A New Hybrid Model of PSO and ABC Algorithms for Optimization in Dynamic Environment

Noosheen Baktash and Mohammad Reza Meybodi

Abstract—Many optimization problems in real world are dynamic in the sense that the global optimum value and the shape of fitness function may change with time. The task for the optimization algorithm in these environments is to find global optima quickly after the change in environment is detected. In this paper we describe a novel algorithm, which we have called DPSABC, and show that it can be applied to dynamic optimization problems. The core of this algorithm is using PSO to optimize the fitness value of population in ABC. Experimental results on various dynamic environments modeled by the moving peaks benchmark show that the proposed algorithm outperforms other algorithms, like mQSO, adaptive mQSO and RPSO.

***Index Terms*—Artificial Bee Colony, Particle Swarm Optimization, Dynamic Environment.**

I. Introduction

In the last two decades, the computational researchers have been increasingly interested to the natural sciences, and especially biology, as source of modeling paradigms. Many research areas are massively influenced by the behavior of various biological entities and phenomena. It gave birth to most of population-based let heuristics such as Evolutionary Algorithms (EAs), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) etc. Honey bees are one of the most well studied social insects. Both of the algorithms are co-operative, population-based global search swarm intelligence met heuristics.

The paper is organized as follows: Sections 2 and 3 give a brief description about basic Artificial Bee Colony and Particle Swarm Optimization. Section 4 describes the proposed Algorithm. Section 5 gives out the experimental results of the proposed model along with its comparison to best results gained in previous works. Finally in section 6 we conclude our paper.

II. ABC Algorithm

The Artificial Bee Colony algorithm has been proposed by Karaboga [1]. It is a population based algorithm.[2] The colony consists of three groups of bees: employed bees, onlookers and scouts. A possible solution to the optimization problem is represented as the position of a food source and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees is equal to the number of solutions in the population. At the first step, a randomly distributed initial population (food source positions) is generated. An employed bee produces a modification on the source position in its memory and evaluates the fitness at that position (calculates the nectar amount). If the fitness (nectar amount) of the new position is higher than that of the previous position, the bee memorizes the new source position and forgets the old one; otherwise it keeps the position of the old one in memory. After all the employed bees have evaluated the new positions, the onlookers go to these positions with more onlookers going towards better positions and less onlookers going towards less fit positions. The onlookers also produce a modification on that position and evaluate the fitness at that position. The scouts randomly select positions to evaluate. This cycle continues until the termination criteria is meet. Also if the fitness of certain employed bee does not improve for some time then that employed bee is converted to a scout bee [3].

Manuscript received February 20, 2012; revised April 30, 2012.

Noosheen Baktash is with the Department of Electrical and Computer Qazvin Branch Islamic Azad University, Qazvin, Iran

Mohammad Reza Meybodi is with the Department of Computer Engineering and Information Technology at Amirkabir University of Technology, Tehran, Iran.

III. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based computational technique inspired from the simulation of social behavior of flock of birds. PSO was originally designed and developed by Eberhart and Kennedy [2]. In PSO a swarm of particle is used to represent the population of candidate solutions. Each particle is a point in the N-dimensional search space. A particle is represented by its current position, and its current velocity. PSO tries to find the optimal solution to the problem by moving the particles and evaluating the fitness of the new position [3].

for each Particle i ∈ [1 .. N]
initialize xi , vi
Pi = xi
End for
$G = \arg\min_{P_i} f(P_i)$
repeat:
for each Particle i ∈ [1 .. N]
update vi using equation (5)
Check the velocity boundaries.
update xi using equation (6)
if f(xi) ≤ f(Pi) then Pi=xi
if f(Pi) < f(G) then G=Pi
end for
until stopping criterion is met

Fig. 2. Pseudo code of PSO.

IV. Proposed Model

This section discusses the infrastructure of the hybrid

algorithm. In Initialization, constant and variable are determined. The colony size, the number of bees, the learning constant, including c1, c2 and x in PSO, should be assigned in advanced. In PSABC, ABC and PSO both work with same population. In this algorithm the population is randomly generated at first. These may be regarded as bee colony in term of ABC, or as particles in terms of PSO. Then, in each cycle, after the fitness of all bees in same population is calculated, the employed bees are mark and we enhance them by PSO. The global best particle of the population is determined according to the sorted fitness values. By performing PSO solution on the employed bee, we increase the search ability.

Initialize the PSABC
Generate the initial population zi, i=1,…,Sn
Select half part of bees as employed bee with PSO
Evaluate the fitness (fi) of the population
Set cycle to 1
repeat
For each employed bee Do
Produce new solution vi
Calculate the value fi
Apply greedy selection process
Calculate the probability values pi for the solutions (zi)
For each onlooker bee
Select a solution zi depending on pi
Produce new solution vi
Calculate the values fi
Apply greedy selection process
If there is an abandoned solution for the scout Then
›lace it with a new solution which will be randomly produced
Memorize the best solution so far
cycle = cycle + 1
until cycle=MCN

Fig. 1. Pseudo code of PSABC.

V. Experimental Result

A. Dynamic Test Function

We used moving peaks benchmark (MPB) problem introduced by Branke [4] to evaluate performance of our proposed model. In this problem, there are some peaks in a multi-dimensional space, where the height, width and position of each peak alter when the environment change. B (x) varies with time and is the basis of prospects and p is a function that determines the peak. Each m peak has parameters changing with time including height h, width w and location p ; after Δe assessment, height, width and location of each peak according to [4], [5] will be changed by adding variable Gvsyn random and location of each peak by fixed-length s vector v will be moved and s will permit change intensity of control. If the V vector is generated dependant on the previous change, change of the peak position will be in line with the previous changes and if it is generated randomly, it will be changed randomly. For evaluating the performance of algorithm we have used an error output line 10criterion [6], which is the average deviation from optimal in all repetitions and according to [5] which defines the fk is the best solution for algorithm, just before the k th environment change, hk is the best value of k th environment state and k is the total number of environments.

$$f(\vec{x},t)=\max(B(\vec{x}),\max P(\vec{x},\boldsymbol{h}_i(t),\boldsymbol{w}_i(t),\vec{\boldsymbol{p}}_i(t))) \quad (1)$$

$$i=1...m$$

$$\begin{aligned}&\sigma\in N(0,1)\\&h_i(t)=h_i(t-1)+height\_severity.\sigma\\&w_i(t)=w_i(t-1)+width\_severity.\sigma\\&\vec{p}_i(t)=p_i(t-1)+\vec{v}_i(t)\end{aligned} \quad (2)$$

$$\mu=\frac{1}{k}\sum_{i=1}^{k}\left(h_f-f_k\right) \quad (3)$$

The default parameter setting of MPB used in the experiments is presented in Table I. In MPB, shift length s is the radius of peak movement after an environment change. M is the number of peaks. f is the frequency of environment change as number of fitness evaluations. H and W denote range of height and width of peaks which will change after a change in environment by height severity and width severity respectively. I is the initial heights for all peaks. Parameter A denotes minimum and maximum value on all dimensions.[7]

TABLE I: Parameters of MPB

Parameter	*Value*
number of peaks m 10	10
f every	5000 evaluations
height severity	7
width severity	1
peak shape	Cone
shift length s	1
number of dimensions D	5
Population size	100
Domain	[0, 100]
H	[30.0, 70.0]
W	[1, 5]
Initial height	50

B. Experimental Setting and Results

Here we present comparison results of our proposed with best reported models. In order to evaluate the efficiency of the algorithms, we have used offline error measure, the average deviation of the currently best individual from the optimum in all iterations. Table II through 6 present offline error results of algorithms for different frequency of environment change (f).

TABLE II: Offline Error for F=500

Peak count	DPSABC	Adaptive MQSO	Adaptive CPSO	MQSO 10(10+10q)[6]	MQSO 10(5+5q)[8]	RPSO
1	2.77E+00	1.41E+00	1.22E+01	4.45E+01	3.37E+01	5.20E+00
10	3.42E+00	9.42E+00	9.18E+00	1.52E+01	9.62E+00	1.81E+01
20	3.12E+00	9.59E+00	8.78E+00	1.30E+01	9.07E+00	1.78E+01
30	3.69E+00	9.44E+00	8.64E+00	1.25E+01	8.80E+00	1.74E+01
40	4.16E+00	9.44E+00	8.68E+00	1.22E+01	8.55E+00	1.68E+01
50	3.22E+00	9.13E+00	8.49E+00	1.22E+01	8.72E+00	1.64E+01
100	3.01E+00	8.81E+00	8.22E+00	1.15E+01	8.54E+00	1.50E+01
200	3.16E+00	8.21E+00	7.89E+00	1.13E+01	8.19E+00	1.40E+01

TABLE III: Offline Error for F=1000

Peak count	DPSABC	Adaptive MQSO	Adaptive CPSO	MQSO 10(10+10q)	MQSO 10(5+5q)	RPSO

1	1.68E+00	6.60E+00	5.83E+00	2.17E+01	1.86E+01	2.40E+00
10	3.23E+00	5.64E+00	5.29E+00	7.66E+00	5.71E+00	1.58E+01
20	3.40E+00	5.95E+00	5.49E+00	7.25E+00	5.85E+00	1.56E+01
30	3.28E+00	5.97E+00	5.42E+00	7.19E+00	5.81E+00	1.49E+01
40	3.11E+00	6.12E+00	5.32E+00	7.19E+00	5.70E+00	1.45E+01
50	2.67E+00	3.56E+00	3.30E+00	4.36E+00	3.63E+00	1.20E+01
100	3.08E+00	5.78E+00	2.08E+00	6.94E+00	5.83E+00	1.25E+01
200	3.01E+00	5.54E+00	2.00E+00	6.98E+00	5.54E+00	1.16E+01

TABLE IV: OFFLINE ERROR FOR F=2500

Peak count	DPSABC	Adaptive MQSO	Adaptive CPSO	MQSO 10(10+10q)	MQSO 10(5+5q)	RPSO
1	1.20E-02	2.48E+00	2.00E+00	9.88E+00	7.64E+00	1.00E+00
10	1.76E-01	2.91E+00	3.03E+00	4.38E+00	3.12E+00	1.37E+01
20	7.91E-01	3.40E+00	3.17E+00	4.34E+00	3.58E+00	1.39E+01
30	2.23E+00	3.47E+00	3.22E+00	4.36E+00	3.63E+00	1.30E+01
40	2.31E+00	3.56E+00	3.32E+00	4.37E+00	3.00E+00	1.25E+01
50	3.12E+00	5.98E+00	5.25E+00	7.14E+00	5.87E+00	1.42E+01
100	3.02E+00	3.53E+00	3.35E+00	4.21E+00	3.58E+00	1.04E+01
200	3.14E+00	3.37E+00	3.29E+00	4.04E+00	3.30E+00	9.63E+00

TABLE V: OFFLINE ERROR FOR F=5000

Peak count	DPSABC	Adaptive MQSO	Adaptive CPSO	MQSO 10(10+10q)	MQSO 10(5+5q)	RPSO
1	2.25E+00	1.09E+00	8.70E+00	5.17E+00	3.82E+00	5.60E+01
10	2.13E+00	1.85E+00	1.91E+00	2.81E+00	1.91E+00	1.30E+01
20	2.07E+00	2.18E+00	2.26E+00	3.22E+00	2.56E+00	1.28E+01
30	1.88E+00	2.36E+00	2.25E+00	3.29E+00	2.68E+00	1.24E+01
40	1.99E+00	2.42E+00	2.41E+00	3.24E+00	2.65E+00	1.14E+01
50	1.91E+00	2.53E+00	2.43E+00	3.27E+00	2.63E+00	1.13E+01
100	1.89E+00	2.50E+00	2.53E+00	3.08E+00	2.52E+00	9.73E+01
200	1.87E+00	2.36E+00	2.46E+00	2.89E+00	2.36E+00	8.90E+01

TABLE VI: OFFLINE ERROR FOR F=10000

Peak count	DPSABC	Adaptive MQSO	Adaptive CPSO	MQSO 10(10+10q)	MQSO 10(5+5q)	RPSO
1	2.67E+00	4.90E-01	4.00E-01	3.56E+00	1.90E+00	5.30E-01
10	9.00E-01	1.10E+00	1.49E+00	2.34E+00	1.10E+00	1.26E+01
20	6.60E-01	1.50E+00	1.57E+00	2.82E+00	1.84E+00	1.26E+01
30	7.70E-01	1.64E+00	1.77E+00	2.91E+00	2.00E+00	1.16E+01
40	7.90E-01	1.70E+00	1.86E+00	2.85E+00	1.99E+00	1.10E+01
50	8.10E-01	1.71E+00	1.91E+00	2.78E+00	1.99E+00	1.06E+01
100	8.34E-01	1.75E+00	1.98E+00	2.48E+00	1.85E+00	9.37E+00
200	8.52E-01	1.66E+00	2.05E+00	2.32E+00	1.71E+00	8.30E+00

VI. CONCLUSION

In this work, a new optimization algorithm based on the intelligent behavior of honey bee swarm and PSO has been described. The new swarm algorithm is very simple and very flexible when compared to the existing swarm based algorithms. It is also very robust, at least for the test problems considered in this work. From the simulation results, it is concluded that the proposed algorithm can be used for solving dynamic optimization problems. In this work, the algorithm was tested on a very limited set of test problems.

REFERENCES

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

[2] J. Kennedy and R. C. Eberhart, "*Particle Swarm Optimization*," in *IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995, pp. 1942-1948,

[3] M. Rashid and A. R. Baig, "Adaptable Evolutionary Particle Swarm Optimization," in *Proceeding of the 2008 3rd International Conference on Innovative Computing Information and Control*,(ICICIC 2008), Dalian, China, June 18 - 20, 2008, pp.5-16.

[4] J. Branke "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems," in *1999 Congress on Evolutionary Computation*, Washington D.C., USA, 1999, pp. 1875-1882.

[5] I. Moser, "All Currently Known Publications on Approaches Which Solve the Moving Peaks Problem," *Swinburne University of Technology*, Melbourne, Australia, 2007.

[6] T. Blackwell and J. Branke, "Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments," *IEEE Transactions on Evolutionary Computation*, pp. 459-472, 2006.

[7] A. Baradaran Hashemi and M. R. Meybodi, "A Multi-Role Cellular PSO for Dynamic Environments," *Proceedings of the 14th International CSI Computer Conference (CSICC'09)*, Amirkabir University of Technology, Tehran, Iran, pp. 412-417, October 20-21, 2009.

Noosheen Baktash received him B.S. degree from the Department of Computer Engineering at science and Technology University, in 2006. Currently; she is pursuing her M.S. degree in the Department of Electrical and Computer Qazvin University. Her research areas Include Swarm intelligence.

Mohammad Reza Meybodi received his B.S. degree from the National University of Iran, in 1974. the M.S. degree in Computer Science from University of Oklahoma, Norman, OK, USA,in 1980 and his Ph.D. degree in Computer Science from University of Oklahoma, Norman, OK, USA, in 1983 . Dr. Meybodi is currently a faculty member at the Faculty of Computer Engineering and Information Technology at AmirKAbir University. His research has focused on Parallel Algorithms, Learning Systems, Software Engineering, Soft Computing, Neural Networks, Grid Computing and Ad Hoc Networks.