International Journal of Control and Automation
Vol. 9, No. 12 (2016), pp.139-150
http//dx.doi.org/10.14257/ijca.2016.9.12.13

ABC_M: a Hybrid Algorithm ABC and BA

Afrooz Moradbeigy, Hassan Rezaei and Farhad Hamidi

University of Sistan and Baluchestan
afroozmoradbeigy@pgs.usb.ac.ir, hrezaei@cs.usb.ac.ir, f_hamidi@math.usb.ac.ir

Abstract

Optimization is ability of find the best solution in the existing situations. Optimization is used in design and maintenance of systems engineering, economic, social and even necessary to reduce costs and increase profits. The widespread importance of optimization problem has a lot of grown. There are many algorithms for optimization and they are trying to reduce the disadvantages of other methods and increase the ability of resolve the problem. This paper proposed an adaptive ABC and Bat algorithm. The idea of algorithm is improved speed of convergence and optimized search in search space for ABC algorithm with Bat algorithm. The proposed algorithm is compared with ABC and Bat algorithm on benchmark function and test shows ABC_M are improved obviously. Also can be known a complete local search is more important from global search.

Keyword: *ABC algorithm, Bat algorithm, optimization, hybrid ABC-BAT algorithm, ABC_M*

1. Introduction

Extension of optimization science is from interest to achieving the best situation but gain all of the best conditions are not possible. For this reason a satisfied answer is selected instead of global optimal. Several approaches for designing solutions with acceptable quality under acceptable time limit are proposed. The decision-making problems can be often declared as a constrained optimization problem with some decision variables that are closed by a set of constraints. Constraint optimization problem as:

- Combinatorial problems: decision variables are discrete
- Continuous problems: decision variables are continuous
- Mixed problems

Some of combinatorial optimization problems are easy to state but difficult to solve. The means of difficult to solve, computation time is not be in polynomial bounded. Two classes of algorithms are available for the solution of combinatorial optimization problems: Exact and heuristics algorithms. The first type is guaranteed to find the optimal solution but for most NP-hard problems the performance of exact algorithms is not satisfactory. Approximate algorithms called heuristic methods. The term heuristic is used for algorithms which find solutions among all possible ones, but they do not guarantee that the best will be found, therefore they may be considered as approximately and not accurate algorithms. These algorithms, usually find a solution close to the best one and they find it fast and easily. A disadvantage of heuristic methods is that they:

- A limited number of different solutions are generated
- They stop at poor quality local optima, which is the case for iterative improvement methods.

Metahuristics have been proposed for bypassing these problems. A metahuristic is defined as an iterative process and used from learning methods and intelligently concepts to find (near-) optimal solutions and exploring and exploiting the search space [1].

ISSN: 2005-4297 IJCA

Metaheuristics are strategies that guide the search process toward goal. The goal is to efficiently explore the search space in order to find (near-) optimal solutions. Techniques which constitute metahuristic algorithms range from simple local search procedures to complex learning processes. Metahuristic algorithms are approximate and usually non-deterministic. These algorithms as: Genetic [2], particle swam optimization (PSO) [3], artificial bee colony (ABC) [4], Bat algorithm (BA) [5], cuckoo algorithm (CA) [6]. Proposed algorithm is based on ABC algorithm introduced by Karaboga and Basturk. The population is divided into two groups the forager and the others. The no foragers are two types:

- The scout bee
- Onlooker bees.

The forager bee exploits from discovered food source and inform to onlooker bees about the quality of food source. The onlooker bees stay in colony and decision about exploiting from a food source using shared information by forager bee. The scout bees searched the solution space for a new food source that has high quality.

The proposed method increased the power of search for scout bees by BA algorithm. ABC_M are based on standard BA introduced in [5]. BA is the first type of algorithms based on frequency. BA used from setting frequency for decreasing distance to object or suitable food source.

This paper is organized as follows. Section 2 is overview of both ABC and BA algorithm and related work. In section 3 proposed algorithm are introduced. Section 4 is consist of result and benchmark function. In section 5 is said conclusion.

2. BA Algorithm

BA is introduced in 2010 by Xin-She Yang [5]. The base of this algorithm is reflexing of voice. This algorithm has 3 rules as follow.

- Each of bat use reflexing of voice for calculate distance from food source and used by them recognized food and bait.
- The bats moved randomly from location x_i with velocityv_i and tuning frequency automatically and amount of propagation are dependent on the distance to goal and shown by $R\in[0,1]$.
- Intensity of pulse have a different values but the value of it is assumed to have a larger positive value of A_{min}

Pseudo code of BA is shown in Figure 1.

```
Initialize the bat population xi and vi (i = 1, 2, ..., n)
Initialize frequencies fi, pulse rates ri and the loudness Ai
while (t <Max number of iterations)
     Generate new solutions by adjusting frequency,
     Update velocities and locations/solutions 1 to 3
     if (rand > ri)
          Select a solution among the best solutions
          Generate a local solution around the selected best solution
     end if
     Generate a new solution by flying randomly
     if (rand < Ai & f(xi) < f(x*))
          Accept the new solutions
          Increase ri and reduce Ai
     end if
     Rank the bats and find the current best x*
end while
```

Figure 1. Pseudo Standard BA Code [5]

Calculation method of location x_i with velocity v_i in d-dimension search space is shown in formulas (1-3)

$$f_i=f_{min}+(f_{max}-f_{min})\beta \tag{1}$$
$$v_i^{t+1}=v_i^t+(x_i^t-x_*)f_i \tag{2}$$
$$x_i^{t+1}=x_i^t+v_i^{t+1} \tag{3}$$

β is a random vector drawn from a uniform distribution between 0 and 1. x_* is the best local location that obtained from comparison n solution.

2.1. ABC Algorithm

The bees are divided into three categories: forager bees, onlooker bees and scout bees. Random search is done by the scout bees. In this algorithm assumed the numbers of forager bees are equal food source. That means there is one forager bee just for each food source. Each of forager bees goes to their food source and shared his information about quality of food source then onlooker bees choose food sources depending on the experience of themselves. Sharing information is done by dance and dancer is forager bees. Sometimes some forager bees leave his job and converted to onlooker bees. Scouts bees fly and choose the food sources randomly without pay attention to shared information. That means a global search is doing for change in search space.

ABC is a powerful combination for local and global search concurrently. Local search is done by forager and onlooker bees and global search is done by onlooker and scout bees. The quality of a food source is dependent to value of attached fitness function. Possibility of selection for a food source is calculated by formulas (4).

$$P_i=\frac{F(\theta_i)}{\sum_{k=1}^{s} F(\theta_k)} \tag{4}$$

Where P_i is possibility of selection for ith forager bee, s is number of forager bee or food sources, θ_i is proposed answer from ith forager bee and $F(\theta_i)$ is fitness function. The new location for onlooker bee is calculated from (5).

$$x_{ij}(t+1)=x_{ij}(t)+\emptyset(x_{ij}(t)-x_{kj}(t)) \tag{5}$$

x_i, t, k, j are respectively a location for onlooker bee, number of repeat, the randomly selected forager bee, dimension of problem and □ is a random number between -1 and 1.

Pseudo code of ABC is shown in Figure 2.

1. Send the scouts onto initial food sources
2. **REPEAT**
2.1. Send the employed bees onto the food sources and determine their nectar amounts
2.2. Calculate the probability value of the sources with which they are preferred by the onlooker bees
2.3. Stop the exploitation process of the sources abandoned by the bees
2.4. Send the scouts into the search area for discovering new food sources, randomly
2.5. Memorize the best food source found so far
3. **Until**(requirements are met)

Figure 2. Pseudo Standard ABC Code

2.2. Related Work

BA considered attractive in recently years and used in optimization problem. In 2011 proposed a multiobjective optimization by Yang [7]. Metahuristic algorithms have many parameters where must be initialized according to the state of problem [8] proposed a hybrid BAT algorithm and fuzzy logic for dynamically conform parameters. Levy walk is a continuous probability distribution for a non-negative random variable. It is a type of the inverse-gamma distribution and a combined of BA and Levy flight is proposed in [9]. Levy flight is used to increase diversity of population and convergence for BA algorithm. Another combined algorithm is PSO and BA that increased convergence in PSO [10]. If dimensions of a problem is high the metahuristics algorithms trapped in local optimal as ABC (have low convergence) therefor tried increase the velocity of convergence for ABC. PS_ABC is hybrid algorithms do it [11]. ACO–ABC–HS is a hybrid algorithm from ACO, ABC and HS introduced in [12] that used for solving the problem of Economic Dispatch (ED) for a multi-generator system. ABC is powerful in exploration but week in exploitation for this reason tried to improve searching method for global search. Add experience to global search phase is a solution proposed by researcher. Liu *et al* proposed a two steps optimization. The first step, to use previous information for global search and secondly, used an S-type adaptive scaling factors for create balance between the exploration and exploitation [13].CABC is introduced by [14] for improved ABC with modified in search equation. Used from orthogonal experimental design (OED) to form an orthogonal learning (OL) for append experiment to search method. The optimization problems are two type, constraint and unconstraint. Upgraded ABC introduced by Brajevic in [15] for constraint problems. Quick ABC that called qABC used neighborhood radius for optimization local search do by onlooker bees [16]. A binary ABC introduced by Hancer for feature selection [17].

3. Proposed Algorithm

ABC has three agents for search. Employer bees and onlooker bees are doing local search and set of onlooker bees and scout bees are doing global search. The ABC algorithm can be do local search and global search together and local or global search can be extended if needed. The method searching ABC is keeping variety of population. A sign of strength for each of algorithm not to remain in local optimal state and moving toward global optimal as a result number of iteration for algorithm decrease and equal with a minimum number. BA used from a population and during all repetitions been tried moved the population toward global optimization. In Standard BA not have population diversity to enough and result trapped in local optimization. Reflection of voice added to algorithm (calculate distance to source) moved the population toward right direction and increased velocity of convergence. Then BA is a good local explorer.

In proposed algorithm used from BA for local search in ABC. In the last section of ABC, the scout bees exchanged low quality food sources with high quality food source. The all of new food sources added are new population if had the better quality but proposed algorithm used from BA for a local search around old food source because a nearby location can has the better answer from a randomly answer in search space. The neighboring areas around of a location with good quality have a high velocity of convergence toward local optimal that could be global optimal. Therefor a complete local search is more important from global search.

The velocity of proposed algorithm is calculated from formula 2 that is equal difference between current location and best location, multiplication frequency parameter. Frequency parameter needed is equal with probability of selection of a food source. If a food source has high quality then the frequency parameter have larger amount. Whatever frequency parameter have larger amount for a food source (because increased the velocity of movement by frequency parameter in formula 2) then the move toward a food source have high speed but a new food source will be changed if have the better answer from a food source with global search or randomly solutions.

Pseudo code of proposed phase is shown in Figure 3.

1. **Function Send_Scout_Bees()**
2. Probabilities of for each food source is equal **Frequency**
3. For i=1 to Food Number
3.1. If i has the maximum Trial
3.1.1. Calculate new food source(i) with **2 and 3**
3.1.2. $x_i = init(i, S)$
3.1.3. If the new food source is better than the x_i
3.1.3.1. $x_i =$ new food source
3.1.3.2. $Trial_i = 0$
3.2. End If
4. End For

Figure 3. The Proposed Phase Scout Bee

Local search is done by onlooker bee. If a local search method append to onlooker phase then will be an emphasis on local optimal and solutions trapped in local optimal. Thus append search method used in scout bee search phase for have a complete search in radius neighbor.

Fitness functions and dimensions are shown in Table1.

Table 1. Used Benchmark Functions [18]

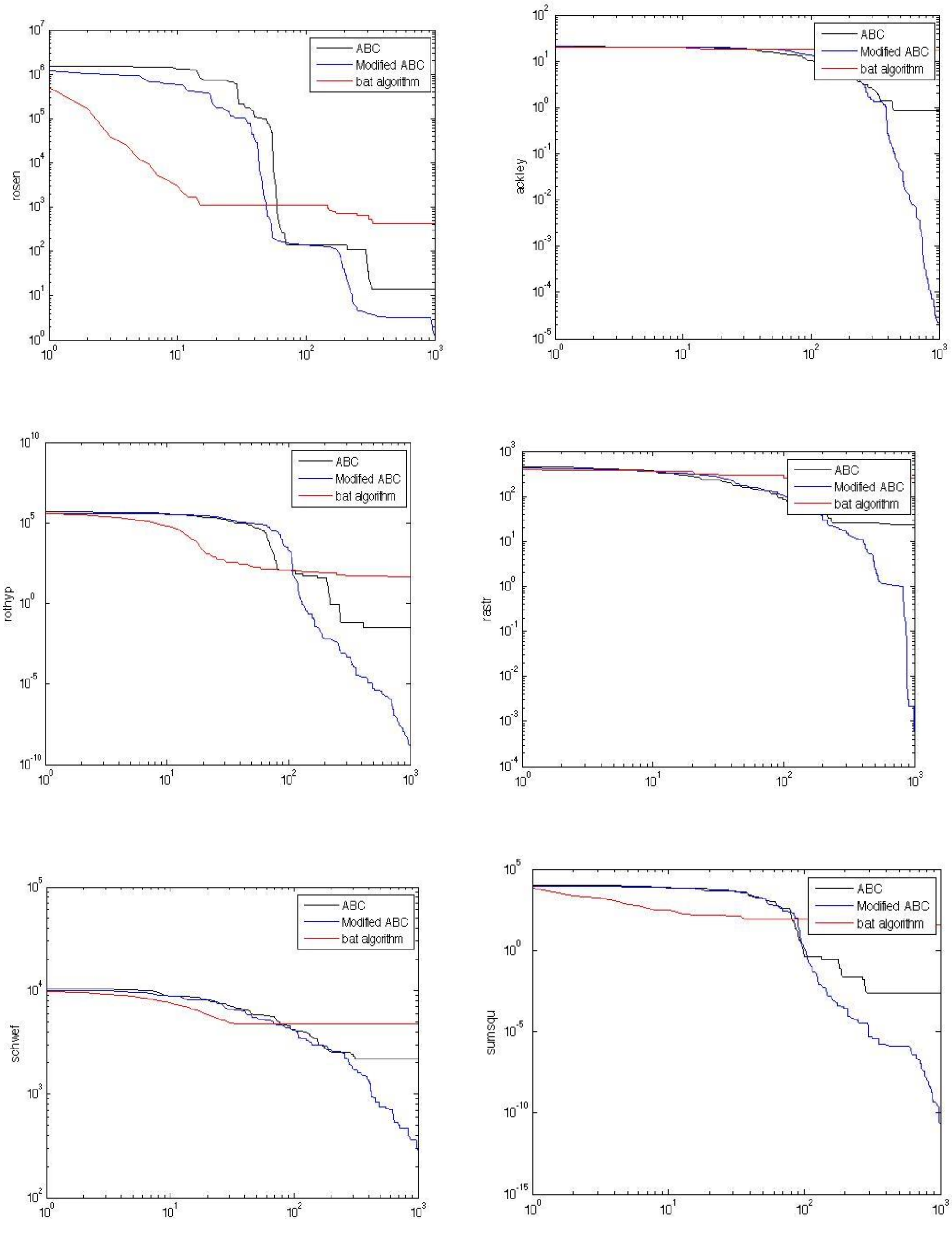
Function name	Function	Range	dimension
Ackley	$f(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$	-32.768 32.768	2 10 30
Rastrigin	$f(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$	-5.12 5.12	2 10 30
Rotated Hyper-Ellipsoid	$f(x) = \sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$	-65.536 65.536	2 10 30
Sum Squares	$f(x) = \sum_{i=1}^{d} ix_i^2$	-10 10	2 10 30
Rosenbrock	$f(x) = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	-5 10	2 10 30
Schwefel	$f(x) = 418.9829d - \sum_{i=1}^{d} x_i\sin(\sqrt{\lvert x_i\rvert})$	-500 500	2 10 30
Dixon-Price	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$	-10 10	2 10 30
Levy	$f(x) = sin^2(3\pi x_1) + (x_1 - 1)^2[1 + sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + sin^2(2\pi x_2)]$	-10 10	2
Schaffer	$f(x) = 0.5 + \frac{\cos(\sin(\lvert x_1^2 - x_2^2\rvert)) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	-100 100	2
Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	-10 10	2
Power Sum	$f(x) = \sum_{i=1}^{d}[\left(\sum_{j=1}^{d} x_j^i\right) - b_i]^2$	0 2	2
Langermann	$f(x) = \sum_{i=1}^{m} c_i\exp(-\frac{1}{\pi}\sum_{j=1}^{d}\left(x_j - A_{ij}\right)^2)\cos(\pi\sum_{j=1}^{d}(x_j - A_{ij})^2)$	0 10	2
Hartmann 3-Dimensional	$f(x) = -\sum_{i=1}^{4}\alpha_i\exp(-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2)$	0 1	3
Hartmann 4-Dimensional	$f(x) = -\sum_{i=1}^{4}\alpha_i\exp(-\sum_{j=1}^{4} A_{ij}(x_j - P_{ij})^2)$	0 1	4

1. Test result

Three different tests have been doing in this paper. In the first test is shown the power of ABC_M versus standard ABC and base BA in 30 dimensions. The second test shown difference of three algorithms in 10 dimension and last section is shown with 2 dimensions.

The populations considered have 50 members and the number of iterations is 1000 for all algorithms.

Figure 4 is shown results of experiments in 30 dimensions.



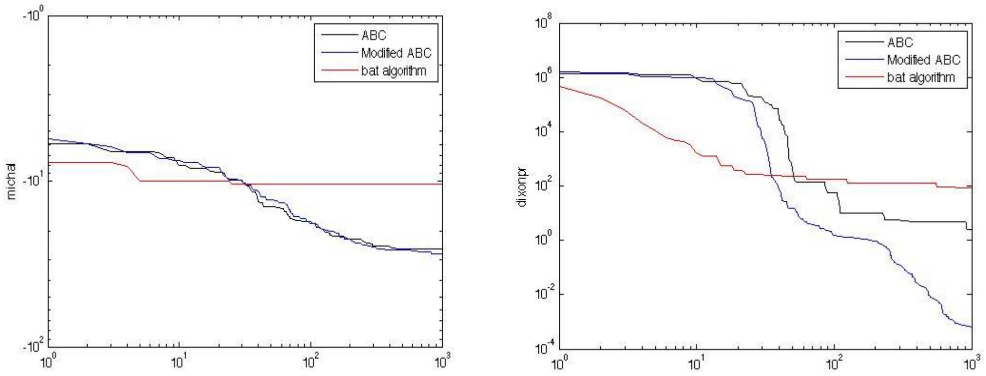


Figure 4. Convergence Toward Global Optimal By 3 Algorithm

X axis is shown the constraint numbers and Y axis is shown the gained number. As shown in Figure 4, often BA stay in local optimal after low number iteration, ABC have the better state from BA and have stepped decrease toward optimal state, but proposed algorithm have regular downward trend and not stay in local optimal and quickly out. Clearly ABC and ABC_M have the better performance and the curve of ABC_M is more convergence. The deference between the best answers is clear for algorithms that shown on Table 2. Table 2 is shown the compare of 6 fitness functions.

Table 2. The Comparison between ABC, ABC_M and BA on Introduced Benchmark Function

Ackley	ABC_M	ABC	BA
Best:	**5.101471e-006**	1.065141e-001	2.037174e+001
Worst:	**2.216343e-005**	2.083150e+000	2.089701e+001
Mean:	**1.204686e-005**	1.027877e+000	2.067176e+001
Median:	**1.062860e-005**	1.095671e+000	2.065904e+001
Std:	**5.007176e-006**	5.368634e-001	1.422591e-001
	rastr		
Best:	**4.758931e-010**	1.204467e+001	3.563558e+002
Worst:	**9.994451e-001**	2.476289e+001	4.802911e+002
Mean:	**3.637441e-002**	1.933593e+001	4.294383e+002
Median	**9.441537e-005**	1.966548e+001	4.326905e+002
Std:	**1.820796e-001**	3.600761e+000	2.728551e+001
	rothyp		
Best:	**5.291545e-010**	2.684995e-003	3.319085e+005
Worst:	**1.291579e-008**	3.631368e-001	5.080897e+005
Mean:	**4.653732e-009**	4.121835e-002	4.083498e+005
Median:	**3.895901e-009**	2.121221e-002	4.099310e+005
Std:	**3.442311e-009**	6.675311e-002	3.920969e+004
	sumsqu		
Best:	**2.724359e-012**	5.232330e-004	5.828139e+003
Worst:	**9.553287e-011**	1.237073e-002	1.212299e+004
Mean:	**2.841885e-011**	3.326384e-003	9.408524e+003
Median:	**2.209743e-011**	2.599462e-003	9.507452e+003
Std:	**2.370625e-011**	2.595479e-003	1.344863e+003
	schwef		
Best:	**3.818339e-004**	1.377434e+003	9.211820e+003

Worst:	**3.883971e+002**	2.394957e+003	1.084633e+004
Mean:	**2.150767e+002**	1.856221e+003	1.011855e+004
Median:	**2.370652e+002**	1.862305e+003	1.012798e+004
Std:	**1.082504e+002**	2.134529e+002	4.638217e+002
	dixonpr		
Best:	**1.635761e-004**	5.635517e-001	7.983915e+005
Worst:	**1.556114e-001**	4.707256e+000	2.261368e+006
Mean:	**1.937928e-002**	2.270301e+000	1.672543e+006
Median:	**3.820555e-003**	2.110952e+000	1.626972e+006
Std:	**3.404066e-002**	9.221321e-001	3.392250e+005

Table 2 is shown comparison of performance for algorithms. Modified ABC have the best answer for all benchmark functions. Comparisons in 10 dimensions are shown in Table 3.

Table 3. The Comparison between ABC, ABC_M and BA on Introduced Benchmark Function

rosen	*ABC_M*	*ABC*	*BA*
Best:	**1.696211e-004**	3.458829e-001	1.570383e+004
Worst:	**6.522282e-002**	2.596179e+000	3.163825e+005
Mean:	**1.321635e-002**	9.818157e-001	1.138710e+005
Median:	**8.223517e-003**	7.508323e-001	1.064571e+005
Std:	**1.564977e-002**	5.582366e-001	6.7
	Ackley		
Best:	**4.440892e-015**	2.176037e-013	1.864162e+001
Worst:	**5.089262e-013**	2.390676e-003	2.045878e+001
Mean:	**3.250733e-014**	2.466865e-004	1.981076e+001
Median:	**4.440892e-015**	1.018185e-005	1.991650e+001
Std:	**9.688422e-014**	5.146518e-004	4.370280e-001
	Rastrigen		
Best:	**0.000000e+000**	6.908305e-003	9.732942e+001
Worst:	**1.278977e-013**	1.113748e+000	1.406937e+002
Mean:	**5.684342e-015**	3.809131e-001	1.195326e+002
Median:	**0.000000e+000**	2.295212e-001	1.236130e+002
Std:	**2.377931e-014**	3.833012e-001	1.098486e+001
	Rothyp		
Best:	**6.255511e-046**	1.846614e-013	1.343584e+004
Worst:	**1.123064e-043**	1.502874e-004	4.145108e+004
Mean:	**1.769728e-044**	3.500195e-005	2.957546e+004
Median:	**6.338529e-045**	2.718758e-005	3.148455e+004
Std:	**2.575881e-044**	3.728033e-005	6.916159e+003
	Sumsqu		
Best:	**3.346181e-049**	2.535251e-045	3.199363e+002
Worst:	**6.976309e-047**	6.354292e-009	9.514835e+002
Mean:	**1.030388e-047**	2.348749e-010	6.666035e+002
Median:	**3.431709e-048**	3.138434e-024	6.730806e+002
Std:	**1.624843e-047**	1.160800e-009	1.561957e+002
	schwef		
Best:	**1.272757e-004**	1.655042e-001	2.411647e+003
Worst:	**1.272801e-004**	1.778498e+002	3.332956e+003
Mean:	**1.272761e-004**	6.676102e+001	2.864260e+003
Median:	**1.272757e-004**	6.429156e+001	2.910966e+003

Std:	**9.639245e-010**	5.927075e+001	2.698886e+002
	michal		
Best:	**-9.660152e+000**	9.659384e+000	6.969835e+000
Worst:	**-9.660065e+000**	9.582708e+000	5.085257e+000
Mean:	**-9.660146e+000**	9.631428e+000	5.961437e+000
Median:	**-9.660152e+000**	9.631517e+000	5.994051e+000
Std:	**1.768865e-005**	2.130625e-002	4.290025e-001
	dixonpr		
Best:	**1.323425e-007**	4.851959e-003	2.621410e+003
Worst:	**5.126544e-005**	9.845917e-002	1.879230e+005
Mean:	**7.225199e-006**	5.459724e-002	8.851881e+004
Median:	**4.104974e-006**	5.267284e-002	8.415593e+004
Std:	**1.034606e-005**	2.560769e-002	4.026242e+004

4. Conclusion

Performance each of algorithms is dependent to local and global search together. A complete and powerful search is more important for lunge out of local optimization. The movement of algorithm is toward global optimal in continuing iteration. Because ABC is simple and so efficient, it attracted the attention of scientists. It is excellent in discovery and weak in exploitation, there for most of optimizations ABC are done in exploitation phase. BA used from tuning frequency for calculation distance from optimal. ABC has a slow convergence speed so a local search append to global search is useful for increase convergence speed. Proposed algorithm is a hybrid ABC and BA for increase speed of convergence, lunge out of local optimization or optimized exploitation. The experiment showed that the proposed algorithm produced the better results and has higher velocity of convergence or decreasing trend and can jump out of local optimization.

References

[1] I.H. Osman and G. Laporte, “Metaheuristics: A bibliography”, Res., vol. 63, **(1996)**, pp. 513–623.

[2] J. H. Holland, “Adaption in natural and artificial systems”, The University of Michigan Press,Ann Harbor, MI. **(1975)**.

[3] J. Kennedy and R. Eberhart, “Particle Swarm Optimization”, Proceedings of the 1995 IEEE International Conference on Neural Networks, **(1995)**, pp. 1942-1948, IEEE Press.

[4] D. Karaboga and B. Basturk, “A comparative study of artificial bee colony algorithm”, Appl. Math. Comput., vol. 214, no. 1, **(2009)**, pp. 108-132.

[5] X.-S. Yang, “A new metaheuristic Bat-inspired Algorithm”, Studies in Computational Intelligence, vol. 28, no. 4, (**2010**), pp. 65–74.

[6] A. Gandomi, X.S.Yang and A.H. Alavi, “Cuckoo search algorithm: a metahuristic approach to solve structural optimization problems”, Engineering with Computers, vol. 29, Issue 1, pp. 17-35.

[7] X.-S. Yang, “Bat algorithm for multi-objective optimization”, International Journal of Bio-Inspired Computation Bio-Inspired Computation (IJBIC), vol. 3, no. 5, **(2011)**, pp. 267–274.

[8] J. Pérez, F. Valdez and O. Castillo, “A New Bat Algorithm with Fuzzy Logic for Dynamical Parameter Adaptation and Its Applicability to Fuzzy Control Design”, Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics, vol. 574, **(2014)**, pp. 65-79.

[9] J. Xie, Y. Zhou and H. Chen, “A Novel Bat Algorithm Based on Differential Operator and Lévy Flights Trajectory”, Computational Intelligence and Neuroscience, **(2013)**.

[10] T.S. Pan, “Hybrid Particle Swarm Optimization with Bat Algorithm”, Genetic and Evolutionary Computing, **(2015)**.

[11] Zh. Li, “PS–ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems”, Expert Systems with Applications, vol. 42, Issue 22, **(2015)**, pp.8881–8895.

[12] T. Sen, H.D. Mathur, “A new approach to solve Economic Dispatch problem using a Hybrid ACO–ABC–HS optimization algorithm”, International Journal of Electrical Power & Energy Systems, vol. 78, pp. 735–744, **(2016)**.

[13] J. Liu, “An Artificial Bee Colony algorithm with guide of global & local optima and asynchronous scaling factors for numerical optimization”, Applied Soft Computing, vol. 37, **(2015)**, pp. 608-618.

[14] W.F. Gao, "A Novel Artificial Bee Colony Algorithm Based on Modified Search Equation and Orthogonal Learning", IEEE Transactions on Cybernetics, Vol. 43, Issue 3, **(2013)**, pp. 1011 – 1024.
[15] I. Brajevic, and T. Milan, "An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems", Intelligent Manufacturing, vol. 24, Issue 4, **(2013)**, pp. 729-740.
[16] D. Karaboga and B. Gorkemli, "A quick artificial bee colony (qABC) algorithm and its performance on optimization problems", Applied Soft Computing, vol. 23, **(2014)**, pp. 227–238.
[17] E. Hancer, "A binary ABC algorithm based on advanced similarity scheme for feature selection", Applied Soft Computing, vol. 36, **(2015)**, pp. 334–348.
[18] http://www.sfu.ca/~ssurjano/index.html.