American Journal of Electrical and Computer Engineering
2018; 2(2): 31-36
http://www.sciencepublishinggroup.com/j/ajece
doi: 10.11648/j.ajece.20180202.14
ISSN: 2640-0480 (Print); ISSN: 2640-0502 (Online)

Solving Initial Value Problems by Flower Pollination Algorithm

Fatima Ouaar*, Naceur Khelil

Department of Mathematics, Faculty of Exact Sciences, Mohamed Khider University, Biskra, Algeria

Email address:
fatimaouaar@yahoo.fr (F. Ouaar), khelilna@yahoo.fr (N. Khelil)
*Corresponding author

To cite this article:
Fatima Ouaar, Naceur Khelil. Solving Initial Value Problems by Flower Pollination Algorithm. *American Journal of Electrical and Computer Engineering*. Vol. 2, No. 2, 2018, pp. 31-36. doi: 10.11648/j.ajece.20180202.14

Received: November 20, 2018; **Accepted**: December 13, 2018; **Published**: January 14, 2019

Abstract: Differential equations are very important in modeling many phenomena mathematically. The aim of this paper is to consider Initial Value Problems (IVPs) in ordinary differential equations (ODEs) as an optimization problem, solved by using a meta-heuristic algorithm which is considered as an alternative way to find numerical approximation of (IVPs) since they can almost be solved simply by classical mathematical tools which are not very precise. By selecting a methodical way based on the use of recent and efficient algorithm, that is, Flower Pollination Algorithm (FPA), inspired by the pollination process of flowers plants to solve approximately an (IVP) when a specified example is selected that is the exponential problem which have an imperative role to describe many real problems. The effectiveness of the proposed method is tested via a simulation study between the exact results, the FPA results and Euler method which is considerate as a classical tool to solve numerically an (IVP). The final results and after a comparison between the performance of FPA and Euler method in terms of solution quality shows that FPA yields satisfactorily precise approximation of the solution. That ensures the ability of FPA to solve such important problems and highly complexes problems efficiently with minimal error.

Keywords: Initial Value Problem (IVP), Optimization Problem, Exponential Model, Flower Pollination Algorithm (FPA)

1. Introduction

Many models of engineering systems involve the rate of change of a quantity. There is thus a need to incorporate derivatives into the mathematical model. These mathematical models are examples of differential equations. The subject of differential equations is one of the most interesting and useful areas of mathematics. We can describe many interesting natural phenomena that involve change using differential equations.

Let $f=f(x,y)$ be a real-valued function of two real variables defined for $a \le x \le b$, where a and b are finite, and for all real values of y. The equations

$$\begin{cases} y' = f(x,y) \\ y(a) = y_0 \end{cases} \tag{1}$$

are called initial-value problem (IVP); they symbolize the following problem: To find a function $y(x)$, continuous and differentiable for $x \in [a,b]$ such that $y' = f(x,y)$ from $y(a) = y_0$ for all $x \in [a,b]$ [12]. This problem possesses unique solution when: f is continuous on $[a,b] \times \mathbb{R}$, and satisfies the Lipschitz condition; it exists a real constant $k>0$, as:

$$| f(x,\theta_1) - f(x,\theta_2) | \le k \mid \theta_1 - \theta_2 \mid, \text{for all } x \in [a,b]$$

and all couple $(\theta_1,\theta_2) \in \mathbb{R} \times \mathbb{R}$.

Finding the optimal solutions numerically of an Initial-Value Problem (IVP) is gotten with approximations:

$y(x_0 + h), \ldots, y(x_0 + nh)$ where $a = x_0$ and $h = (b - a)/n$.

For more precision of the solution, a very small step size h must be used that includes a larger number of steps, thus more computing time which not available in the useful numerical methods like Euler and Runge-Kutta methods [12], which may approximate solutions of (IVP) and perhaps yield useful information, often sufficing in the absence of exact, analytic solutions.

There is a huge variety of real life problems optimized by using differential equations; when we have to find the optimal solution to a given problem under highly complex constraints. Such constrained optimization problems are often highly nonlinear; finding the optimal solutions is frequently very difficult charge. The majority of usual optimization techniques do not perform well for problems with nonlinearity and multimodality. Hence the nature-inspired meta-heuristic algorithms become trending in the optimization field by dealing with such difficult problems [10], and it has been shown that meta-heuristics are unexpectedly very efficient. For this reason, the literature of meta-heuristics has prolonged enormously in the last two decades.

Biological, physical, or chemical systems in nature are the subject of many nature-inspired meta-heuristic algorithms [26]. Swarm Intelligence has become popular among researchers working on optimization problems all over the world [23] which demonstrate its capability in solving many optimization problems they took a dissimilar forms according to the inspired process of the natural systems like Genetic algorithm [1, 9], Ant colony optimization algorithm [6], Bee algorithm [7, 15], Particle swarm optimization [4, 13], Bat algorithm [25], etc. All these algorithms have several advantages illustrated via a wide range of applications.

Among most recent, bio-inspired algorithms, flower pollination algorithm (FPA), has demonstrated very good efficiency in solving both single objective optimization and multi-objective optimization problems. This algorithm mimics the main characteristics of the pollination process of flowering plants, Pollination considered in this algorithm follows biotic pollination and cross pollination process which leads to both local and global search capabilities. After the improvement of the fundamental FPA by Xin-She Yang (2013) [24], many researchers from many field have utilize this algorithm into diverse applications and hybridize it with many algorithms in order to overcome its limitations via a lot of research papers available in the literature. For example, Chiroma *et al.* gave a brief review of some past applications of FPA [5]. Also Alam *et al.* carried out photovoltaic parameter estimation using FPA [3]. Sayed *et al.* [18] have treated the feature selection by using a clonal FPA. Moreover, Velamuri *et al.* used FPA to optimize economic load dispatch [20]; while a binary flower pollination algorithm to do EEG identification was developed by Rodrigues *et al.* [17]. Furthermore, Zhou *et al.* introduced an elite opposition-based FPA [27] as well as Mahdad *et al.* presented an adaptive FPA to solve optimal power flow problems [14], whereas Abdelaziz *et al.* solved placement problems in distribution systems using FPA [2]. New variants of FPA are still emerging [11, 19].

The importance of this paper resides by considering (IVPs) as an optimization problem implemented by means of FPA [24] in order to find numerical solutions for this problem, hence the obtained results are compared by those of Euler method and the exact results of the studied example. Comparisons are made in terms of solution quality.

This paper is organized as follows. The formulation of the problem is revealed in section 2; section 3 provides basics on FPA and its main steps for finding an approximate solution of (IVP). The Section 4 exposes an example to show how the FPA can lead to a satisfactory result for solving (IVP). The comments and conclusion are made in section 5.

2. Problem Formulation

2.1. Objective Function

The main idea behind the algorithm is to use the finite difference formula for the derivative and equation (1) we obtain,

$$\frac{y(x_j) - y(x_{j-1})}{h} \approx f\left(x_{j-1}, y\left(x_{j-1}\right)\right).$$

Thus,

$$\frac{y_j - y_{j-1}}{h} \approx f(x_{j-1}, y_{j-1}).$$

Consequently, the error formula is considered as:

$$[\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1})]^{\,2}$$

The objective function, associated to $Y = (y_1, y_2, \ldots, y_d)$ will be:

$$F(y) = \sum_{i=1}^{d} [\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1})]^{\,2} \qquad (2)$$

2.2. Consistency

Calculation of $Y=(y_1, y_2, \ldots, y_d)$ is very interested because it minimizes the objective function in equation (2). Taylor's Formula order 1 yields:

$$y_j = y_{j-1} + hy'_{j-1} + O(h^2), j = 1, \ldots, d.$$

So,

$$\frac{y_j - y_{j-1}}{h} = y'_{j-1} + O(h)$$

By subtracting $f(x_{j-1}, y_{j-1})$ from both sides of last equation:

$$\frac{y_j - y_{j-1}}{h} - f\left(x_{j-1}, y_{j-1}\right) = y'_{j-1} - f\left(x_{j-1}, y_{j-1}\right) + O, j=1,\ldots,d.$$

The last relation shows that the final value $Y=(y_1, y_2, \ldots, y_d)$ is an approximate solution of (IVP), for small value of *h*.

3. Flower Pollination Algorithm (FPA)

3.1. Flower Pollination Description

Pollination is very important. It leads to the creation of new seeds that grow into new plants. It begins in the flower. Flowering plants have several different parts that are important in pollination [21]. Flowers have male parts called stamens that produce a sticky powder called pollen. Flowers also have a female part called the pistil. The top of the pistil is called the stigma, and is often sticky. Seeds are made at the base of the pistil, in the ovule. To be pollinated, pollen must be moved from a stamen to the stigma [23]. There are two types of pollination:

1. Self Pollination (Abiotic pollination): Only about 10% of plants fall in this category, it's the fertilization of one flower, when the pollen from a flower pollinates the same flower or flowers of the same plant; it does not require any pollinators. It occurs when a flower contains both the male and the female gametes, is a process where the pollination happens without involvement of external agents [22] (Figure 1)

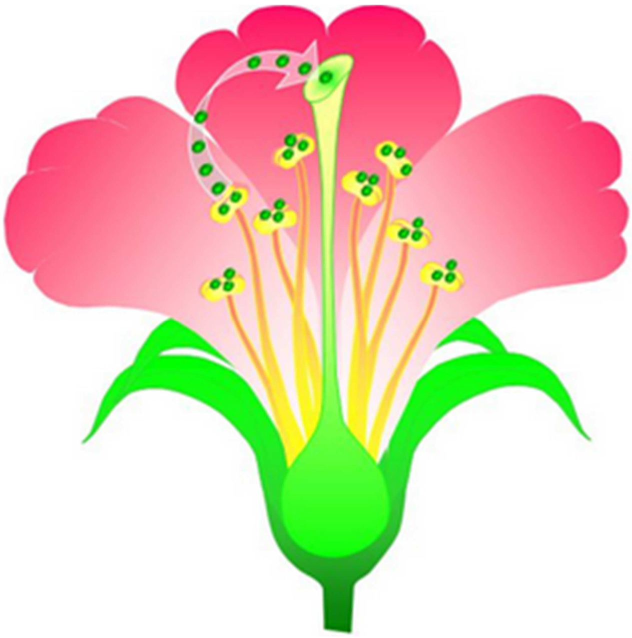

Figure 1. *Self pollination.*

2. Cross Pollination (biotic pollination): Is typically associated when pollen from a plant's stamen is transferred to a different plant's stigma (of the same species), and such transfer is often linked with pollinators. Pollination occurs in several ways:

   a. People: They can transfer pollen from one flower to another, but most plants are pollinated without any help from people.
   b. Animals: such as bees, butterflies, moths, flies pollinate plants by an accidental way when they are at the plant to get food. The pollinators can fly a long distance, thus they can consider as the global pollination [8]. In addition, bees and birds may behave as Levy flight behavior [16, 24], with jump or fly distance steps obey a Levy distribution. Furthermore, flower constancy can be used an increment step using the similarity or difference of two flowers [8, 21].
   c. Wind and Diffusion in water: it picks up pollen from one plant and blows it onto another (Figure 2).

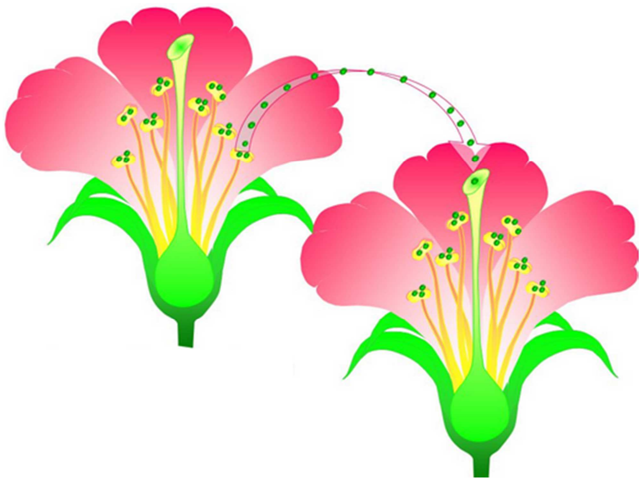

Figure 2. *Cross pollination.*

3.2. FPA formulation

The four rules given below are used to summarize the above characteristics of pollination process, flower constancy and pollinator behavior [21].

1. Biotic and cross-pollination is considered as global pollination process and pollinators carrying pollen move in a way that confirms to Levy flights.
2. For local pollination, abiotic pollination and self-pollination are used.
3. Flower constancy can be considered as the reproduction probability is proportional to the similarity of two flowers involved.
4. Local pollination and global pollination is controlled by a switch probability $p \in [0,1]$.

Since the pollination process happen by producing multiple flowers by each plant and the same every flower patch often release millions and even billions of pollen gametes, in FPA it was assumed that each plant only has one flower, and each flower only produce one pollen gamete for simplicity reasons. Thus, there is no need to distinguish a pollen gamete, a flower, a plant or solution to a problem. This simplicity means a solution x_i is equivalent to a flower and/or a pollen gamete.

Based on the previous discussions, FPA was designed on two principle supports that are the global pollination and the local pollination:

In the global pollination step, flower pollens are carried by pollinators such as insects, and pollens can travel over a long distance because insects can often fly and move in a much longer range. This ensures the pollination and reproduction of the fittest, and thus we represent the fittest as g_*. The first rule plus flower constancy can be represented mathematically as:

$$\mathrm{x}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{x}_\mathrm{i}^\mathrm{t} + \mathrm{L}(\mathrm{g}_* - \mathrm{x}_\mathrm{i}^\mathrm{t}),$$

where x_i^t is the pollen i or solution vector x_i at iteration t, and g_* is the current best solution found among all solutions at the current generation/iteration. The parameter L is the strength of the pollination, which essentially is a step size.

Since insects may move over a long distance with various distance steps, we can use a Levy flight to mimic this characteristic efficiently. That is, we draw $L > 0$ from a Levy distribution

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}};\ (s \gg s_0 > 0).$$

Here $\Gamma(\lambda)$ is the standard gamma function, and this distribution is valid for large steps $s > 0$.

The local pollination (Rule 2) and flower constancy can be represented as

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

Where x_j^t and x_k^t are pollens from the different flowers of the same plant species. This essentially mimics the flower constancy in a limited neighborhood. Mathematically, if x_j^t and x_k^t comes from the same species or selected from the same population, this become a local random walk if we draw ϵ from a uniform distribution in *[0,1]*. Most flower pollination activities can occur at both local and global scale [24].

In principle, flower pollination process can happen at both local and global levels. But in reality, flowers in the neighborhood have higher chances of getting pollinated by pollen from local flowers than those which are far away. To simulate this feature, a proximity probability (Rule 4) can be commendably used to switch between intensive local pollination to common global pollination. To start with, a raw value of $p=0.5$ may be used as an initial value. A preliminary parametric study indicated that $p=0.8$ may work better for most applications. To formulate the updating formulas, these rules have to be changed into correct updating equations.

The main steps of FPA or simply the flower algorithm [24] are illustrated via Table 1, while FPA flowchart is represented in Figure 1.

4. Example Results

To illustrate the treated method and to demonstrate its computationally efficiency, the exponential problem is considered by taking a uniform step size h.

The main motivation in the selection of the application example comes from the great importance of the exponential equation in modeling the real life problems. Exponential functions are frequently used to model the growth or depreciation of financial investments, population growth, radioactive decay, and phenomena where a quantity is allowed to undergo unrestrained growth.

In Table 3 the FPA results vs. the exact results for the studied problem is presented. Table 4 presents the absolute error. For convenience, the parameters settings to generate the FPA are presented in Table 2. All computations were performed on an MSWindow 2007 professional operating system in the Matlab environment version R2013a compiler on Intel Duo Core 2.20 Ghz PC.

Let a simple (IVP):

$$\begin{cases} \dfrac{dy}{dx} = y, 0 \le x \le 1 \\ y(0) = 1 \end{cases}$$

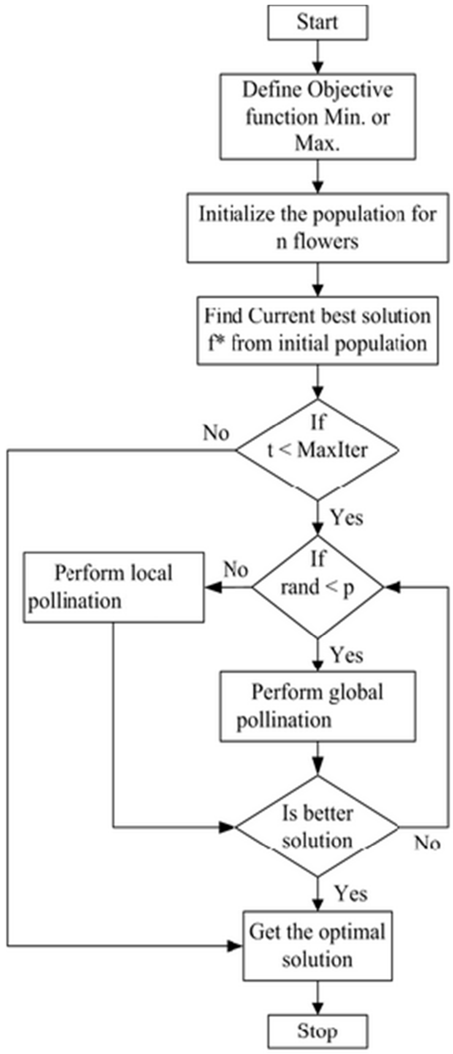

Figure 3. *Flowchart of FPA.*

For $d = 10, h =, x_0 = 0, y_0 = 1$, The exact solution is

$$y(x) = e^x.$$

The objective function

$$F(y_1, y_2, , \ldots, y_{10}) = \sum_{i=1}^{10} \left[\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1})\right]^2$$
$$= \sum_{i=1}^{10} \left[\frac{y_j - y_{j-1}}{h} - y_{j-1}\right]^2$$

Table 1. *Pseudo code of FPA.*

Pseudo code of the proposed Flower Pollination Algorithm
Objective min or max $f(x), x = (x_1, x_2, \ldots, x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution g_* in the initial population
Define a switch probability $p \in [0,1]$
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 Levy distribution
Global pollination via $x_i^{t+1} = x_i^t + L(g_* - x_i^t)$
else
Draw from a uniform distribution in [0,1]
Randomly choose j and k among all the solutions
Do local pollination via $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$
end if
Evaluate new solutions
If new solutions are better, update them in the population
end for
Find the current best solution g_*
end while

Parameters adopted for the FPA algorithm are given in Table 2:

Table 2. *Parameters adopted for the FPA.*

Parameter	Value
Dimension of the search variables (d)	10
Total number of iterations (N)	2000
Population size (n)	20
Probability switch (p)	0.8

The obtained results, the comparison of the method to the exact solution and Euler's method are shown in Table 3.

Table 3. *Exact and Numerical results by FPA method of example for d=10.*

i	x_i	Exact Results	FPA Results	Euler Results
1	0.1	1.1052	1.1053	1.1000
2	0.2	1.2214	1.2215	1.2100
3	0.3	1.3499	1.3492	1.3310
4	0.4	1.4918	1.4907	1.4641
5	0.5	1.6487	1.6467	1.6105
6	0.6	1.8221	1.8191	1.7716
7	0.7	2.0138	2.0093	1.9487
8	0.8	2.2255	2.2195	2.1436
9	0.9	2.4596	2.4524	2.3579
10	0.1	2.7183	2.7094	2.5937

Numerical results in Table 3 are illustrated in the Figure 4. The comparison between the performances of FPA and Euler face to the exact results confirm that FPA is better than Euler because it has a very close curve to the exact curve contrary to Euler method.

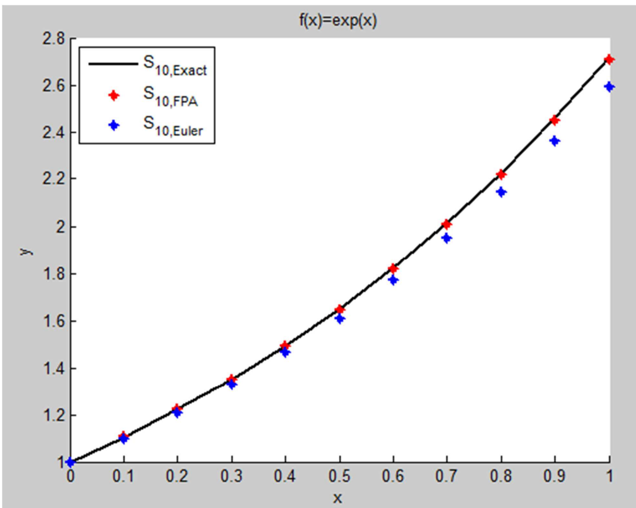

Figure 4. *Numerical solutions plot of example for d=10.*

Table 4 presents the absolute error between exact results and the studied methods outcomes.

Table 4. *Absolute Error of FPA and Euler methods of studied example.*

i	x_i	Abs. Error of FPA method	Abs. Error of Euler method
1	0.1	0.0001	0.0052
2	0.2	0.0001	0.0114
3	0.3	0.0007	0.0189
4	0.4	0.0011	0.0227
5	0.5	0.0020	0.0382
6	0.6	0.0030	0.0505
7	0.7	0.0045	0.0651
8	0.8	0.0060	0.0819
9	0.9	0.0072	0.1017
10	0.1	0.0089	0.1246

The Figure 5 shows their graphical representations. In both representations of the absolute error, FPA method offers a very negligible absolute error compared to Euler method.

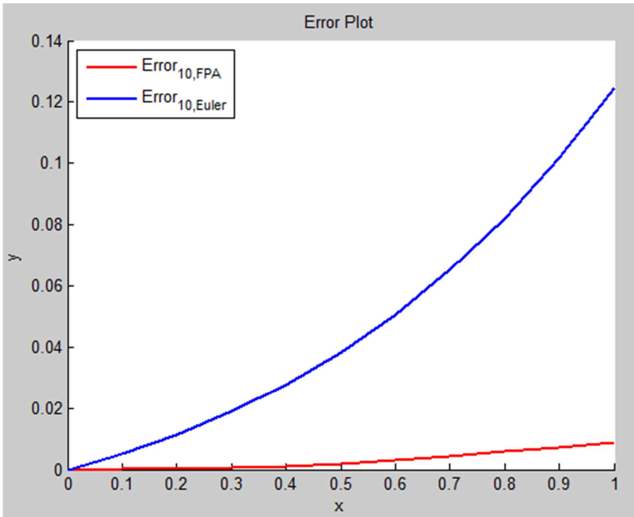

Figure 5. *Absolute Error plot of example for d=10.*

5. Discussions

The FPA have successfully developed to take off the characteristics of flower pollination. Our simulation results indicate that FPA is simple, reduces time, flexible and exponentially better to solve optimization (IVP).

6. Conclusion

The flower pollination algorithm is an efficient optimization algorithm with a wide range of applications. In this study, we apply the FPA to solve approximately an (IVP), via a chosen example and after a comparison between the exact solutions, the algorithm outcomes and Euler method results; FPA was found exponentially better by offering accurate solutions with smallest amount error.

It is important pointing out that the current results are mainly for the standard flower pollination algorithm. It will be useful if further research can focus on the extension of the proposed methodology to optimize (IVP) by other variants of

FPA. Ultimately, it can be expected that the proposed problem can be optimized by other meta-heuristic algorithms as well.

FPA having remarkable ability to solve a wide range of problems and highly non linear problems efficiently, it works well with complicated problems, As a future research, there are profound studies on FPA that will improve the algorithm such as the parameter tuning, parameter control, speedup of coverage, using of more diverse parameters, more extensive comparison studies with more open sort of algorithms…etc. Also, FPA should be applied in several applications of engineering and industrial optimization problems.

Acknowledgements

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] D. H. Ackley, (1987), “A Connectionist Machine for Genetic Hillclimbing”, Kluwer Academic Publishers.

[2] A. Y. Abdelaziz., E. S. Ali., S. M. Abd Elazim., (2016), “Flower pollination algorithm and loss sensitivity factors for optimal sizing and placement of capacitors in radial distribution systems”, Int. J. Electrical Power and Energy Systems, 78 (1), 207-214.

[3] D. F. Alam., D. A. Yousri., M. B. Eteiba., (2015), “Flower pollination algorithm based solar PV parameter estimation”, Energy Conversion and Management, 101, 410-420.

[4] L. C. Cagnina., S. C. Esquivel., and C. A. Coello, (2008), “Solving engineering optimization problems with the simple constrained particle swarm optimizer”, Informatica, 32, 319-326.

[5] H. Chiroma., N. L. M. Shuib., S. A. Muaz., A. I. Abubakar., L. B. Ila., J. Z. Maitama., (2015), “A review of the application of bio-inspired flower pollination algorithm”, Procedia Computer Science, 62, 435-441.

[6] M. Dorigo., V. Maniezzo., A. Colorni., (1996), “The ant system: optimization by a colony of cooperating agents”, IEEE Trans. Syst. Man Cybern, B 26, 29-41.

[7] L. Djerou., N. Khelil., S. Aichouche., (2017), “Artificial Bee Colony Algorithm for Solving Initial Value Problems”. Communications in Mathematics and Applications Published by RGN Publications, 8 (2), 119-125.

[8] B. J. Glover., (2007), “Understanding Flowers and Flowering: An Integrated Approach”. Oxford University Press.

[9] D. E. Goldberg., (1989), “Genetic Algorithms in Search. Optimization and Machine Learning. Addison Wesley”, Boston.

[10] J. H. Holland., (1975), “Adaptation in Natural and Artificial Systems”. University of Michigan Press, Ann Arbor.

[11] X. He., X. S. Yang., M. Karamanoglu., Y. Zhao., (2017), “ Flower pollination algorithm: a discrete-time Markov chain approach”, Procedia Computer Science, Elsevier, 108, 1354-1363.

[12] P. Henrici., (1964), “Elements of Numerical Analysis”, Mc Graw-Hill, New York.

[13] J. Kennedy., R. C. Eberhart., (1995), “Particle swarm optimization”, In: Proceedings of IEEE International Conference on Neural Networks No. IV. 27 Nov-1 Dec, pp. 1942--1948, Perth Australia.

[14] B. Mahdad., and K. Srairi., (2016), “Security constrained optimal power flow solution using new adaptive partitioning flower pollination algorithm”, Applied Soft Computing, 46, 501-522.

[15] S. Nakrani., C. Tovey., (2004), “On honey bees and dynamic allocation in an internet server colony”. Adapt. Behav. 12 (3--4). 223-240.

[16] I. Pavlyukevich., (2007), “Levy flights, non-local search and simulated annealing”, J. Computational Physics, 226, 1830-1844..

[17] D. Rodrigues., G. F. A. Silva., J. P. Papa., A. N. Marana., X. S. Yang., (2016), “EEG-based person identification through binary flower pollination algorithm”, Expert Systems with Applications, 62 (1), 81-90.

[18] S. A. Sayed., E. Nabil., A. Badr., (2016), “A binary clonal flower pollination algorithm for feature selection”, Pattern Recognition Letters, 77 (1), 21-27.

[19] R. Salgotra., and U. Singh., (2017), “Application of mutation operators to flower pollination algorithm”, Expert Systems with Applications, 79 (1), 112-129.

[20] S. Velamuri., S. Sreejith., P. Ponnambalam., (2016), “Static economic dispatch incorporating wind farm using flower pollination algorithm”, Perspectives in Science, 8, 260-262.

[21] N. M. Waser., (1986), “Flower constancy: definition, cause and measurement”, The American Naturalist, 127 (5), 596-603.

[22] P. Willmer., (2011), “Pollination and Floral Ecology”, Princeton University Press.

[23] X. S. Yang., (2104), “Book Nature Inspired Optimization Algorithm”, Elsevier.

[24] X. S. Yang., (2013), “Flower Pollination Algorithm for Global Optimization”, arXiv: 1312. 5673v1 [math. OC] 19 Dec.

[25] X. S. Yang., A. H. Gandomi., (2012), “Bat algorithm: a novel approach for global engineering optimization” Eng. Comput. 29 (5), 464-483.

[26] X. S. Yang., (2008), “Nature-Inspired Metaheuristic Algorithms”, Luniver Press.

[27] Y. Q. Zhou., R. Wang., Q. F. Luo., (2016), “Elite opposition-based flower pollination algorithm”, Neurocomputing, 188, 294-310.