Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/050104

A Hybrid Flower Pollination Algorithm with Tabu Search for Unconstrained Optimization Problems

Ibrahim M. Hezam[1,*], *Mohamed Abdel-Baset*[2] *and Bayoumi M. Hassan*[3]

[1] Department of computer, Faculty of Education, Ibb University, Ibb city, Yemen.
[2] Operations Research Department, Faculty of Computers &Informatics, Zagazig University, Sharqiyah, Egypt.
[3] Decision Support Department, Faculty of Computers & Information, Cairo University, Cairo, Egypt.

Received: 4 Apr. 2015, Revised: 22 Sep. 2015, Accepted: 24 Sep. 2015.
Published online: 1 Jan. 2016.

Abstract: Flower pollination algorithm (FPA) is a novel metaheuristic optimization algorithm with quick convergence, but its population diversity and convergence precision can be limited in some problems. In order to enhance its exploitation and exploration abilities, in this paper a novel hybrid flower pollination algorithm with Tabu Search (TS-FPA) has been applied to unconstrained optimization problems. TS-FPA is validated by ten benchmark functions. The results show that the proposed algorithm is able to obtained accurate solution, and it also has a fast convergence speed and a high degree of stability.

Keywords: Flower Pollination Algorithm; Tabu Search; Unconstrained Optimization; Metaheuristic.

1 Introduction

Optimization is a field of applied mathematics that deals with finding the extremal values of a function in a domain of definition, subject to various constraints on the variable values [1].Global optimization refers to finding the extreme value of a given nonconvex function in a certain feasible region and such problems are classified in two classes; unconstrained and constrained problems. Solving global optimization problems has made great gain from the interest in the interface between computer science and operations research [1-5].

The increasing complexities of life leads to the complex optimization problems; therefore continuous development and improvement to optimization algorithms is demanded in order to confront and resolve these problems. There are two general approaches to solve optimization problems, namely, mathematical programming and metaheuristic methods [1]. Mathematical programming as deterministic methods use gradient information to search the solution space near an initial starting point. This is a gradient-based methods where higher accuracy investigation fulfill local search task. However, the variables and cost function of the generators need to be continuous. Moreover, a good starting point is vital for these methods to be executed successfully. High computational effort is another drawback of deterministic gradient methods especially at high-dimensional search space. In many optimization problems, prohibited zones, side limits, and non-smooth or non-convex cost functions need to be considered. As a result, these non-convex optimization problems cannot be solved using traditional mathematical programming methods.

On the other hand, heuristic and metaheuristic methods relies on stochastic algorithms to generate different trade of solutions instead of gradients. However the obtained solution value almost converge to same optimal solution with slight differences [2]. Difference between heuristic and metaheuristic is small loosely speaking [3] heuristic means ‘to find’ or ‘to discover by trial and error’. In metaheuristic algorithms, meta- means ‘beyond’ or ‘higher level’, and they generally perform better than simple heuristics. All metaheuristic algorithms use certain tradeoff of local search and global exploration.

Two important characteristics of metaheuristic are intensification and diversification [4]. Intensification intends to search around the current best solutions and select the best candidates or solutions. Diversification makes sure that the algorithm can explore the search space more efficiently, often by randomization. [5] Presents brief review of nature-inspired metaheuristic algorithms, where all existing algorithms are divided into four major categories: swarm intelligence (SI) based, bio-inspired (but not SI-based), physics/chemistry-based, and other algorithms. The swarm intelligence (SI) drawing inspiration

*Corresponding author e-mail: ibrahizam.math@gmail.com

from swarm-intelligence systems in nature. Depends on the collective behavior of decentralized, self-organized systems, natural or artificial. For example, Particle Swarm Algorithm (PSO) [6] Ant colony optimization (ACO) [7] , Bat algorithm (BA) [8], Cuckoo Search (CS) [8], Firefly Algorithm (FA) [8], Krill Herd (KH) [9] etc. Bio-inspired, but not SI do not use directly the swarming behavior. For example, Genetic Algorithms (GA), Flower Algorithm (FA) [8], Differential Evolution (DE) [10], Human-Inspired Algorithm [11].

Where the third category Physics and Chemistry inspiration often come from physics and chemistry sciences. For example, harmony search [12], intelligent water drop [13], simulated annealing [14], stochastic diffusion search [15]. And the last category is other algorithms inspiration away from nature. Consequently, some algorithms are not bio-inspired or physics/chemistry-based, for example Differential search algorithm [16], grammatical evolution [17], social emotional optimization [18]. In [19] developed optimization algorithm based on bacterial chemo taxis, where the way in which bacteria react to chemo attractants in concentration gradients plays an important role in reaching the global optimal solution. Recently a modified metaheuristic optimization techniques are introduced by [20-28] in ordered to improve the convergence speed and accuracy.

In this paper we propose a novel hybrid algorithm named TS-FPA for solving unconstrained optimization problems. The motivation for a new hybrid algorithm is to overcome the drawback of classical Tabu search which is not suitable for continuous optimizations. In this hybrid algorithm achieves a balance between the diversification and the intensification by incorporating the ideas of FPA into a TS algorithm. TS-FPA is validated by ten benchmark functions. The results show that the proposed algorithm is able to obtained accurate solution, and it also has a fast convergence speed and a high calculation precision.

The rest of this paper is organized as follows. In Section 2, a standard flower pollination algorithm is introduced. In Section 3, we present the standard Tabu Search. In Section 4, a description of the proposed algorithm is given. The numerical results of proposed algorithm are reported in Section 5. Finally, conclusion and future works are presented in Section 6.

2 The Flower pollination Algorithm

Flower Pollination Algorithm (FP) was founded by Yang in the year 2012. Inspired by the flow pollination process of flowering plants are the following rules [25]:

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 $p \in [0,1]$, 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 [39].Therefore, Rule 1 and flower constancy can be represented mathematically as:

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

Where x_i^t is the pollen i or solution vector x_i 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, $L(\lambda)$ 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:

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

Here, $\Gamma(\lambda)$ 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 3 can be represented as

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t) \tag{3}$$

Where x_j^t and x_k^t are pollen from different flowers of the same plant species. This essentially imitates 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 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 = 0.5 as an initially value. A preliminary parametric showed that p = 0.8 might work better for most applications [25]. The basic steps of FPA can be summarized as the pseudo-code shown in Figure 1.

Algorithm 1: Flower pollination algorithm
Define Objective function $f(x)$, $x = (x_1, x_2, ..., x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution B in the initial population
Define a switch probability $p \in [0, 1]$
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

Fig. 1 Pseudo code of Flower pollination algorithm

3 Tabu Search

Tabu Search (TS) was formalized in 1986 by Glover [29]. TS was designed to manage an embedded local search algorithm. It explicitly uses the history of the search, both to escape from local minima and to implement an explorative strategy. Its main characteristic is indeed based on the use of mechanisms inspired by the human memory. The basic steps of TS can be summarized as the pseudo-code shown in Figure 2.

Algorithm 2: Tabu Search
$S_0 \leftarrow$ *an initial solution;*
$S_{best} \leftarrow S_0$ *and* $S \leftarrow S_0$;
$i \leftarrow 0$;
while (t <MaxPerturbation) Do
$S' \leftarrow$ *the best solution found by local_s*
If $S' \leftarrow$ *is better than* S_{best} *then*
$S_{best} \leftarrow S'$ *and* $i \leftarrow 0$;
else
$i \leftarrow i+1$;
end if
$S \leftarrow perturb(S')$;
End while
Return S_{best}

Fig. 2 Pseudo code of Tabu search

4 The Proposed Algorithm (TS-FPA)

The proposed TS-FPA algorithm is collaborative combinations of the TS and FPA techniques. In this hybrid algorithm achieves a balance between the diversification and the intensification by incorporating the ideas of FPA into a TS algorithm. The steps of the proposed algorithm as follows:

Algorithm 3: Proposed Algorithm
Step 1 *Define Objective function* $f(x)$, $x = (x_1, x_2, ..., x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution B in the initial population
Define a switch probability $p \in [0, 1]$
Initialize the tabu lists and the values of parameters such
Step 2 *(Generation of an initial solution). Randomly*
Step 3 *(Tabu search-based local search procedure). Search the neighborhood based TS, and store a set of local minimum solutions. If the current Tabu Solution is better than* S_{best}*, go to Step 4. Otherwise, return to Step 2.*
Step 4 *(Flower pollination algorithm -based diversification procedure).*
Expand the exploration area based on FPA to increase the diversity of the solutions.
Step 5 *(Terminal condition). If the current computational time is greater than Time limit, terminate the algorithm. Otherwise, return to Step 2.*

Fig. 3 Pseudo code of proposed algorithm (TS-FPA)

5 Experimental Results and Analysis

In this section, we will investigate the performance of the proposed algorithm. To evaluate the performance of our proposed algorithm, we applied it to 10 standard benchmark functions [30]. These functions have been widely used in the literature. The functions are unimodal and multimodal test functions. These functions have been widely used as benchmarks for research with different methods by many researchers. All programs are written in MATLAB code, and the experiments were performed on a Windows 7 Ultimate 64-bit operating system; processor Intel Core i7 760 running at 2.81 GHz; 6 GB of RAM. The proposed algorithm compared with TS and FPA [10], using the mean and standard deviation to compare their optimal performance. For all test functions, the algorithms carry out 50 independent runs .the parameters in the proposed algorithm are chosen as follows: n= 50 ; the number of iterations is set to t = 1000 ; the value of tabu life-span obtained by observing a number of experimental values and choosing the value for which the solution reaches faster).

Definitions of benchmark problems are described as follows:

1. The first is Sphere function, defined as:

$$mn\ f_1 = \sum_{i=1}^{N} x_i^2$$

Where global optimum x*= (0,0,…,0) and f(x*) = 0 for $-100 \le x_i \le 100$.

2. The second is Rosenbrock function, defined as:

$$min\ f_2 = \sum_{i=1}^{N-1}(100(x_{i+1} - x_i^i)^2 + (x_i - 1)^2)$$

Where global optimum x*= (1,1,…,1) and f(x*) = 0 for $-100 \le x_i \le 100$.

3. The third is generalized Rastrigrin function, defined as:

$$min\ f_3 = \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i) + 10)$$

Where global optimum x*= (0,0,…,0) and f(x*) = 0 for $-10 \le x_i \le 10$.

4. The fourth function is as follows:

$$min\ f_4 = \sum_{i=1}^{N} z_i^2 - 450$$

Where Z=x-o; $o=[o_1,o_2,\ldots,o_n]$ is the shifted global optimum; global optimum x*= o and f(x*) = -450 for $-100 \le x_i \le 100$.

5. The fifth is generalized Griewank function, defined as:

$$mn\ f_5 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Where global optimum x*= (0,0,…,0) and f(x*) = 0 for $-600 \le x_i \le 600$.

6. The sixth is Schwefel's function, defined as:

$$min\ f_6 = \sum_{i=1}^{N}|x_i| + \prod_{i=1}^{N}|x_i|$$

Where global optimum x*= (0,0,…,0) and f(x*) = 0 for $-100 \le x_i \le 100$.

7. The seventh is rotated hyper-ellipsoid function, defined as:

$$min\ f_7 = \sum_{i=1}^{N}\left(\sum_{j=1}^{i} x_j\right)^2$$

Where global optimum x*= (0,0,…,0) and f(x*) = 0 for $-100 \le x_i \le 100$.

8. The eighth is Ackley's function, defined as**:**

min f$_8$=

$$20 + e - 20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}}\right) - \exp\left(\sqrt{\frac{\sum_{i=1}^{N} x_i^2\cos(2\pi x_i)}{N}}\right)$$

Where global optimum x*= (0,0,…,0) and f(x*) = 0 for $-32 \le x_i \le 32$.

9. The ninth is Schwefel's function, defined as:

$$min\ f_9 = 418.9829N - \sum_{i-1}^{N}(x_i \sin(\sqrt{|x_i|}))$$

Where global optimum x*= (420.9687, 420.9687,…, 420.9687) and f(x*) = 0 for $-500 \le x_i \le 500$.

10. The tenth function defined as:

$$min\ f_{10} = \sum_{i=1}^{N}\left(\sum_{j=1}^{i} Z_j\right)^2 - 450$$

Where Z=x-o; $o=[o_1,o_2,\ldots,o_n]$ is the shifted global optimum; global optimum x*= o and f(x*) = -450 for $-100 \le x_i \le 100$.

The comparison of test results is shown in Table 1. The Best, Mean, Worst and Std. represent the optimal fitness value, mean fitness value, worst fitness value and standard deviation, respectively. The results have demonstrated the superiority of the proposed approach to finding the global optimal solution.

Fig. 4,5,6 and 7 show the mean best fitness over the number of generations for TS, FPA and TS-FPA. These figures clearly show that the search efficiency of TS-FPA is superior to that of TS, and FPA on the first four benchmark functions.

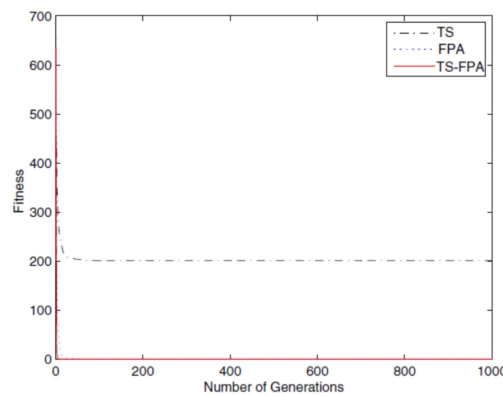

Fig. 4 Sphere function

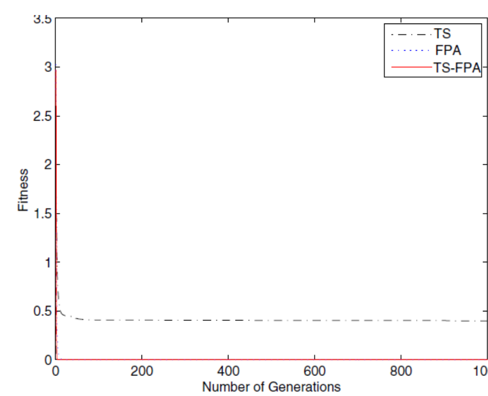

Fig. 5 Rosenbrock function

Table 1 the optimal solution results of proposed algorithm and other algorithms.

Benchmark Functions	Algorithm	Results			
		Best	Worst	Mean	Std.
f1	FPA	6.3643e-001	3.3831e-008	2.1536e-008	6.4556e-009
	TS	2.9129e-008	6.1045e-007	2.0999e-007	1.2421e-007
	TS-FPA	3.9296e-041	2.9471e-031	2.1333e-032	6.9650e-032
f2	FPA	6.5349e+000	9.4594e+003	1.5642e+003	2.8488e+003
	TS	1.6436e-002	4.9914e+003	2.3417e+002	9.0439e+002
	TS-FPA	5.3463e-004	5.4967e+003	4.9368e+002	1.3524e+003
f3	FPA	1.1661e-005	1.3006e-004	4.5934e_005	2.5863e-005
	TS	2.4508e-006	2.9849e+000	9.7833e-001	8.8010e-001
	TS-FPA	0	9.9496e-001	3.3165e-002	1.8165e-001
f4	FPA	-4.5000e+002	-4.5000e+002	-4.5000e+002	5.7304e-009
	TS	-4.5000e+002	-4.4993e+002	-4.4999e+002	1.6373e-002
	TS-FPA	-450	-450	-450	0
f5	FPA	9.0619e-003	2.0167e-001	7.6168e-002	4.6718e-002
	TS	5.4412e-002	2.6813e-001	1.4351e-001	5.7705e-002
	TS-FPA	0	1.2046e-001	6.8529e-002	3.3204e-002
f6	FPA	5.9582e-004	2.1651e-003	1.1033e-003	3.4149e-004
	TS	2.6277e-004	4.9911e-004	3.7745e-004	6.3608e-005
	TS-FPA	5.4770e-021	2.6652e-017	2.5681e-018	6.4570e-018
f7	FPA	1.2325e+001	6.8309e+002	2.4584e+002	1.9314e+002
	TS	2.9975e+001	3.3575e+002	1.1424e+002	7.7109e+001
	TS-FPA	1.0224e-001	9.4449e+000	2.5160e+000	2.3538e+000
f8	FPA	2.6202e-004	1.0237e-003	5.4888e-004	1.7181e-004
	TS	9.7298e-005	1.8063e-001	6.1802e-003	3.2949e-002
	TS-FPA	7.9936e-015	5.0626e-014	2.6468e-014	9.4872e-015
f9	FPA	1.2728e_004	1.2728e_004	1.2728e_004	1.0104e_009
	TS	1.2731e_004	9.8712e_003	6.5087e_004	2.0524e_003
	TS-FPA	1.2728e_004	1.1844e+002	3.9481e+000	2.1624e+001
f10	FPA	-4.4180e+002	-1.5849e+002	-3.3589e+002	7.3644e+001
	TS	-4.4870e+002	5.3709e+002	-1.9303e+002	2.3832e+002
	TS-FPA	-4.4983e+002	-4.1824e+002	-4.4334e+002	8.0172e+000

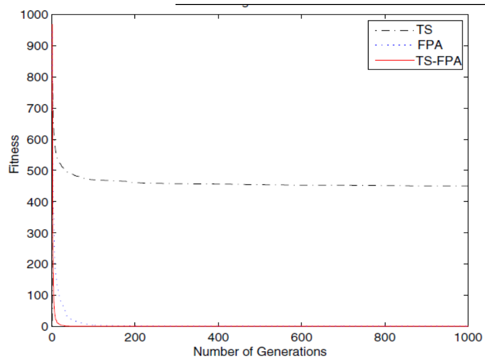

Fig. 6 Rastrigin Function

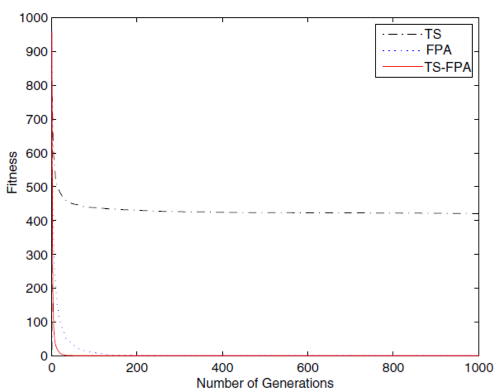

Fig.7 Griewark function

6 Conclusions and Future Works

In this paper, we propose a novel hybrid algorithm named TS-FPA, which integrate flower pollination algorithm (FPA) with Tabu search (TS) to solve unconstrained optimization problems. The proposed algorithm TS-FPA is tested on several benchmark problems from the usual literature and the numerical results have demonstrated the superiority of the proposed algorithm for finding the global optimal solution.

The future work will be focused on:

i. Using TS-FPA for solving constrained optimization problems;

ii. The proposed algorithm should be used to solve multi-objective optimization problems;

iii. Applied TS-FPA for solving combinatorial optimization problem for example (Traveling salesman problem, Graph Coloring problem, Knapsack problem and Quadratic assignment);

iv. Also we can use this algorithm for solving most popular engineering optimization problems.

Acknowledgment

We are grateful to an anonymous referee for the detailed comments on revising the original draft of the paper.

References

[1] Ravindran, A., Reklaitis G. V., and Ragsdell K. M., 2006. *Engineering optimization: Methods and applications*: Wiley. Com.

[2] Rao S. S. and Rao S., *Engineering optimization: theory and practice*: John Wiley & Sons, 2009.

[3] Liberti L. and Maculan N., 2006. *Global Optimization: Volume 84, From Theory to Implementation* vol. 84: Springer.

[4] Floudas C. A., 2000. *Deterministic global optimization: theory, methods and applications* vol. 37: Springer.

[5] Floudas C. A., Pardalos P. M., Adjiman C. S., Esposito W. R., Gumus Z. H., Harding S. T., *et al.*, 1999. *Handbook of test problems in local and global optimization* vol. 33: Kluwer Academic Publishers Dordrecht.

[6] Kennedy, J., Eberhart, R., others, 1995. Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks. Perth, Australia, pp. 1942–1948.

[7] Dorigo, M., 1992. Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy.

[8] Yang, X.-S., 2011. Review of meta-heuristics and generalized evolutionary walk algorithm. International Journal of Bio-Inspired Computation 3, 77–84.

[9] Gandomi, A.H., Alavi, A.H., 2012. Krill herd: a new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation 17, 4831–4845.

[10] Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11, 341–359.

[11] Zhang, L.M., Dahlmann, C., Zhang, Y., 2009. Human-inspired algorithms for continuous function optimization, in: Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference On. IEEE, pp. 318–321.

[12] Geem, Z.W., Kim, J.H., Loganathan, G., 2001. A new heuristic optimization algorithm: harmony search. Simulation 76, 60–68.

[13] Shah-Hosseini, H., 2007. Problem solving by intelligent water drops, in: Evolutionary Computation, 2007. CEC 2007. IEEE Congress On. IEEE, pp. 3226–3231.

[14] Kirkpatrick, S., 1984. Optimization by simulated annealing: Quantitative studies. Journal of statistical physics 34, 975–986.

[15] Bishop, J., 1989. Stochastic searching networks, in: Proc. 1st IEE Conf. on Artifical Neural Networks. pp. 329–331.

[16] Civicioglu, P., 2012. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Computers & Geosciences 46, 229–247.

[17] Ryan, C., Collins, J., Neill, M.O., 1998. Grammatical evolution: Evolving programs for an arbitrary language, in: Genetic Programming. Springer, pp. 83–96.

[18] Xu, Y., Cui, Z., Zeng, J., 2010. Social emotional optimization algorithm for nonlinear constrained optimization problems, in: Swarm, Evolutionary, and Memetic Computing. Springer, pp. 583–590.

[19] Muller, S.D., Marchetto, J., Airaghi, S., Kournoutsakos, P., 2002. Optimization based on bacterial chemotaxis. Evolutionary Computation, IEEE Transactions on 6, 16–29.

[20] Raouf O., I. El-henawy, and M. Abdel-Baset. 2014. A novel hybrid flower pollination algorithm with chaotic harmony search for solving Sudoku puzzles." International Journal of Modern Education and Computer Science vol. 3, pp. 38-44.

[21] Abdel-Raouf O., M. Abdel-Baset, and I. El-henawy, 2014. Improved Harmony Search with Chaos for Solving Linear Assignment Problems", International Journal of Intelligent Systems and Applications, vol. 6,pp. 55-61.

[22] Abdel-Raouf O., M. Abdel-Baset, and I. El-Henawy, 2014. An Improved Chaotic Bat Algorithm for Solving Integer Programming Problems, *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 6, p. 18.

[23] Abdel-Raouf O., M. Abdel-Baset, and I. El-henawy, 2014. An Improved Flower Pollination Algorithm with Chaos, *I.J. Education and Management Engineering*, vol. 2, pp. 1–8.

[24] Abdel-Raouf O., M. Abdel-Baset, and I. El-henawy, 2014. Chaotic firefly algorithm for solving definite integral, International Journal of Information Technology and Computer Science, vol. 6,pp. 19-24.

[25] I. M.Hezam and M. Abdel-Baset, 2015. An Improved Flower Pollination Algorithm for Ratios optimization Problems. Applied Mathematics & Information Sciences Letters An International Journal, vol. 3, No. 2, pp. 83–91.

[26] Abdel-Baset M. and I.M. Hezam, 2015. An Effective Hybrid Flower Pollination and Genetic Algorithm for Constrained Optimization Problems, *Advanced Engineering Technology and Application An International Journal*, vol. 4, pp. 27 – 27–34.

[27] Raouf O. A., M. A. Baset, I. M. Elhenawy, 2014. Improved harmony search algorithm with chaos for solving definite integral, International Journal of Operational Research, Vol. 21, No. 2, pp. 252-261, 2014.

[28] Hezam I. M., M. Abd-ElBaset and I. Selem, 2015. Cuckoo Search Algorithm for Stellar Population Analysis of Galaxies,International Journal of Information Technology and Computer Science, vol. 7, pp.29-33.

[29] Glover F., 1986. Future paths for integer programming and links to artificial intelligence, Computers and Operations Research 13, 533–549.

[30] Mallipeddi R., Mallipeddi S., Suganthan P.N., 2010. Ensemble strategies with adaptive evolutionary programming, Inf. Sci. 180 (9), 1571–1581.