Blend of Local and Global Variant of PSO in ABC

Tarun Kumar Sharma
School of Mathematics and Computer Applications
Thapar University
Patiala, India
taruniitr1@gmail.com

Millie Pant
Department of Applied Sciences & Engineering
Indian Institute of Technology
Roorkee, India
millidma@gmail.com

Ajith Abraham[1, 2]
[1] Machine Intelligence Research Labs (MIR Labs), WA, USA
[2] IT4Innovations, VSB-Technical University of Ostrava, Czech Republic
ajith.abraham@ieee.org

***Abstract*—Artificial bee colony is a recently proposed metaheuristic optimization technique and is a new member of swarm intelligence based algorithms. It mimics the foraging behavior of honey bees. The performance of Artificial Bee Colony (ABC), like other metaheuristics, is heavily dependent on the tradeoff between their exploration and exploitation aptitude. In this paper a variant called Local Global variant Artificial Bee Colony (LGABC) is proposed to balance the exploration and exploitation in ABC. The proposal harnesses the local and global variant of Particle Swarm Optimization (PSO) into ABC. The proposed variant is investigated on a set of thirteen well known constrained benchmarks problems and three chemical engineering problems, which show that the variant can get high-quality solutions efficiently.**

***Index Terms*—Artificial Bee Colony, Metaheuristic, PSO, Optimization, Swarm Intelligence**

I. Introduction

Now a day's swarm intelligence gathers a strong attraction of researchers, academicians and scientist to solve many complex optimization problems emerging in almost every field. Swarm intelligence is a branch of nature inspired algorithms focused on insect behavior to develop metaheuristics which can imitate social insect's problem solution abilities [1]-[[5]. Swarm intelligence is a heuristic method that models the population of entities that are able to self-organize and interact among them [6][7][24-29].

Many swarm intelligence algorithms have been proposed to solve optimization problems such as PSO (Particle Swarm Optimization) [8], CSO (Cat Swarm Optimization) [9], ACO (Ant Colony Optimization) [10] and ABC (Artificial Bee Colony) [1].

Karaboga in 2005 [1] proposed ABC, inspired and motivated by the intelligent foraging social behavior of honey bees. The performance of ABC in terms of efficiency and accuracy are analyzed with that of PSO, DE (Differential Evolution) and the EA (Evolutionary Algorithms) for both unconstrained and constrained numerical optimization problems [11][12].

Like other metaheuristics, ABC has certain inherent drawbacks like: ABC is good at exploration while poor at exploitation. Further, it is also observed that in structure of basic ABC the onlooker bee, can only move straight to one of the food sources of those are discovered by the employed bees. This characteristic may constrict the search area in which the bees can explore and could become a drawback of the ABC.

In this study an attempt is made to balance the trade-off between exploration and exploitation in ABC by embedding the local and global variants of particle swarm optimization (PSO). The proposed variant based on this scheme is termed as LG-ABC (Local Global variant Artificial Bee Colony).

The rest of the paper is structured as follows. Section 3 describes basic PSO; Section 4 is devoted to the description of the proposed variant, LG-ABC of ABC. Parameter settings are given in section 5 and considered constrained test benchmark problems and chemical engineering problems with result discussions are presented in Section 6 & 7 respectively. The paper closes with conclusions in Section 8.

II. Artificial Bee Colony (ABC)

A. Unconstrained ABC

In ABC the colony of honey bees is comprised of three types of bees namely scouts, employed and onlooker bees. The bees in the colony perform tasks like searching for the nector and sharing the information about the food source intelligently by dividing the labor themselves. The main difference between ABC and other intelligent swarm based algorithms is that in ABC food sources (the population generated) represents the solutions of the problem, not the bees.

The scout bee initiates the food sources randomly which is later exploited by employed bees. The employed bees pass the information about the food source based on their nectar quality to the onlooker bees waiting in the hive. This sharing of information is done by performing a special dance called waggle dance. In ABC the number of employed bees is equal to the number of food sources and each employed bee is assigned to one of the food sources. Employed bees upon reaching to the food source, calculate a new location or fly to the nearby position from the old and preserve the best position. This is a greedy selection process. The number of onlooker bees is also the same as that of employed bees and are allocated to the food sources based on their profitability. Similarly as the employed bees, onlooker bees also calculate the new position from the old one. If the food source does not improve after predetermined number of iterations, then employed bees abandons that food source and becomes scouts and searches the new food source randomly. Mathematical explanation of the complete process is described below.

- Mathematical presentation of ABC

Define *SN* as the colony size of the bees. SN_E is the colony size of employed bees and SN_O as the onlooker bees, which satisfies the equation: $SN = SN_E + SN_O$. As stated above SN_E is equals to SN_O.

N is the dimension of the individual solution vector.The basic ABC can be expressed as:

1) **Initialization**: A set of feasible food sources $(x_1,x_2,..., x_N)$ is randomly initialized and the specific solution x_i can be generated using equation (1) given below:

$$x_{ij} = x_{Lj} + rand(0,1)(x_{Uj} - x_{Lj}) \quad (1)$$

where $j \in \{1,2,\ldots,N\}$ is the j^{th} dimension of the solution vector. Calculate the fitness value of each solution vector respectively.

2) **Employed Bee Movements**: Search new solutions for an employed be in the neighborhood of the current position vector according to the equation (2):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

where $x \in SN$, $j \in \{1,2,\ldots,N\}$, $k \in \{1,2,\ldots,SN_E\}$, $k \neq i$, $Ø_{ij}$ is random number between -1 and 1.

3) **Selection**: Apply the greedy selection operator to choose the better solution between searched new vector v_{ij} and the original vector x_{ij} into the next generation. The greedy selection operator ensures that the population is able to retain the elite individual, and accordingly the evolution will not retreat.

4) **Nectar Evaluation by Onlooker Bee**: Each onlooker bee selects an employed bee from the colony according to their fitness values. The probability distribution (p_i) of the selection operator can be described as follows.

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN_E} fit_i} \quad (3)$$

where fit_i is the fitness value of the solution *i* which is proportional to the nectar amount of the food source in the position *i*.

5) **Onlooker Bee Movements**: The onlooker bee searches in the neighborhood of the selected employed bee's position to find new solutions using equation (2). The updated best fitness value can be denoted with *f* best, and the best solution parameters.

6) **Scout Movement**: If the searching times surrounding an employed bee exceeds a certain threshold *limit*, but still could not find better solutions, then the location vector can be reinitialized randomly according to the equation (1).

7) If the iteration value is larger than the maximum number of the iteration then stop, else, go to 2.

B. Constrained ABC

ABC was originally designed for solving the unconstrained optimization problems [12]. However, with small changes it can easily be modified for dealing with problems having constraints as well. In the present study, we have followed '*three feasibility rules*' method given in [13] to decide which solution vector (*food source)* will be beneficial for handling constraints. An advantage of this method is that unlike penalty method we need not have a penalty constant, which itself is a tedious work to decide. Moreover, here we consider feasible as well as infeasible solutions and prioritize these solutions as per the following rules:

1). If we have two feasible food sources, we select the one giving the best objective function value.
2). If one food source is feasible and the other one is infeasible, we select the feasible one;
3). If both food sources turn out to be infeasible, the food source giving the minimum constraint violation is selected.

It can be observed that these rules bias feasible food sources over infeasible food sources and a pairwise comparison (tournament selection) is done to select the best option.

In this method a control parameter called *modification rate* (MR), pre defined by the user is introduced. With the help of MR, it is decided stochastically whether a food source x_i should be retained or not. It given by the equation:

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), & if\ R_j \le MR \\ x_{ij}\ , & otherwise \end{cases} \quad (4)$$

Where $k \in \{1, 2,..., SN\}$ – randomly chosen such that $k \neq i$; $j \in \{1, 2,...,N\}$.R_j is generated randomly between 0 and 1in each iteration.

III. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO), introduced in 1995 by Eberhart and Kennedy [8], is a stochastic, population–based metaheuristic algorithm for solving numerical optimization problems. Its dynamics is based on principles that govern socially organized groups of individuals called particles. In PSO's context, the population is called a swarm and its individuals (search points) are called particles. Each particle in the swarm has the following three main characteristics:

1) an adaptable velocity with which it moves in the search space,

2) a memory where it stores the best position it has ever visited in the search space (i.e., the position with the lowest function value), and

3) the social sharing of information, i.e., the knowledge of the best position ever visited by all particles in its neighborhood.

The particles of the swarm fly through a multidimensional search space looking for a potential solution. Each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors (or colleagues). For an N-dimensional search space the position of the i^{th} particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$. Each particle maintains a memory of its previous best position $P_{besti} = (p_{i1}, p_{i2} \ldots p_{iN})$. The best one among all the particles in the population is represented as $P_{gbest} = (p_{g1}, p_{g2} \ldots p_{gN})$. The velocity of each particle is represented as $V_i = (v_{i1}, v_{i2}, \ldots v_{iN})$. In each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. The two basic equations which govern the working of PSO are that of velocity vector and position vector given by:

$$v_{ij} = \chi[v_{ij} + c_1 r_1 (p_{ij} - x_{ij}) + c_2 r_2 (p_{gj} - x_{ij})] \qquad (5))$$

$$x_{ij} = x_{ij} + v_{ij} \qquad (6)$$

)

where i= 1,2,...,N; χ is the constriction coefficient; c_1 and c_2 are positive acceleration constants, referred to as cognitive and social parameters, respectively; and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1].

IV. LG-ABC: A Proposed Approach

The performance of a population–based algorithm is heavily dependent on the trade–off between its exploration and exploitation abilities, i.e., its ability to explore wide areas of the search space and its ability to converge rapidly towards the most promising solutions, respectively. The basic ABC performs good exploration but poor exploitation [14]. In this study we tried to balance exploration and exploitation aptitude of basic ABC by embedding the local and global variants of particle swarm optimization [15].

Exploration is promoted by local variant of PSO since the information regarding the best position of each neighborhood is communicated to the rest of the particles in swarm through neighboring particles. On the other hand, the global variant has better exploitation properties is promoted by the global variant of PSO since all particles are attracted by the same best position, thereby converging faster towards the same point.

Let G_i and L_i denote the velocity update of the particle X_i for the global and local PSO variant, respectively [15],

$$G_{ij} = \chi\, [v_{ij} + c_1 r_1 (p_{ij} - x_{ij}) + c_2 r_2 (p_g - x_{ij})] \qquad (7)$$

$$L_{ij} = \chi\, [v_{ij} + c_1 r'_1 (p_{ij} - x_{ij}) + c_2 r'_2 (p_{gi} - x_{ij})] \qquad (8)$$

where g is the index of the best particle of the whole swarm (global variant); and g_i is the index of the best particle in the neighborhood of x_i (local variant). Now the scheme is defined by:

$$U_{ij} = u\, G_{ij} + (1 - u)\, L_{ij} \qquad (9)$$

$$x_{ij} = x_{ij} + U_{ij} \qquad (10)$$

where $u \in [0,1]$ is a *unification factor* parameter [15], which balances the influence of the global and local search directions in the unified scheme. The standard global PSO variant is obtained by setting $u = 1$ in Eq. (5), while $u = 0$ corresponds to the standard local PSO variant. All values $u \in [0,1]$, correspond to composite variants of PSO that combine the exploration and exploitation characteristics of the global and local variant.

In LG-ABC the position of new food source is calculated using:

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), & if\ R_j \le MR \\ x_{ij} = x_{ij} + U_{ij}, & otherwise \end{cases} \qquad (11)$$

The pseudocode of the proposed variant is given below in Figure 1.

```
Begin
    Initialize the population set of food sources x_i, i=1,...,SN
    Evaluate each x_i
    g = 1
Repeat
    For i = 1 to SN
    Generate v_i,g with x_i,g-1 (new solutions for employed
     bees) using equation (4)
    Apply Deb's method for selection process
    Evaluate v_i,g

    If v_i,g < x_i,g-1
            x_i,g = v_i,g
    Else
            x_i,g = x_i,g-1
    End If
    End For
    For i = 1 to SN
            Select, based on fitness proportional
             selection food source x_i,g
            Generate v_i,g with x_i,g by equation (11)
     Apply Deb's method [13] for selection process
     between v_i,g & x_i,g
Evaluate v_i,g

    If v_i,g < x_i,g
            x_i,g = v_i,g
    End If
    End For
             Generate new food sources at random for
             those whose limit to be improved has been
             reached using equation (1)
            Keep the best solution so far
    g = g + 1
     Until g = MCN
  End
```

Fig. 1. Pseudocode of proposed variant LG-ABC.

V. Parameter Settings

In this case study we have compared the performance of proposed LG-ABC with the basic versions of GA, PSO, DE and ABC. Their key characteristics are described as:

- **GA** [17]: The size of population is taken as 100 and two point crossover along with standard single point mutation and ranking selection are used.
- **PSO** [18]: A classic Particle Swarm Optimization model for numerical optimization has been considered. The parameters are c1 = 2.8, c2 = 1.3, and w from 0.9 to 0.4. Population is composed by 100 individuals
- **DE** [19]: A classic Differential Evolution model is considered where F & CR are fixed to 0.5 & 0.9, respectively, and the population size to 100.
- **ABC and LG-ABC:** The colony size (*SN*) or the number of solutions in the colony is 40, the value of modification rate (*MR*) is 0.4, and the maximum cycle number (*MCN*) is 6000.

All the considered algorithms have been run for 30 times for each test function in C++. The stopping criterion is, for all algorithms, 1000 iterations.

VI. Benchmark Test Functions

To validate and inspect the performance of the LG-ABC, 13 benchmark test functions have been taken from [16]. The considered test function include different type of objective functions (e.g., linear, nonlinear, and quadratic) and constraints [e.g., linear inequality, nonlinear equalities, and nonlinear inequalities). Among which the test functions g02, g03, g08, and g12 are maximization problems and are transformed into minimization problem by multiplying it with "−" i.e. $-f(x)$. And rest of the test functions are minimization problems. The simulated results for benchmark test functions obtained by proposed LG-ABC are almost equal to the *known* optimal values and are presented in Table 1. From the Table 1 it can be observed that LG-ABC is able to find the global optima consistently in 12 test functions over 30 runs with the exclusion of test function $g02$. In case of $g02$ the optimal solutions are not consistently found, but the result achieved is very close to the global optimal solution.

From Table 1, a better result is indicated in boldface or that the global optimum (or best known solution) was reached. (–) Means that no feasible solutions were found.

The success rates of GA, DE, PSO, and basic ABC algorithm with the proposed LG-ABC algorithm are presented in Table 2. Comparing the results in terms of success rates, LG-ABC algorithm outperforms all algorithms employing Deb's rules.

The statistical simulation results in terms of best, median, worst and standard deviation (Std. Dev.) for the proposed variant LG-ABC are presented in Table 3.

TABLE I. The Mean Solutions Obtained by GA, PSO, DE, ABC & LG-ABC for 13 Benchmark Test Functions

f	**Optimal**	**GA**	**PSO**	**DE**	**ABC**	**LG-ABC**
*g*01	-15.000	-14.236	-14.710	-14.555	-15.000	**-15.000**
*g*02	0.803619	0.788588	0.419960	0.665	0.792412	**0.799672**
*g*03	1.000	0.976	0.764813	**1.000**	**1.000**	**1.000**
*g*04	-30665.539	**-30590.455**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
*g*05	5126.498	-	5135.973	5264.270	5185.714	**5135.242**
*g*06	-6961.814	-6872.204	**-6961.814**	-	-6961.219	-6961.723
*g*07	24.306	34.980	32.407	**24.310**	24.473	24.413
*g*08	0.095825	0.095799	**0.095825**	**0.095825**	**0.095825**	**0.095825**
*g*09	680.63	692.064	**680.630**	**680.630**	680.640	**680.630**
*g*10	7049.25	10003.225	7205.5	**7147.334**	7224.407	7213.786
*g*11	0.75	**0.75**	0.749	0.901	**0.750**	**0.750**
*g*12	1.000	**1.000**	0.998875	**1.000**	**1.000**	**1.000**
*g*13	0.053950	-	0.569358	**0.872**	0.968	0.897

TABLE II. Success rates of algorithms when compared with that of the ABC algorithm run through 1000 Iterations, dually. + indicates that algorithm is better while − indicates it is worse than other. If both algorithms show similar performance, they are both +

f	**LG-ABC − GA**		**LG-ABC − PSO**		**LG-ABC − DE**		**LG-ABC − ABC**	
	LG-ABC	***GA***	***LG-ABC***	***PSO***	***LG-ABC***	***DE***	***LG-ABC***	***ABC***
g01	+	−	+	−	+	−	+	+
g02	+	−	+	−	+	−	+	−
g03	+	−	+	−	+	+	+	+
g04	+	−	+	+	+	+	+	+
g05	+	−	−	+	+	−	+	−
g06	+	−	−	+	+	−	+	−
g07	+	−	+	−	−	+	+	−
g08	+	−	+	+	+	+	+	+
g09	+	−	−	+	−	+	+	−
g10	+	−	−	+	−	+	+	−
g11	+	+	+	−	+	−	+	+
g12	+	+	+	−	+	+	+	+
g13	−	−	−	+	−	+	−	−
Total	12	2	8	7	9	8	12	6

TABLE III. EXPERIMENTAL STATISTICAL RESULTS OF LG-ABC FOR 13 BENCHMARK TEST FUNCTIONS

f	Optimal	Best	Mean	Median	Worst	Std. Dev.
g01	-15.000	-15.000	-15.000	-15.000	-15.000	6.84E-11
g02	0.803619	0.803494	0.799672	0.809833	0.683522	3.28E-02
g03	1.000	1.000	1.000	1.000	1.000	6.84E-15
g04	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	8.64E-13
g05	5126.498	5129.045	5135.242	5137.9934	5138.036	5.12E-13
g06	-6961.814	-6961.842	-6961.723	-6961.703	-6961.401	4.84E-12
g07	24.306	24.306	24.413	24.516	24.963	6.98E-11
g08	0.095825	0.095825	0.095825	0.095825	0.095825	4.74E-18
g09	680.63	680.63	680.630	680.639	680.641	9.17E-12
g10	7049.25	7076.867	7213.786	7224.68	7425.98	4.52E-03
g11	0.75	0.750	0.750	0.750	0.750	5.23E-14
g12	1.000	1.000	1.000	1.000	1.000	7.01E-01
g13	0.053950	0.62192	0.897	0.90233	0.90412	3.05E-13

VII. OPTIMIZATION OF CHEMICAL ENGINEERING PROBLEMS

Three chemical engineering problems are considered to further test the efficiency of the proposed variant of ABC. The first problem is:

1. *Reactor network design problem*

The problem taken from [20] is about the design of a sequence of two CSTR reactors (*say* V_1 & V_2). The objective of this problem is to find the volumes of both reactors and concentration of each product in each tank (C_{A1}, C_{A2}, C_{B1} & C_{B2}) in order to maximize the concentration of product in the exit stream. The detail can be found in [21]. The optimization problem converted to minimization is

$$Minimize\ F = -C_{B2}$$

$$w.r.t \quad C_{A1} + k_1 C_{A1} V_1 - 1 = 0$$

$$C_{A2} - C_{A1} + k_2 C_{A2} V_2 = 0$$

$$C_{B1} + C_{A1} + k_3 C_{B1} V_1 - 1 = 0$$

$$C_{B2} - C_{B1} + C_{A2} - C_{A1} + k_4 C_{B2} V_2 = 0$$

$$4 - V_1^{0.5} - V_2^{0.5} \geq 0$$

such that $0 \leq C_{A1}, C_{A2}, C_{B1}, C_{B2} \leq 1$ and $10^{-5} \leq V_1$ & $V_2 \leq 16$
where $k_1 = 0.09755988$, $k_2 = 0.99k_1$, $k_3 = 0.0391908$, and $k_4 = 0.9k_3$

Result & discussions: The reported global optimum values are ($V_1 = 3.036504$, $V_2 = 5.096052$, $C_{A1} = 0.771462$, $C_{B1} = 0.204234$, $C_{A2} = 0.516997$ and $C_{B2} = 0.388812$ with $F = -0.388812$).

This problems was solved by [21] & [22] by reformulating it. In this study we tried to solve it without reformulating it with colony size of 40 bees and simulations were executed 25 times with MR=0.4 and obtained the values for $V_1 = 3.036876$, $V_2 = 5.097129$, $C_{A1} = 0.771447$, $C_{B1} = 0.204238$, $C_{A2} = 0.516986$ and $C_{B2} = 0.388799$ with $F = -0.388812$. The problem took 2113 NFE to reach to the reported global value. The graph showing average objective function value is plotted in Figure 2.

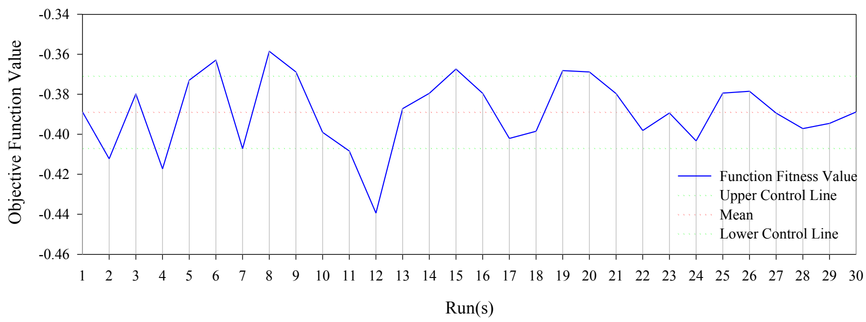

Fig. 2. Plot of Mean objective function value for Reactor network design problem

2. *Heat exchanger network design problem*

This problem has been taken from [23] which address the design of a heat exchanger (A_1, A_2, A_3) network. The objective in this problem is to minimize the overall heat exchange area.

$$Minimize\ F = A_1 + A_2 + A_3$$

$$w.r.t \quad 0.0025(T_1 + T_3) - 1 = 0$$

$$0.0025(-T_1 + T_2 + T_4) - 1 = 0$$

$$0.01(-T_2 + T_5) - 1 = 0$$

$$100A_1 - A_1T_3 + 833.33252T_1 - 83333.333 \leq 0$$

$$A_2T_1 - A_2T_4 - 2500T_1 + 1250T_2 \leq 0$$

$$A_3T_2 - A_3T_5 - 2500T_2 + 1{,}250{,}000 \leq 0$$

such that $100 \leq A_1 \leq 10{,}000$; $1000 \leq A_{i=2,3} \leq 10{,}000$; $10 \leq T_{i=1,2,3,4,5} \leq 1000$;

The reported global optimum is at (A_1 = 579.19, A_2 = 1360.13, A_3 = 5109.92, T_1 = 182.01, T_2 = 295.60, T_3 = 217.99, T_4 = 286.40 and T_5 = 395.60 with F = 7049.25).
Similarly this problem was also reformulated by [21] and solved. But again in this case we have solved this problem with out reformulating it and with the same parameter settings as above. This problem took 37823 NFE to converge to the reported global optima. The obtained values using the proposed variant are A_1 = 579.21, A_2 = 1360.25, A_3 = 5108.99, T_1 = 182.08, T_2 = 295.67, T_3 = 217.97, T_4 = 286.64 and T_5 = 395.76 with F = 7049.249. The graph showing average objective function value is plotted in Figure 3.

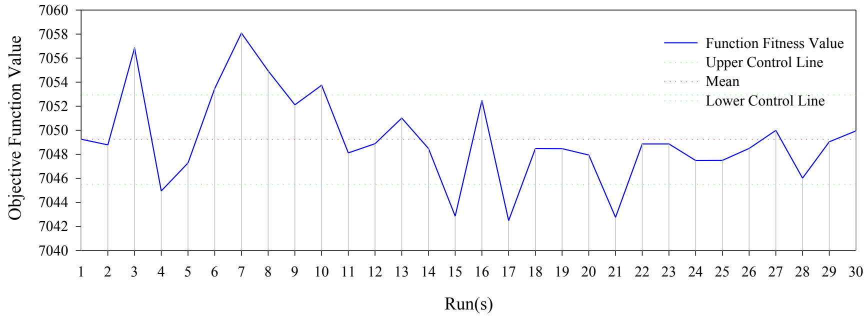

Fig. 3. Plot of Mean objective function value for Heat exchanger network design problem.

3. *Separation network synthesis problem*

This problem is taken from [20]. The detail problem may be consulted from [20] & [22] and superstructure for the separation of a three component mixture into two products. There are 22 variables and 16 equality constraints in the problem. The optimization problem is expressed as:

$$\textit{Minimize } F = 0.9979 + 0.00432F_1 + 0.00432F_{13} + 0.01517F_2 + 0.01517F_9$$

$$\begin{aligned}
\textit{w.r.t}\quad & F_1 + F_2 + F_3 + F_4 - 300 = 0 \\
& F_5 - F_6 - F_7 = 0 \\
& F_8 - F_9 - F_{10} - F_{11} = 0 \\
& F_{12} - F_{13} - F_{14} - F_{15} = 0 \\
& F_{16} - F_{17} - F_{18} = 0 \\
& F_{13}x_{A,12} - F_5 + 0.333F_1 = 0 \\
& F_{13}x_{B,12} - F_8x_{B,8} + 0.333F_1 = 0 \\
& -F_8x_{C,8} + 0.333F_1 = 0 \\
& -F_{12}x_{A,12} + 0.333F_2 = 0 \\
& F_9x_{B,8} - F_{12}x_{B,12} + 0.333F_2 = 0 \\
& F_9x_{C,8} - F_{16} + 0.333F_2 = 0 \\
& F_{14}x_{A,12} + 0.333F_3 + F_6 - 30 = 0 \\
& F_{10}x_{B,8} - F_{14}x_{B,12} + 0.333F_3 - 50 = 0 \\
& F_{10}x_{C,8} + 0.333F_3 + F_{17} - 30 = 0 \\
& x_{B,8} - x_{C,8} - 1 = 0 \\
& x_{A,12} - x_{B,12} - 1 = 0
\end{aligned}$$

such that $0 \le F_{i=1,2,\ldots,18} \le 150$; $0 \le x_{A,j=1,2,\ldots,18} \le 1$; $0 \le x_{B,j=1,2,\ldots,18} \le 1$; $0 \le x_{C,j=1,2,\ldots,18} \le 1$.

It is reported in the literature that the problem was solved using a α-BB algorithm [20] and obtained the solution at F = 1.8640. With the same parameter settings stated above and with out formulating it we have solved and obtained the following values: F_1 = 60.02, F_2 = 0.0, F_3 = 90.0, F_4 = 152.0, F_5 = 20.0, F_6 = 0.0, F_7 = 20.0, F_8 = 40.0, F_9 = 40.0, F_{10} = 0.0, F_{11} = 0.0, F_{12} = 20.0, F_{13} = 0.0, F_{14} = 20.0, F_{15} = 0.0, F_{16} = 20.0, F_{17} = 0.0, F_{18} = 20.0, $x_{B,8}$ = 0.5, $x_{C,8}$ = 0.5, $x_{A,12}$ = 0.0 and $x_{B,12}$ = 1.0 with F = 1.86401. The graph showing average objective function value is plotted in Figure 4.

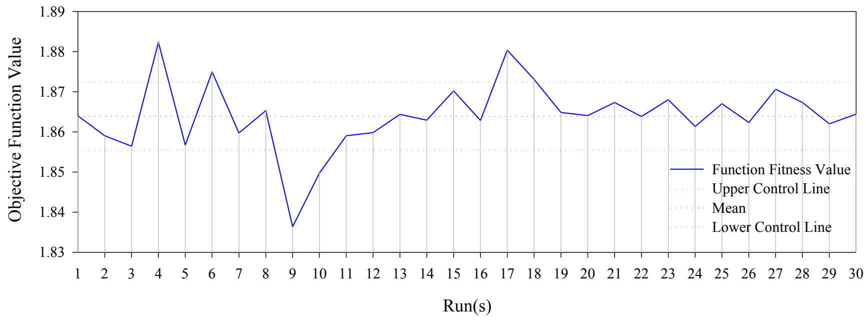

Fig. 4. Plot of Mean objective function value for Separation network synthesis problem

VIII. CONCLUSIONS

In this research, the two global and the local variant of PSO, are embedded into the structure of basic ABC to enhance and balance the exploration and exploitation capability. In the global variant of PSO, the whole swarm is considered as the neighborhood of each particle, while in the local, strictly smaller neighborhoods are used. The proposed variant called LG-ABC is investigated on a set of six benchmark and three problems taken from the chemical industry. The statistical inferences show the efficiency of the proposal. Interesting findings on the behavior of LG-ABC and ABC were observed from the results of 13 test benchmark functions and three chemical engineering problems:

- LG-ABC clearly improved ABC's exploration and exploitation capabilities to reach better final results, based on both, quality and consistency.
- LG-ABC was able to reach the vicinity of the best known or global optimum solution more frequently with respect to ABC and other state-of-art algorithms taken for comparison.

ACKNOWLEDGMENTS

This work was partially supported in the framework of the IT4 Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 by operational programme 'Research and Development for Innovations' funded by the Structural Funds of the European Union and state budget of the Czech Republic, EU.

REFERENCES

[1] D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University; 2005.

[2] Karaboga D, Basturk B, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Vol. 4529/2007, No. 0302-9743, 2007, pp.789–798.

[3] N. Gomez, L. F. Mingo, J. Bobadilla, F. Serradilla, J. A. C. Manzano: Particle Swarm Optimization models applied to Neural Networks using the R language, WSEAS Transactions on Systems, 9(2), 2010, pp. 192-202

[4] R. M. Idris, A. Khairuddin and M.W. Mustafa, Optimal Allocation of FACTS Devices in Deregulated Electricity Market Using Bees Algorithm, WSEAS Transactions on Power Systems, Vol. 5, Issue 2, Apr 2010, pp. 108-119.

[5] A.V. Baterina, C. Oppus, Image Edge Detection Using Ant Colony Optimization, WSEAS Transactions On Signal Processing, Vol. 6, No. 2, 2010, pp.58-67.

[6] Milan Tuba: Swarm Intelligence Algorithms Parameter Tuning, Proceedings of the American Conference on Applied Mathematics, Harvard, Cambridge, USA, pp. 389-394, 2012.

[7] L. Jiann-Horng, H. Li-Ren: Chaotic bee swarm optimization algorithm for path planning of mobile robots, Proceedings of the 10th WSEAS international conference on evolutionary computing, pp. 84-89, 2009.

[8] Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings Sixth Symposium on Micro Machine and Human Science, Piscataway, NJ, IEEE Service Center (1995) 39–43G.

[9] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, Cat Swarm Optimization, Proc. of 9th Pacific Rim International Conference on Artificial Intelligence, Guilin, China, LNCS 4099, pp.854-858, 2006.

[10] M. Dorigo and L. M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. on Evolutionary Computation, vol.26, no.1, pp.53-66, 1997.

[11] D. Karaboga and B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing, 8 (2008), pp.687-697, 2008.

[12] D. Karaboga and B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Springer-Verlag, IFSA (2007), pp. 789–798.

[13] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2/4, pp. 311–338, 2000.

[14] Zhu, G, Kwong, S., 2010, Gbest-guided artificial bee colony algorithm for numerical function optimization, Applied Mathematics and Computation, 217(7), 3166-3173.

[15] Parsopoulos, K.E., Vrahatis, M.N.: UPSO: A unified particle swarm optimization scheme. In: Lecture Series on Computer and Computational Sciences, Vol. 1, Proc. Int. Conf. Computational Methods in Sciences and Engineering, VSP International Science Publishers, Zeist, The Netherlands (2004) 868–873.

[16] Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4(3):284–294.

[17] Golberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Massachusetts, 1989.

[18] A. Kennedy, J. and R. Eberhart, \Particle swarm optimization," Proceedings of the IEEE International Conference Neural Networks, Vol. 4, 1942{1948, 1995.

[19] D. Storn, R. and Price, K., "Differential Evolution — a Simple and Effecient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, 11 (1997) pp. 341–359.

[20] Adjiman,C.S. , Androulakis,I.P. , Floudas,C.A. , A Global Optimization Method, alphaBB, for General Twice-Differentiable Constrained NLPs: II - Implementation and Computational Results, Computers and Chemical Engineering, 1998, Vol:22, 1159-1179.

[21] B. Babu, R. Angira, Modified differential evolution (MDE) for optimization of non-linear chemical processes. Computers and Chemical Engineering 30 (2006) 989–1002.

[22] S. Kheawhom, Efficient constraint handling scheme for differential evolutionary algorithm in solving chemical engineering optimization problem, Journal of Industrial and Engineering Chemistry, 16 (2010) 620–628.

[23] Floudas, C. A., & Pardalos, P. M. (1990). A collection of test problems for constrained global optimization algorithms. Lecture notes in computer science: vol. 455. Berlin, Germany: Springer.

[24] H. Izakian, A. Abraham, V. Snasel, Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments, The 2009 IEEE International Workshop on HPC and Grid Applications (IWHGA2009), IEEE Press, pp. 8-12, 2009.

[25] S. Das, A. Biswas, S. Dasgupta and A. Abraham, Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications, Foundations of Computational Intelligence Volume 3: Global Optimization, Studies in Computational Intelligence, Springer Verlag, Germany, ISBN: 978-3-642-01084-2, pp. 23-55, 2009.

[26] Hongbo Liu, Ajith Abraham and Maurice Clerc, Chaotic Dynamic Characteristics in Swarm Intelligence, Applied Soft Computing Journal, Elsevier Science, Volume 7, Issue 3, pp. 1019-1026, 2007.

[27] F. Xhafa and A. Abraham, Meta-heuristics for Grid Scheduling Problems, Metaheuristics for Scheduling: Distributed Computing Environments, Studies in Computational Intelligence, Springer Verlag, Germany, ISBN: 978-3-540-69260-7, pp. 1-37, 2008.

[28] R. Thangaraj, M. Pant, A. Abraham and P. Bouvry, Particle Swarm Optimization: Hybridization Perspectives and Experimental Illustrations, Applied Maths and Computation, Elsevier Science, Netherlands, Volume 217, No. 1, pp. 5208-5226, 2011.

[29] H. Liu, A. Abraham, O.K. Choi and S.H.Moon, Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-shop Scheduling Problems, The Sixth International Conference on Simulated Evolution And Learning (SEAL06), China, Springer Verlag, Germany, Lecture Notes in Computer Science, T.D.Wang et al. (Eds.): SEAL 2006, LNCS 4247, pp. 197-204, 2006.