Bat algorithm to discover the precision and frequent path. By evaluating the algorithms based on the iterations in future it can be implemented in different domains.

REFERENCES

[1] B. J. Glover, Understanding Flowers and Flowering: An Integrated Approach, Oxford University Press, (2017).
[2] R. Prathiba , M. Balasingh Moses , S. Sakthivel, Flower Pollination Algorithm Applied for Different Economic Load Dispatch Problems, International Journal of Engineering and Technology (IJET), Vol 6 No 2 Apr-May 2014, pp. 1009-1016.
[3] G.M. platt, Application of Flower Pollination Algorithm in nonlinear algebraic systems with multiple solutions, Engineering Optimization IV.
[4] D Rodrigues, X.S. Yang, A.N. de Souza, J.P. Papa, Binary Flower Pollination Algorithm and Its Application to Feature Selection , Springer, Recent Advances in Swarm Intelligence and Evolutionary Computation Studies in Computational Intelligence Volume 585, 2015, pp 85-100.
[5] S Yang, M. Karamanoglu , . He , Flower pollination algorithm: A novel approach for multiobjective optimization , Engineering Optimization, Volume 46, Issue 9, 2014.
[6] S. Łukasik, P. Kowalski, Study of Flower Pollination Algorithm for Continuous Optimization, Intelligent Systems'2014, Advances in Intelligent Systems and Computing Volume 322, 2015, pp 451-459.
[7] R. Wang and Y.Zhou, "Flower Pollination Algorithm with Dimension by Dimension Improvement," Mathematical Problems in Engineering, vol. 2014, Article ID 481791, 9 pages, 2014. doi:10.1155/2014/481791.
[8] D. N. WilkeAnalysis particle swarm optimization algorithm", Master's Dissertation, University of Pretoria, 2015.
[9] A.S. Khalil "An Investigation into Optimization Strategies of Genetic Algorithms and Swarm Intelligence". Artificial Life (2016
[10] X. S., Yang, A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. Cruz, C.; Gonzalez, J. R.; Pelta, D. A.; Terrazas, G), Studies in Computational Intelligence Vol. 284, Springer Berlin, pp. 65–74, 2010.
[11] X. S., Yang and A. H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Engineering Computations, Vol. 29, No. 5, pp.464–483, 2012
[12] H. A. Abbass and R. Sarker, "The Pareto diffential evolution algorithm," Int. J. Artificial Intelligence Tools, vol. 11(4), pp. 531-552, 2002.
[13] Y. Xin-She, "Nature-inspired Metaheuristic Algorithms," Luniver Press, 2015.
[14] K. Deb, Multi-Objective optimization using evolutionary algorithms, New York: John Wiley & Sons, 2014
[15]Y. Xin-She, Engineering Optimization: An Introduction with Metaheuristic Application, Wiley, 2010.
[16]G. Wang and L. Guo, "A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization," Journal of Applied Mathematics, vol. 2013, p. 21, 2013.
[17] Y. Xin-She, "A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimiztion (NISCO 2010)," Springer, vol. 284, no. Springer Berlin,, pp. 65-74 , 2010.
[18]K. Khan and A. Sahai, "A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context," I.J. Intelligent Systems and Applications, pp. 23-29, June 2016.
[19] N. S. S. M. M. S. A. Md. Wasi Ul Kabir, "A Novel Adaptive Bat Algorithm to Control Explorations nd Exploitations for Continuous Optimization Problems," International Journal of Computer Applications , vol. 94, no. 13 , 2014.
[20] A Rekaby, "Directed Artificial Bat Algorithm (DABA) A New Bio-Inspired Algorithm," in International Conference on Advances in Computing, Communications and Informatics (ICACCI), Cairo, 2015.
[21] Y. Selim and U. K. Ecir, "Improved Bat Algorithm (IBA) on Continuous Optimization Problems," Lecture Notes on Software Engineering, vol. 1, no. 3, pp. 279-283, 2017.
[22] G. Q. Huang, W. J. Zhao and Q. Q. Lu, "Bat algorithm with global convergence for solving large scale optimization problem," Application Research of Computers , vol. 30, no. 3, pp. 1-10, 2015.

INNO SPACE
SJIF Scientific Journal Impact Factor

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com