Flower Pollination Algorithm with Linearly Varying Exploration: A Novel Approach for Continuous Function Optimization Problem

Mohammad Shafiul Alam
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Sanonda Datta Gupta
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Samiha Sara Prima
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Nur-E-Asma Tabassum
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

ABSTRACT

The Flower Pollination Algorithm (FPA) a recently introduced bio–inspired algorithm based on the natural process of pollination of flowers. FPA has quickly drawn the attention of the research community due to its nature of both exploitation and exploration. In this paper we have introduced a novel variation in its mutation operation by linearly varying the value of its proximate probability parameter. We have evaluated the proposed algorithm on a number of benchmark problems and the experimental results are compared with the basic FPA. The empirical study indicates that the proposed FPA with variable exploration rate has better optimization performance than the basic FPA algorithm on most of the benchmark functions.

Keywords

Flower pollination algorithm; exploration; exploitation; proximate probability.

1. INTRODUCTION

To find out the optimum solution of different optimization problems, nature inspired algorithms have been applied sincelast few decades. The algorithms which are inspired from biological system such as pollination system, reproduction system, swarm behavior based system and many more are being used to minimize and maximize many optimization problems through exploitation and exploration. Some examples such systems are — Artificial Bee Colony (ABC) which was based on the intelligent foraging behavior of honey bee swarm [1], Cuckoo Search Algorithm (CSA) that is inspired by the obligate brood parasitism of some cuckoo species [2] and BAT algorithm based on the echolocation behavior of micro bats [3]. All these algorithms have produced better performance on wide range of applications such as mathematical problems, e.g., image processing, Data mining, Clustering, Single Integration, finding Eigenvalue and Eigenvector, Set covering problems, and so on [3]–[9].

Flower Pollination Algorithm (FPA) is a recently introduced bio inspired algorithm that is popular for its outstanding performance on both global and local search space [4]. FPA has become one of the most popular bio-inspired algorithms which has been used to solve many challenging optimization problems. It is observed in nature on the pollination system of flowering plants. Due to its nature of exploitation and exploration, it has been a popular algorithm and has opened a new research era in optimization problems. In comparison to other swarm intelligence algorithms, FPA got the attention from last few years due to its excellent performance with nonlinearity and multimodality of the objective functions. The main advantage of FPA is that it explores and exploits the search space to locate the neighborhood of the globally optimum solution.

In this paper we have proposed a new algorithm based on the core idea of flower pollination algorithm. In the following section II, We have briefly described the FPA algorithm, followed by some existing variants of FPA in section III. Then we have utilized one of its parameters to establish our new algorithm in section IV. We have evaluated the new algorithm using a set of benchmark test functions, which has been used widely in the literature [10]-[12] in the following section V. After that, we will analyzed the result and drawn a few conclusions from the experiments.

2. FLOWER POLLINATION ALGORITHM (FPA)

The Flower pollination system shows remarkable adaptive and co-ordination skill. Flowering plants have been found very influential in the theory of evolution. The main purpose of flower pollination is reproduction of plants. Flower pollination generally occurs with the transfer of pollens from one flower to others. Insects, birds, air etc mediums are responsible for the transference of pollens. Such mediums are known as pollinators. Even some flowers and pollinators have special relationship which is known as flower-pollinator partnership. For example, some insects get attracted to some special flowers and thus those pollinators cause pollination of that specific flower species.

Pollination process can be divided into two different categories — self-pollination and cross pollination. Cross pollination occurs from pollen of a flower of a different plant, while self-pollination is the fertilization within one flower. Again, pollination process can be divided into two different process based on the pollinators — Biotic pollination and Abiotic pollination. Biotic pollination is basically cross-pollination occurring over long distance, pollens are being transferred by insects, birds, bees etc. This type of pollination is known as global pollination. This long distance can be simulated by using the Lévyfight behavior. In contrast, the local pollination process that is abiotic with the flower constancy can be simulated using a different mutation formula, as shown in Figure 1. The overall process of flower pollination can be summarized into the following four basic steps [4].

Step-1: Biotic and cross-pollination is considered as the global pollination process with pollen carrying pollinators performing Lévyflights.

Step-2: Abiotic and self-pollination are considered as local pollination.

Step-3: Flower constancy can be considered by making the reproduction probability to be proportional to the similarity of two flowers involved.

Step-4: Local pollination and global pollination is controlled by a switch probability $p\in$ [0, 1].Due to the physical proximity and other factors such as wind, local pollination can have a significant fraction *p* in the overall pollination activities.

From the above discussion we can say that the flower pollination algorithm is based on two key steps: global pollination and local pollination. In global pollination, pollens transfer to long distance by insects, birds and bees. Thus through this process there is a chance to get the reproduction of the fittest **g*** [best solution], as shown in Figure 1.

```
Objective min or max $f(x), x = (x1, x2, ..., xd)$
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 )
    fori = 1 : n ( all n flowers in the population)
        if rand < p
            Draw a ( d – dimensional ) step vector L which obeys a Lévy distribution
            Global Pollination via
                $x_i^{t+1} = x_i^t + L(g^* - \mathrm{x}_\mathrm{i}^\mathrm{t})$
        else
            Draw $\mathcal{E}$ from a uniform distribution in [0, 1]
            Randomly choose j and k among all the solutions
            Do local pollination via
                $x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t)$
        end if
        Evaluate new solutions.
        If new solutions are better, update them in the population
    end for
```

Figure 1: Pseudo-code of Flower Pollination Algorithm by Xin-She Yang [4]

Maximum flower pollination procedures perform both local and global pollination. In step 4 the switch probability or proximate probability *p*=0.8 explores the search space and works better in many existing problems [4].

3. EXISTING VARIANTS AND APPLICATION OF FPA

FPA has been extended to solve many high dimensional and multi-objective problems. For example, the multi-objective FPA (MOFPA) algorithm adds the weighted sum to integrate multi objects into a single one [5]. To improve the speed of annealing, combining with the flower pollination algorithm, an improved simulated annealing algorithm (FPSA) has been proposed in [6]. Another hybrid algorithm, the hybrid FPA with particle swarm optimization (FP-PSO) is used to solve many constrained optimization problems [7]. The FPA with the chaotic Harmony Search (FP-CHS) has improved the search performance and is used to solve Sudoku problems [8]. FPA has opened a new era in research field due to its exploitation and exploration nature. It produces better results in many unimodal and multimodal functions. The proposed variation — FPA-LVE produces a range of results where the search space is exploited and explored by varying the proximate probability *p* linearly..

4. PROPOSED ALGORITHM: FPAWITHLINEARLY VARYING EXPLORATION (FPA-LVE)

The performance of FPA significantly depends on the parameter $p \in [0,1]$, which is responsible for local vs. global pollination. As mentioned earlier in the algorithm of FPA, the global pollination cause large steps to be taken in the search space with the Lévy distribution. Exploitation or exploration depends on the values of the proximate probability *p*.

FPA follows two types of mutation process —exploration and exploitation. With the Lévy fight behavior, it explores the search spaces and this is controlled by the parameter *L* which is known as the strength of the pollination, and has been addressed as the step size in [4]. Through the larger step sizes it explores the search space and tries to find out the neighborhood of the fittest value g*. The value is *L* should be in [0, 1], as suggested in [4].

In every swarm intelligence based search and optimization algorithm, it is important to explore the search space in order to escape the locally minimal points. As the global optimum is unknown, the proposed algorithm FPA-LVE performs both exploitation and exploration in the search space by varying the proximate probability *p* linearly. Small values of *p* decrease explorations (i.e., increase exploitations) by reducing the probability of larger Lévy jumps (Fig. 1), while larger values of *p* increase explorations by inducing more Lévy jumps. However, some degree of control is necessary to limit the values of *p* within some range in order to avoid complete domination by exploitation or exploration. This is why we have restricted extreme values of *p* (e.g., 0 or 1) by always limiting *p* within [0.2, 0.8].

In our experiments, we have initialized the value of the proximate probability *p* to 0.5 and employed it on the benchmark functions to collect the result of the basic FPA algorithm. Then we varied the value of *p* by increasing and decreasing *p* linearly to exploit and explore the search space and collected the result of FPA-LVE. Actually we have implemented two different variants of FPA-LVE, which are — FPA with Linearly Decreasing Exploration with proximate probability p less than 0.5 (FPA-LDE) and FPA with Linearly Increasing Exploration with proximate probability p greater than 0.5 (FPA-LIE).The minimum value (0.2) of p exploits the search area and gives good result especially in some of the unimodal functions. Gradually, we varied the value of p linearly, and collect results for each value of p, until we reach the value of 0.8 for which we get good values for some of the functions. However, the best performance of FPA-LVE is found with $p = 0.6$, followed by $p = 0.5$, which indicates that somewhat 'balance' between explorations and exploitations is overall most suitable for most of the functions. With the variation of the proximate probability value p, the degree of exploration of FPA is varied, and performance is varied significantly to produce good result for many unimodal and multimodal functions, which indicates that FPA with high explorative characteristics may perform better than the basic FPA in many situations. This experiment also reveals the exploitative and explorative requirements of the different unimodal and multimodal functions.

5. EXPERIMENTAL STUDIES

The experiment is conducted on a standard benchmark function set consisting of 15 functions of varying complexity. TABLE I presents a brief overview on the functions, their characteristics, dimensions and global minimum values.

Both FPA and the proposed FPA-LVE are executed using the same parameter values — iteration number $N = 5000$, the population size = 20, the dimensionality $D = 30$, initial $p = 0.2$. FPA-LVE gradually increases p up to 0.8. The initial population is generated randomly within the search space, as shown in Table I.

Unimodal Functions:

Among the 15 benchmark functions the first eight functions f_1-f_8 are unimodal functions where f_1, f_2, f_3 are unimodal separable functions and f_4–f_8 are unimodal non-separable functions. Among these eight unimodal functions f_1 and f_8 functions obtain their optimum values in Basic FPA and FPA–LIE, which is shown in TABLE II and Figure 2. Both the algorithms perform similarly on the Rosenbrock Function (f_4). It is difficult to find its global minimum due to its long parabolic slope and flat valley. Function f_6 (BentCigar) produces better result with proximate probability $p = 0.5$ due to its smooth and narrow ridge. The function f_7 (Discuss) also gives comparatively better performance with Basic FPA due to its one sensitive direction. On the otherhand f_2 (Quartic) and f_5 (Schwefel 1.2) functions perform better in FPA–LIE. The Quartic function f_2 is highly explorative as it has the same infinite limit when the argument goes to the positive or negative infinite. So to find out its global optimum, it requires high degree of exploration, and FPA-LIE performs best on this function because it employs linearly increasing exploration during its execution.

Multimodal Functions:

Multimodal functions (both separable and non-separable) are often considered the most challenging to optimize. From Table I, f_9–f_{15} are the multimodal functions among them f_{11} and f_{15} are multimodal separable functions and the rest are multimodal non-separable functions. Multimodal functions have exponentially many locally minimal points and hence require more explorations to reach to the global optimum solution. Table II shows that f_9, f_{11}, f_{13} are moderate exploration demanding functions. On the other hand f_{10} performs almost similar in Basic FPA, FPA-LDE and FPA-LIE. The function f_{15} (Schwefel 2.26) is highly exploitative, and hence the proximate probability 0.2 produces best performance [12]. For this function, the number of local optima is huge and the second best local optimum is far from the global optimum, so it produces better result at $p = 0.6$ and then reached the final result by fine tuning using $p = 0.2$, which is quite far from the previous value. As the number of locally minimal points is huge, it requires high exploration first, followed by exploitation to pinpoint the global minimum. We can summarize our overall observation using the following points.

1. Through varying the proximate probability p, different types of behaviors have been observed, such as Exploitation demanding, Exploration demanding, Exploitation and Exploration independent and moderate Explorative and Exploitative functions.
2. FPA with highest proximate probability value (0.7-0.8) helps to escape strong local minima and achieve globally optimum solution for explorative demanding functions.
3. FPA with lowest proximate probability value (0.2-0.3) helps exploitation demanding functions to obtain their global minimum where the number of locally minimal points is large.

Table 1: Standard Benchmark Functions where *D*: Dimensionality, *S*: Search Space, *U*: Unimodal, *M*: Multimodal, *S*: Separable, *NS*: Non- Separable, $f_{\min}$: Global Minimum.

No	Function	*C*	*D*	*S*	f_{min}
f_1	Sphere	*US*	30	$[-100, 100]^D$	0
f_2	Quartic	*US*	30	$[-1.28, 1.28]^D$	0
f_3	Step	*US*	30	$[-100, 100]^D$	0
f_4	Rosenbrock	*UN*	30	$[-30, 30]^D$	0

f_{5}	High Conditioned Elliptic Function	*UN*	30	$[-100, 100]^{D}$	0
f_{6}	Bent Cigar	*UN*	30	$[-100, 100]^{D}$	0
f_{7}	Discus	*UN*	30	$[-100, 100]^{D}$	0
f_{8}	Schwefel 1.2	*UN*	30	$[-100,100]^{D}$	0
f_{9}	Ackley	*MN*	30	$[-32, 32]^{D}$	0
f_{10}	Griewank	*MN*	30	$[-600, 600]^{D}$	0
f_{11}	Rastrigin	*MS*	30	$[-5.12, 5.12]^{D}$	0
f_{12}	Katsura	*MN*	30	$[-100, 100]^{D}$	0
f_{13}	Happy Cat	*MN*	30	$[-100,100]^{D}$	0
f_{14}	HGBAT	*MN*	30	$[-100, 100]^{D}$	0
f_{15}	Schwefel 2.26	*MS*	30	$[-500, 500]^{D}$	-12569.5

Table 2: Performance Of Flower Pollination Algorithm With Linearly Varying Exploration Compared To The Basic FPA On Unimodal and Multimodal Functions. Minimum Value of Eeach Function is Shown After 5000 Iterations. Performance which are almost 90% same are reffered as "similar".

FPA with Linearly Decreasing Exploration (FPA-LDE) using *p*< 0.5				**Basic FPA (*p* = 0.5)**	**FPA with Linearly Increasing Exploration (FPA-LIE) using *p*> 0.5**			**Best Performance**
No	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	
f_{1}	1.32e-06	3.22e-07	2.11e-08	**4.92e-11**	**6.55e-11**	1.12e-07	1.01e-06	**Both Basic FPA And FPA – LIE**
f_{2}	0.02461	0.00904	0.00886	0.01256	0.010216	0.00876	**0.006254**	**FPA-LIE**
f_{4}	-999997	-999994	-999997	-999984	-999999	-999981	-9999	Similar
f_{5}	0.10406	0.00220	0.00012	2.31e-05	**3.37e-06**	3.99e-06	0.0029	**FPA-LIE**
f_{6}	109.836	21.7285	0.07651	**0.03215**	1.4291	0.46097	49.245	Basic FPA
f_{7}	0.00465	0.00068	0.00019	**6.26e-07**	1.46e-06	8.30e-06	0.0004	Basic FPA
f_{8}	2.84e-04	6.72e-07	2.85e-07	**4.55e-09**	**1.03e-09**	1.66e-08	4.47e-06	**Both Basic FPA and FPA-LIE**
f_{9}	6.9438	2.4433	0.075547	1.8261	**0.019946**	0.24634	0.037415	**FPA-LIE**
f_{10}	-3.752	-3.6152	-3.4397	-3.6136	-3.7624	-3.6779	-3.6734	Similar
f_{11}	33.6237	43.3227	35.8532	40.1899	**14.8581**	30.3175	43.4354	**FPA-LIE**
f_{13}	0.3449	0.20351	0.32162	0.286	**0.18914**	0.39618	0.47191	**FPA-LIE**
f_{14}	0.21587	0.13487	0.34076	**0.19655**	0.22971	0.20189	0.88374	Basic FPA
f_{15}	**-3347.01**	-3171.8	-3253.11	-3246.61	-3311.73	-3235.63	-3179.12	**FPA-LDE**

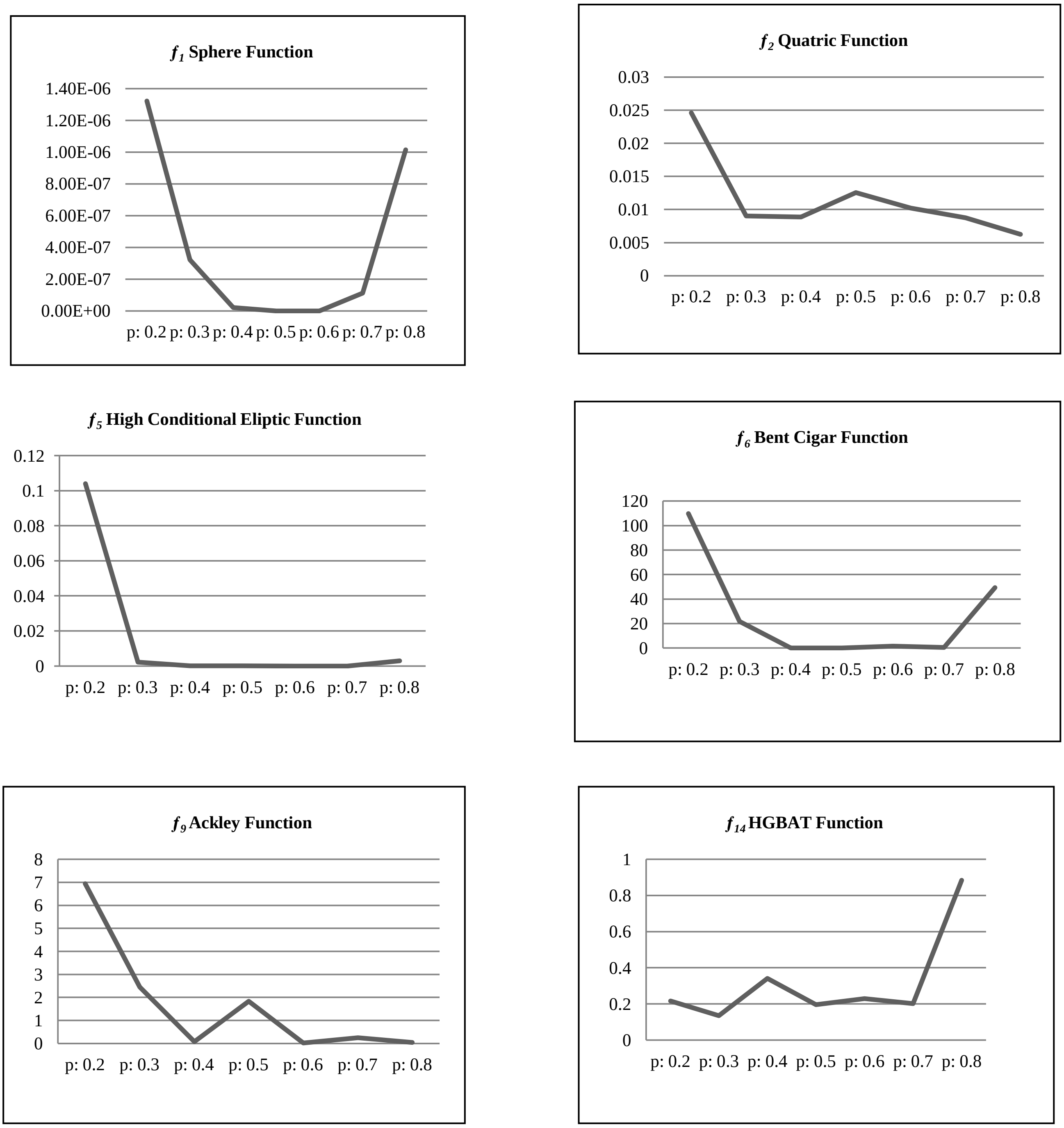

Figure 2: FPA–LVE on Unimodal and Multimodal Separable and Non Separable functions (f_1,f_2 .f_5,f_6, f_9,f_{14}). The vertical axis represents the minimum values and horizontal axis shows the proximate probability *p* [0.2–0.8]. Results are shown over 5000 iterations, with function dimensionality = 30.

6. CONCLUSION

This paper introduces a new modified variant of the basic FPA algorithm, which is varying one of its parameter linearly to affect the degree of exploitation and exploration. This algorithm is evaluated on a benchmark suite consisting of 15 functions, including both unimodal and multimodal, separable and non-separable functions. This overall comparison shows that FPA, with varying its exploitation and exploration characteristics, can perform better on many highly exploration and exploitation demanding functions. By varying the proximate probability *p* from 0.2 to 0.8, different behavior patterns of the functions have been observed. There might have many scope to improve this algorithm as it is initially a simple strategy based on the FPA algorithm. Some of the functions like the Rosenbrock, Griewank and Schwefel 2.26 show flat and semi-flat optimization behavior during the runtime, indicating that the algorithm is getting stuck around the locally optimal points failing to explore the search space sufficiently. A simple hill climbing phase can be added after high explorations to exploit the neighborhood of the recently found optimal points to help FPA-LVE to further improve the final solution quality.

7. REFERENCES

[1] Y. Xu, P. Fan and L. Yuan, "A Simple and Efficient Artificial Bee Colony Algorithm", *Mathematical Problems in Engineering*, vol. 2013, pp. 1-9, 2013.

[2] X. Yang and S. Deb, "Cuckoo search: recent advances and applications", *Neural Computing and Applications*, vol. 24, no. 1, pp. 169-174, 2013.

[3] X. Yang and X. He, "Bat algorithm: literature review and applications", *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, p. 141, 2013.

[4] X. Yang, "Flower Pollination Algorithm for Global Optimization", *Unconventional Computation and Natural Computation*, pp. 240-249, 2012.

[5] X. Yang, M. Karamanoglu and X. He, "Multi-objective Flower Algorithm for Optimization", *Procedia Computer Science*, vol. 18, pp. 861-868, 2013.

[6] M. Abdel-Baset and I. Hezam, "A Hybrid Flower Pollination Algorithm for Engineering Optimization Problems", *International Journal of Computer Applications*, vol. 140, no. 12, pp. 10-23, 2016.

[7] O. Raouf, M. Baset and I. henawy, "A New Hybrid Flower Pollination Algorithm for Solving Constrained Global Optimization Problems", 2014.

[8] O. Abdel Raouf, I. El henawy and M. Abdel Baset, "A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles", *International Journal of Modern Education and Computer Science*, vol. 6, no. 3, pp. 38-44, 2014.

[9] B. Nozohour-leilabady and B. Fazelabdolabadi, "On the application of artificial bee colony (ABC) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (PSO) methodology", *Petroleum*, vol. 2, no. 1, pp. 79-89, 2016.

[10] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization", Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report #2005005, IIT Kanpur, India.

[11] B. Liu, Q. Chen and Q. Zhang, J. J. Liang, P. N. Suganthan, B. Y. Qu, "Problem Definitions and Evaluation Criteria for Computationally Expensive Single Objective Numerical Optimization", Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, December 2013.