Using Cooperation Approaches at Different Levels of Artificial Bee Colony Method

Vahid Zeighami, Mohsen Ghasemi, Reza Akbari

Abstract—In this work, a Multi-Level Artificial Bee Colony (called MLABC) for optimizing numerical test functions is presented. In MLABC, two species are used. The first species employs n colonies where each of them optimizes the complete solution vector. The cooperation between these colonies is carried out by exchanging information through a leader colony, which contains a set of elite bees. The second species uses a cooperative approach in which the complete solution vector is divided to k sub-vectors, and each of these sub-vectors is optimized by a colony. The cooperation between these colonies is carried out by compiling sub-vectors into the complete solution vector. Finally, the cooperation between two species is obtained by exchanging information. The proposed algorithm is tested on a set of well-known test functions. The results show that MLABC algorithm provides efficiency and robustness to solve numerical functions.

Keywords—Artificial bee colony, cooperative artificial bee colony, multilevel cooperation.

I. INTRODUCTION

THERE are many population based optimization techniques available for unconstrained numerical optimization. The algorithms which are inspired from intelligent behaviors of honey bees are among the newest optimization techniques which have been developed and applied on different engineering fields. In recent years, different types of bee algorithms have been presented in literature [1]-[3]. These algorithms are based on social-psychological principles and provide insights into social behaviors. Social influence and social learning are two major components of a social organism that enable individuals to maintain consistency. These components are based on interaction between individuals. Social influence encourages an individual to move toward another individual in a solution space. Social learning makes an individual to tune its behavior by observing the behavior of the other individuals.

Social influence and learning guarantee the success of a swarm in solving highly complex optimization problems. However, using only one population in these algorithms has disadvantage such as stagnation, premature convergence, low convergence speed, side effect, etc. To mitigate these disadvantages and improve the performance of the population

based algorithms, one can use other forms of social behaviors such as cooperation and competition. Cooperative individuals provide benefit to each other in the same species. Cooperation encourages individuals working together in order to obtain social improvement in their performance. In recent years, many cooperative optimization approaches have been proposed. The idea of using cooperation in population based algorithms was first introduced in GA by Potter and De Jong [4] for optimizing numerical functions by partitioning the solution vector into the two or more smaller vectors. After that, Van den Berg and Engelbrecht applied the Potter technique on standard PSO and proposed three models of cooperative PSO, called CPSO-S, CPSO-S_k, and CPSO-H_k [5], [6]. In CPSO-S_k method, the solution vector is divided to the k sub-vectors, and k swarms were used to optimize these sub-vectors concurrently. The CPSO-S method is a type of CPSO-S_k in which the complete D-dimensional solution vector is partitioned to the D one-dimensional vectors and finally CPSO-H_k combines standard PSO and CPSO-S_k. Similar to these cooperative variants of PSO, recently this idea has been applied on ABC algorithm and different cooperative ABC such as CABC [7], [8], CABC_S [9], and CABC_H [9] have been proposed.

Apart from sub-dividing the solution vector, different approaches for cooperative PSO such as MCPSO, CONPSO were presented in [10]-[12], where the solution vector is not partitioned, but multiple cooperative populations were employed. Usually, in these methods, cooperation is obtained through exchanging information about the global best individuals [13]. The idea of using multiple populations in bee algorithms was used by Akbari and Ziarati to design Cooperative bee algorithms [14].

In current work, a multi-level artificial bee colony algorithm based on cooperative behaviors in multiple levels of ABC is introduced. The proposed MLABC method can be seen as a multi-level optimization approach which optimize the numerical functions based on the social behaviors in three levels [15], [16].

The paper is organized as follows. Section II introduces the basic concepts of ABC algorithm. Description of the proposed MLABC algorithm is presented in Section III. Section IV reports experimental result of the proposed approach in comparison with the other variants of ABC. Finally, Section V concludes this work.

II. BASIC CONCEPTS OF ARTIFICIAL BEE COLONY

The ABC algorithm was introduced by Karaboga and Basturk in [1]. There are three types of bees in ABC. A bee

Vahid Zeighami is with Department of Mathematics and Industrial Engineering, Ecole Polytechnique, de Montreal, Montreal, Quebec, Canada (e-mail: vahid.zeighami@polymtl.ca).

Mohsen Ghsemi is with E-Learning College, Shiraz University, Shiraz, Iran, (e-mail: ghsemy39@gmail.com).

Reza Akbari is with Department of Computer Engineering and Information Technology, Shiraz University of Technology, Shiraz, Iran(e-mail: akbari@sutech.ac.ir).

waiting on the dance area for making decision to choose a food source is called onlooker; the bee going to the food source visited by herself just before is named as employed bee, and the bee who fly spontaneously is called scout. ABC employs these bees in three phases. At the initialization phase, ABC generates a randomly distributed initial population of *SN* solutions, where *SN* denotes the size of the employed or onlooker bees. Each solution presents a D-dimensional vector $x_i (i = 1,2,...,D)$, where *D* is the number of parameters to be optimized.

The second phase has three steps. At the first step, the employers come into the hive and share the nectar information of the food sources with the onlookers waiting on the dance area. After sharing their information with onlookers, every employed bee goes to the food source area visited by her at the previous iteration since that food source exists in her memory, and then chooses a new food source by means of visual information in the neighborhood of the one in her memory and evaluates its nectar amount.

At the second step, considering the information shared by employers in the dance area, an artificial onlooker bee chooses a food source based on the probability value associated with that food source, p_i, calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (1)$$

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 and *SN* is the number of food sources which is equal to the number of employed bees or onlooker bees. An onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. The probability of a food source selection increases as its nectar amount increases.

After an onlooker arrives at a promising area, she chooses a new food source in the neighborhood of the one in the memory depending on visual information as in the case of employed bees. The new food position is selected based on:

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

where $k \in \{1,2,...,SN\}$ and $j \in \{1,2,...,D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. ϕ_{ij} is a random number in range of [-1, 1].

At the third step, the nectar amount of the food sources are evaluated; the scout bees are determined and they are sent onto the possible new food sources. For this purpose, the positions are evaluated, and if a food source cannot be improved after a predetermined number of iterations (called limit), then the corresponding food source is abandoned. The limit parameter is determined manually. The abandoned food source is replaced with the new one founded by the scouts. A scout produces a new position randomly and replaces the abandoned food source if the new food source has better

nectar. Assume that the abandoned source is x_i and $j \in \{1,2,...,D\}$, then the scout discovers a new food source to be replaced with x_i. This operation can be defined as:

$$x_i^j = x_{\min}^j + rand[0,1](x_{\max}^j - x_{\min}^j) \qquad (3)$$

After each candidate source position v_{ij} is produced and evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the candidate one. The aforementioned processes are repeated through a predetermined number of iterations or until a termination criterion is satisfied.

III. Multi-Level Artificial Bee Colony

In some bee algorithms such as MP-ABC [8] and CABC_S [9], the colonies have the same structure and evolve in the same way. It is possible to embed the colonies in such algorithms in a larger community to create a species. This approach encourages us to develop a new optimization method, called MLABC, containing two species where the first and second species concurrently evolve under different approaches. In MLABC, cooperation carried out in three levels. From biological perspective, the selection of a level in which the cooperation is carried out between individuals has important role in success of individuals in a population. This idea which is called multilevel selection (MLS), proposed by Sober and Wilson [17] as a perspective to the cooperation concept. Multi-level selection approach suggests that cooperation occurs in more than one level [18]: for example, it may occur in an atomic and molecular level in cells, at the level of cells within an individual, and then again at level of individuals within a population, at the level of population within a species, and finally at the level of species. This approach encourages individuals to cooperate to avoid behaviors which favor themselves short-term, but destroy the community long term.

TABLE I
LEVELS OF COOPERATION AND CORRESPONDING BEHAVIORS

Level	Cooperative Behavior
Within colony	Selection
Between colonies	sharing information, dividing works
Between species	Exchanging information

According to the aforementioned levels of cooperation, the evolving process of an individual in MLABC depends on a set of operations which are performed at three levels: within colonies, between colonies (or within species), and between species. Hence, MLABC is a multi-level cooperative approach that uses two species each of which containing a different type of cooperative approach. In a multilevel cooperative approach, each level may provide different behaviors for the individuals

for contributing in problem solving. The possible behaviors at each level are presented in Table I.

The optimization process starts by applying flying patterns within each of the colonies. The results of the flying patterns are processed and the selected results are passed to the second level of cooperation. The cooperation in this level is obtained through selection and the types of actions which are performed by different types of bees. The selection is a behavior used by onlooker and employed bees. The onlooker bees select their interesting employed bee and follow them to find new food sources. Also, this behavior is used by employed bees to select the elite food sources. The type of individual has important role in cooperation. One well accepted theory for why cooperative behaviors occur between individuals with different types is the theory of kin selection. This theory suggests that individuals act cooperatively in order to help others which are genetically similar.

At the second level, cooperation is carried out between different colonies of one species. In the first species, the cooperation between colonies is carried out using the approach proposed by MC-CABC module, while the colonies in the second species cooperate through the approach proposed by S-CABC module. The first species shares information through the leader colony in order to establish cooperation between its colonies. In the second species, the work is divided to the smaller part, and each colony responsible to accomplish a part of the work. The colonies share the information to complete the work.

At the third level, cooperation is carried out by exchanging the elite bees between species. In the first species, cooperative colonies result a set of fittest bees which are maintained in the leader colony. Also, the cooperative colonies in the second species result a set of fittest solutions which are formed by concatenating the partial solutions provided by each of that colonies. These solutions are maintained in C_{elite}. The MLABC algorithm uses the information provided by these species to carry out cooperation between them. The cooperation in the level of species is carried out through two exchange channels S2M and M2S. In MLABC algorithm, two species concurrently optimize an objective function. After predefined number of iterations ($iter_{ex}$), the information is exchanged using $\text{M2SEx}(Sp_1, Sp_2)$ and $\text{S2MEx}(Sp_2, Sp_1)$ functions. The $\text{M2SEx}(Sp_1, Sp_2)$ function chooses randomly k_1 bees from the leader colony $C_{lead} \in Sp_1$ and overwrites k_1 randomly selected bees from the colonies of species Sp_2. The solution vector in Sp_2 is split by the split factor sf_2, and each of the colonies of the Sp_2 optimizes one component of the solution vector. Hence, the $\text{M2SEx}(Sp_1, Sp_2)$ function should split the position vector of a chosen bee by the split factor sf_2.

A reverse process in exchanging information from Sp_2 to Sp_1 species is done by $\text{S2MEx}(Sp_2, Sp_1)$ function. Unlike Sp_2, each of the colonies in Sp_1 works on the complete solution vectors. Hence, we need to compile the partial solutions produced by the colonies of Sp_2 in complete solution vectors. The process starts by constructing complete solution vectors

from partial solutions. As described previously, by considering two best experienced employed bees from each of the colonies $S_i \in Sp_2$ we can construct 2^{sf_2} candidate position vectors. These complete solution vectors constitute the elite colonies (S_{elite}). After that, the k_2 best solution vectors will be selected from S_{elite}. The $\text{S2MEx}(Sp_2, Sp_1)$ function overwrites k_2 randomly selected bees in the colonies of the species Sp_1 at the exchange time.

Initialize Sp_1, and Sp_2
Do
    Execute one iteration of S-CABC
    Execute one iteration of MC-CABC
    If (*iter* mode $iter_{ex}$ = 0) **Then**
        Update S_{elite} in Sp_2
        Do $\text{M2SEx}(Sp_1, Sp_2)$
        Do $\text{S2MEx}(Sp_2, Sp_1)$
Until termination condition is met

Fig. 1 Pseudo code of the MLABC algorithm

Fig. 1 presents the MLABC in pseudo code. The algorithm starts by initializing the species. At initial step, the number of colonies in each species is defined and the colonies are initiated as the ways described in sub-sections A and B. Also, the new parameters k_1, k_2, and $iter_{ex}$ are defined. After that, at each cycle of the algorithm, the position vectors in Sp_1 are updated by executing one iteration of MLABC, as well as the position vectors in Sp_2 are updated by executing an iteration of S-CABC. The information is exchanged after $iter_{ex}$ iterations. At the exchange time, the algorithm first updates the elite colony in Sp_2 by the complete solution vectors which are formed by concatenating two elite bees from each swarm. Second, the information is exchanged through $\text{M2SEx}(Sp_1, Sp_2)$ and $\text{S2MEx}(Sp_2, Sp_1)$ channels.

The multilevel cooperative variant of the ABC algorithm has the ability to solve any problem that ABC algorithm can solve it. However, its performances are affected by different factors. Beyond the factors described in Section I, there are other factors that affect the performance of the proposed multilevel cooperative optimization approach. The type of exchanged or shared information as well as the amount of that information have important role on performance of the algorithms. Using a large amount of information, the number of replaced individuals is increased. This may influence the position update mechanism in the algorithm.

Another factor that affects the performance of the multilevel cooperative optimization algorithm is the time points at which the information are exchanged or shared. In MLABC algorithm, this time point is controlled by $iter_{ex}$. The small value for $iter_{ex}$ increases the replacement of the individuals throughout iterations. This decreases the diversity of algorithm and the premature convergence may occur. At the other hand, the large $iter_{ex}$ decreases the beneficial effects of employing multiple species. Hence, the proper exchange time should be used.

A. Multi Colony Cooperative ABC (MC-CABC) Module

The MC-CABC module is used by the first species. The MC-CABC employs n colonies similar to the work presented in [15]. The colonies contain the same number of bees. The complete solution space is u sed by each of the colonies, and the colonies concurrently optimize the problem. MC-CABC algorithm evaluates fitness of each bee i in a colony j independently without considering other bees. The cooperation approach is based on sharing information between colonies. The MC-CABC shares information between colonies through a community of fittest bees.

Given a D-dimensional optimization problem, each of the n colonies concurrently optimizes all the D parameters in the solution space S. In an ideal way, a bee in one colony needs to cooperate with all the bees in other colonies in order to extend its capabilities. Such a way establishes exhaustive cooperation which needs high computational effort and increases the complexity of the a lgorithm. So, we need alternative approaches for cooperation between colonies. As a one way to achieve this goal, we introduce a further colony C_{lead}, called leader colony, which contains a set χ of the fittest bees from all the colonies. The l eader colony acts as community for sharing information between colonies. This co mmunity consists of the best bee from each of the colonies.

Module MC-CABC (*n, Max_Iter, limit, h*)
For j=1 to n
  Initialize the population Pop$_j$ of solutions $x_{j,i}, i=1,2,...,h$

  Evaluate population Pop$_j$
End For
For *iter*=1 to *Max_Iter*
  For j=1 to n
    Step 1) Produce new solutions $v_{j,i}$ for the employed bees using (3) and
        evaluate them
      Apply the greedy selection process for the employed bees
    Step 2) Calculate the probability values $P_{j,i}$ for the solutions $x_{j,i}$ by (2)
      Produce the new solutions $v_{j,i}$ for the onlookers from the
      solutions $x_{j,i}$ selected depending on $P_{j,i}$ and evaluate them
      Apply the greedy selection process for the onlookers
    Step 3) Determine the abandoned solution for the scout, if exists, and
        replace it with a new randomly produced solution $x_{j,i}$ by (4)
      Memorize the best solution achieved so far
  End For
  Step1) Select best solutions b_j from all the populations Pop$_j$, $1 \le j \le n$

  Step2) For j=1 to n
    Produce new solutions $v_{j,i}$ for the employed bees using (5) and
    evaluate them
    Apply the greedy selection process for the employed bees
    End For
End For
Return best solution

Fig. 2 Pseudo code of the MC-CABC algorithm

Fig. 2 p resents the MC-CABC algorithm in pseudo code. ABC method is extended to encompass cooperation operators. The position of bee i in colony j is represented by $\vec{x}_{j,i}$ that its fitness is evaluated independently. The onlooker and scout bees use only local information available in their own colony,

and their flying patterns are the same as the p atterns in ABC algorithm.

Unlike onlooker and scout bees, the employed bees in MC-CABC use a n extended ver sion of their flying patterns proposed in ABC in order to model information sharing between colonies. At each cycle of the algorithm, the $lead(C_j, \chi)$ returns the best bee from colony C_j and updates j-th position vector in the leader colony by the positions of this bee. The position of the j bee in the leader colony is updated if the best bee in colony j has better fitness.

Considering the leader colony, an e mployed bee uses the shared information to adjust its movement trajectory in the next time. The bee evaluates the provided information by the leader colony C_{lead}. The flying trajectory of an employed bee i of the colony j is controlled using:

$$\vec{v}_{j,i} = \vec{x}_{j,i} + \sum_{m=1}^{n} \vec{\emptyset}_{j,i}(\vec{x}_{j,i} - \vec{\chi}_{j,i}) \qquad (4)$$

where $\vec{\chi}_{j,i}$ is th e position vector of the lea der bee which is selected by the employed bee i in the colony j and $\vec{\emptyset}_{j,i}$ controls the importance of the shared information.

Module S-ABC(*Max_Iter, limit, SN, D,k*)
$D_1 = \lfloor D/k \rfloor$, $D_2 = D - (D_1 \times k)$
Initialize $k-1$ population of solutions $x_i, i=1,2,...,SN$ with D_1 dim
Initialize 1 population of solutions $x_i, i=1,2,...,SN$ with D_2 dim
  For each population P_j, $j \in \{1,2,...,k\}$
    For each bee $b_i \in P_j$
      Construct the context vector using $CV(i,j)$
    End For
    Evaluate the population j
  End For

For *iter*=1 **to** *Max_Iter*
  For each population P_j, $j \in \{1,2,...,k\}$
    Step 1) Produce new solutions t_i for the employed bees by using (2)
      Construct the context vector using $CV(t_i, j)$ and evaluate
      them
      Apply the greedy selection process for the employed bees

    Step 2) Calculate the probability values P_i for the solutions x_i by (1)
      Produce the new solutions t_i for the onlookers from the
      solutions x_i selected depending on P_i
      Construct the context vector using $CV(t_i, j)$ and evaluate
      them
      Apply the greedy selection process for the onlookers

    Step 3) Determine the abandoned solution, if exists, and replace it
      with a new randomly produced solution x_i by (3)
      Construct the context vector using $CV(t_i, j)$ and evaluate it
  End For
  Memorize the best solution achieved so far
End For
Return best solution

Fig. 3 Pseudo code of the CABC algorithm

B. Split Cooperative ABC (S-CABC) Module

The S-CABC is another cooperative version of ABC which is used in the second species. S-CABC is similar to the other cooperative variants of ABC presented in [7]-[9]. S-CABC

mainly focuses on p artitioning the search space. In t his module, the solution vector is partitioned to the k components. After that, a set of k population are employed to cooperatively optimize these components. The pseudo code of the S-CABC module is presented in Fig. 3.

In this approach, each population only has local information about the solution. The global information is achieved by compiling the local information for constructing the complete solution vectors. The complete solution is constituted by concatenating partial solutions from different populations and the fitness function is ca lculated based on the complete solutions. Similar to CPSO-S_k [6], a context vector is used to establish cooperation between populations. In this way, a context vector is con structed by taking the global best individual from each population and assembling them in to the complete solution vector. The fitness of an individual i in population j is calculated by fixing the other k-1 components of the context vector to their corr esponding global best individuals in populations, and replacing the j-th component by individual i. Although this approach ignores valuable information that may obtained through concatenation of non-global best individuals from each of the k populations, the computational complexity forces the al gorithm to avoid considering the possible combination of partial solutions. For implementation, we used function $CV(i, j)$ which returns a context vector whose components are filled with the global best solutions from all the populations except the j-th component which is filled by i-th solution vector of the j-th population.

IV. EXPERIMENTS

To test the performance of MLABC in comparison with the other variants of ABC, five well known benchmark functions are used her e. These benchmarks are widely used in evaluating performance of population based methods. The first two functions are u nimodal and others are multimodal. A unimodal function has only one optimum. A fu nction called multimodal if it has two or more local optima.

The performance of the new method is c ompared with the performance of the ABC, S-CABC, and MC-CABC. All experiments were run for 30 dimensional test functions. The number of evaluations was chosen at 5000 00. For all the test functions, the population size is set to 200, and a total of 30 runs for each experi mental setting are conducted. The following sentences describe t he specific parameter settings for each of the aforementioned algorithms. In ABC algorithm, the trial factor is set at 100. The solution vector for S-CABC algorithm is split into five parts. All the other parameters have equal settings with the parameters of the ABC. In MC-CABC five colony (n=5) concurrently optimize the test f unctions. Each colony optimizes the complete solution vector. Two species in the MLABC method exchange information at every 5 iterations ($iter_{ex} = 5$). The Sp_1 employing MC-CABC with n=5, while the Sp_2 employing S-CABC with k=6. The exchange parameters k_1 and k_2 are respectively set at 2 and 5. So, after five iterations, one of the bees in each of the colonies

of Sp_1 is replaced by the selected elite bees. Also, two randomly selected bees in th e swarms of Sp2 are replaced by the elite bees from C_{lead}.

A. Experimental Results

Table II presents three measures (mean, standard deviation, and average number of evaluations before convergence) for four algorithms on the five test functions. Also Fig. 4 s hows the evolution of the four variants of ABC over the five test functions. For Rosenbrock test function, the success criterion is set at 100 and for the others is set at 0.01. After the final evaluation, if the minimum value was reached by the algorithm was not below the success c riteria, the run was considered unsuccessful.

According to the results reported in Table II and convergence behavior in Fig. 4, it can be seen that, the b est result on the Sphere function is obtained by MLABC a nd it has the fastest convergence speed. The Rosenbrock function is a hard problem to solve. Table II shows the MC-CABC has the best p erformance. However, the MLABC has the fastest convergence speed. The Rast rigrin function is a h ighly multimodal with f requent local optima. An algorithm with poor balance between exploration and ex ploitation simply trapped in local optima in early iterations. Table II presents evaluation results for the Rastrigrin function. The best result is obtained by the MC-CABC and MLABC al gorithms. Also, from Table II, it is apparent that the MLABC has fast speed in converging to the success criteria.

As can be see n from Table II, MC-CABC and MLABC algorithms outperform other algorithms in optimizing Griewank function. MLABC is the fastest algorithm in converging to the success criteria. The best result on the Ackley is o btained by MLABC and it has t he fastest convergence speed.

TABLE II
EVALUATION RESULTS FOR THE FIVE TEST FUNCTIONS

| Func.	Method	Mean(Stdv.)	#evaluations
Sph.	ABC	3.21E-016 ± (5.02E-017)	5.34E+004
	MC-CABC	2.01E-031 ± (2.67E-030)	3.04E+004
	S-CABC	1.15E-016 ± (1.16E-017)	2.26E+004
	MLABC	**4.71E-045 ± (7.73E-046)**	**9.00E+003**
Ros.	ABC	6.42E-001 ± (7.03E-001)	8.42E+004
	MC-CABC	**4.13E-005 ± (4.98E-005)**	2.28E+004
	S-CABC	1.41E+001 ± (2.26E+001)	1.72E+004
	MLABC	4.24E-003 ± (9.97E-003)	**7.00E+003**
Ras.	ABC	7.70E-016 ± (1.85E-015)	2.34E+005
	MC-CABC	**0.00E+000 ± (0.00E+000)**	1.37E+005
	S-CABC	8.48E-018 ± (1.65E-018)	4.34E+004
	MLABC	**0.00E+000 ± (0.00E+000)**	**3.48E+004**
Grie.	ABC	3.03E-016 ± (6.99E-017)	4.54E+004
	MC-CABC	**0.00E+000 ± (0.00E+000)**	8.28E+004
	S-CABC	1.04E-016 ± (1.28E-017)	1.64E+004
	MLABC	**0.00E+000 ± (0.00E+000)**	1448.29
Ack.	ABC	4.45E-014 ± (3.60E-015)	1.20E+005
	MC-CABC	1.20E-013 ± (3.44E-014)	1.21E+005
	S-CABC	1.76E-014 ± (3.53E-015)	3.04E+004
	MLABC	**4.45E-015 ± (3.27E-015)**	**1.68E+004**

In general, the results showed that cooperation in more than one level has positive effect on th e performance of the standard ABC and competitive performance can be ob tained by employing this type of optimization manner.
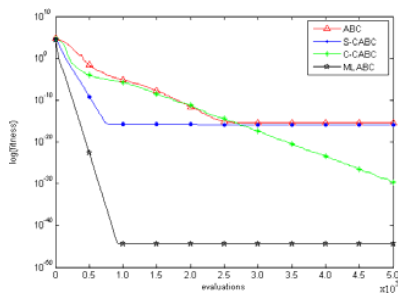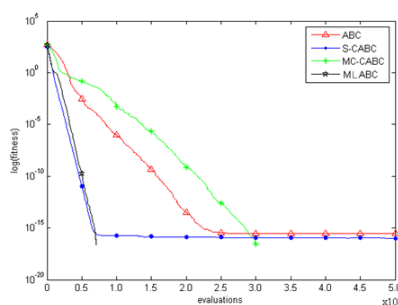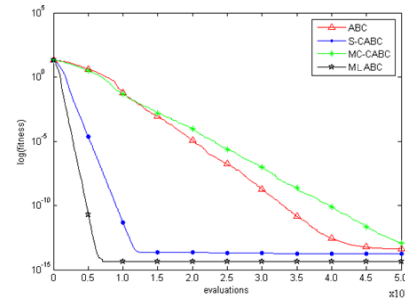

(a) Sphere
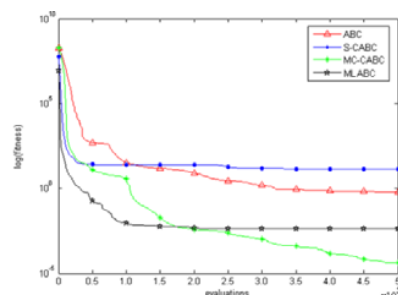
(b) Rosenbrock
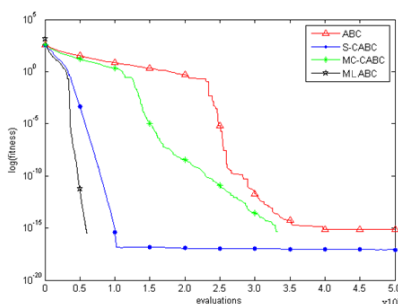
(c) Rastrigrin

(d) Griewank

(e) Ackley

Fig. 4 Evolution of average fitness for the five test functions.

V. CONCLUSIONS

The optimization algorithms which are inspired from intelligent behavior of honey bees are among the most recently introduced population based algorithms. In this paper, we have described a multi-level ABC algorithm which incorporates two different types of cooperative approach to improve its performance. The MLABC was formed in a multi-level approach in which the cooperation is carried out in more than one level. The results showed that, optimizing under this way is efficient and better performance could be obtained.

REFERENCES

[1] D. Karaboga, B. B asturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", in *Journal of Global Optimization* vol. 39, pp. 459–471, 2007.
[2] R. Akbari, A. Mohammadi, K. Ziarati, A powerful bee swarm optimization algorithm, in: *Proceedings of the IEEE 13th Multitopic Conference*, 2009, pp. 1–6.
[3] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The bees algorithm—a novel tool f or complex optimisation problems, Proceedings of the Intelligent Production Machines and Systems (2006) 454–459.
[4] M. A. Potter and K. A. de Jong, "A cooperative coevolutionary approach to function optimization," in *the Third Parallel Problem Solving from Nature*. Berlin, Germany: Springer-Verlag, 1994, pp. 249–257.
[5] F. van den Bergh, and A . P. Enge lbrecht, "Cooperative learning in neural networks using particle swarm optimizers," *South African Comput. J.*, vol. 26, pp. 84–90, 2000.
[6] F. van den Bergh, and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization", in *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 225-239, 2004.
[7] W. Zou, Y. Zhu, H. Che n, and Z. Zhu, Cooperative Approaches to Artificial Bee Colony Algorithm, 2010 *International Conference on Computer Application and System Modeling (ICCASM 2010)*, Vol. 9, pp. 44-48 2010.
[8] R. Akbari, V. Zeighami, K. Ziarati, A Cooperative Artificial Bee Colony Optimizer, 7th *International Industrial Engineering Conference*, Isfahan, Iran, pp. 1-10, 2010.
[9] M. El-Abd, A Cooperative Approach to The Artificial Bee Colony Algorithm, *IEEE World Congress on Evolutionary Computation*, 2010.
[10] S. Baskar and P. N. Suganthan, "A novel concurrent particle swarm optimization," in *Proc. of IEEE Congress on Evolutionary Computation*, vol.1, pp. 792– 796, 2004.
[11] M. Belal and T. El-Ghazawi, "Parallel models for particle swarm optimizers," in *International Journal on Intelligent Cooperative Information Systems*, vol. 4, no. 1, pp. 100–111, 2004.
[12] Ben Niu, Y. Zhu, X. He, H. Wu, "MCPSO: A multi-swarm cooperative particle swarm optimizer", *In Journal of Applied Mathematics and Computation*, 185, pp. 1050–1062, 2007.
[13] M. El-Abd, and M. S. Kamel, " A Taxonomy of Cooperative Particle Swarm Optimizers", in *International Journal of Computational Intelligence Research*, Vol.4, No.2, pp.137–144, 2008.

[14] R. Akbari, and K. Ziarati, A Cooperative Approach to Bee Swarm Optimization, *Journal of Information Science and Engineering*, Vol. 27, No. 3, pp. 799-818, 2011.

[15] V. Zeighami, R. Akbari, and K. Ziarati, "An Efficient Multi Population Artificial Bee Colony," International Journal of Machine Learning and Computing vol. 2, no. 3, pp. 195-199, 2012.

[16] R. Akbari, V. Zeighami, and K. Ziarati, "MLGA: A Multilevel Cooperative Genetic Algorithm", *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2010)*, pp. 271-277, September 2010.

[17] E. Sober, and D. S. Wilson, "Unto others, the evolution and psychology of unselfish behavior", *Harvard University Press*, 1998.

[18] R. Bergmuller, A. F. Russell, R. A. Johnstone, R. Bshary, "On the further integration of cooperative breeding and cooperation theory", in *Journal of Behavioural Processes*, 76 (2007), pp. 170–181.