

Cellular PSO-ABC: A New Hybrid Model for Dynamic Environment

Noosheen Baktash, Fariborz Mahmoudi, and Mohammad Reza Meybodi

Abstract—Most research in evolutionary computation focuses on optimization of static, non-changing problems. Many real-world optimization problems, however, are dynamic, and optimization methods are needed that are capable of continuously adapting the solution to a changing environment. In this paper we describe a novel algorithm, which we have called Cellular-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.

Cellular automata make up of cells like points in a lattice or like squares of checker boards and it follows a simple rule. Colonies are distributed randomly among the cells of the cellular automaton that each colony is allocated to one cell. The cells exchange their best solutions to the others in greedy manner and with this strategy they try to find the best solution in the dynamic environment. Experimental results on various dynamic environments modeled by the moving peaks benchmark show that the proposed algorithm outperforms other algorithms, like CABC, CPSO, mQSO, adaptive mQSO and RPSO.

Index Terms—Artificial Bee Colony, Particle Swarm Optimization, Dynamic Environment, Cellular Automata

I. INTRODUCTION

In the last two decades, the computational researchers have been increasingly interested to the natural sciences, and especially biology, as source of modelling paradigms. Many research areas are massively influenced by the behaviour of various biological entities and phenomena. It gave birth to most of population-based Met 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].

III. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population based computational technique inspired from the simulation of social behaviour 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].

IV. CELLULAR AUTOMATA

Cellular automata are mathematical models for systems consisting of large number of simple identical components with local interactions in which space and time are discrete. It is called cellular because it is made up of cells like points in a lattice or like squares of checker boards, and it is called automata because it follows a simple rule[4]. Informally, a

Manuscript received February 20, 2012; revised April 29, 2012.

Noosheen Baktash and Fariborz Mahmoudi are with the Department of Electrical and Computer Qazvin Branch Islamic Azad University, Qazvin, Iran (e-mail: baktashnoosheen@gmail.com).

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

d-dimensional CA consists of a d-dimensional lattice of identical cells. The simple components act together to produce complicated patterns of behavior. Each cell can assume a state from a finite set of states. The cells update their states synchronously on discrete steps according to a local rule. The new state of each cell depends on the previous states of a set of cells, including the cell itself, and constitutes its neighborhood. The state of all cells in the lattice is described by a configuration. The rule and the initial configuration of the CA specify the evolution of CA that tells how each configuration is changed in one step. Cellular automata perform complex computations with a high degree of efficiency and robustness [4], [5].

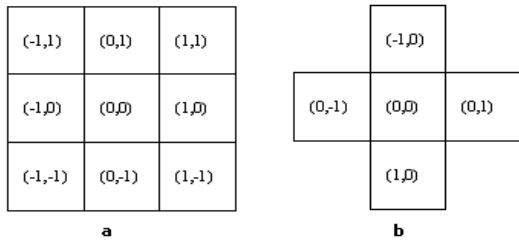


Fig. 1. Two Model of neighborhood for cellular automata (a) Moore; (b) Von Neumann.

V. 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 c_1 , c_2 and x in PSO, should be assigned in advanced.

```

Initialize the CA
For Each cell do in parallel
    Generate the initial population  $x_i$ ,  $i=1, \dots, S_n$ 
    Select half part of bees as employed bee with PSO
    Evaluate the fitness( $f_i$ )of the population
Do
For Each cell do in parallel
    For each employed bee{
        Produce new solution  $v_i$ 
        Calculate the value  $f_i$ 
        Apply greedy selection process}
        Calculate the probability values  $p_i$  for the solutions  $x_i$ 
    For each onlooker bee{
        Select a solution  $z_i$  depending on  $p_i$ 
        Produce new solution  $v_i$ 
        Calculate the values  $f_i$  with PSO function
        Apply greedy selection process}
    If there is an abandoned solution for the scout Then
        replace it with a new solution which will be randomly
        produced
    Memorize the best solution so far
    Calculate cbest
    Calculate cworst
    replace cworst with the best of neighbor
While maximum iterations OR
    minimum error criteria is not attained

```

Fig. 2. Pseudo code of cellular PSABC.

In Cellular-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. Then each colony is assigned to one cell of cellular automata with Moore neighbourhood model. This model utilized local interactions in cellular automata and split the colonies into different groups across cells of cellular automata. Each group tries to find an optimum locally which results in finding the global optima.

VI. 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) will be changed by adding variable $G_{v\text{syn}}$ 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 f_k is the best solution for algorithm, just before the k th environment change, h_k is the best value of k th environment state and k is the total number of environments.

$$f(\vec{x}, t) = \max(B(\vec{x}), \max P(\vec{x}, h_i(t), w_i(t), \vec{p}_i(t))) \quad (1)$$

$$i = 1 \dots m$$

$$\begin{aligned} \sigma &\in N(0,1) \\ h_i(t) &= h_i(t-1) + \text{height_severity} \cdot \sigma \\ w_i(t) &= w_i(t-1) + \text{width_severity} \cdot \sigma \\ \vec{p}_i(t) &= p_i(t-1) + \vec{v}_i(t) \end{aligned} \quad (2)$$

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

The default parameter setting of MPB used in the experiments is presented in Table 1. 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 f every height severity | 10 5000 evaluations 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 |
|----------------|----|
|----------------|----|

A. 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 4 present offline error results of algorithms for different frequency of environment change (f).

TABLE II: OFFLINE ERROR FOR F=500

| Peak count | Cellular DPSABC | CABC | CPSO | Adaptive MQSO | Adaptive CPSO | MQSO 10(10+10q) | MQSO 10(5+5q) | RPSO |
|------------|-----------------|--------------|--------------|---------------|---------------|-----------------|---------------|--------------|
| 1 | 2.30E-01 0 | 2.15E+0 0 | 1.35E+0 0 | 1.41E+00 0 | 1.22E+01 0 | 4.45E+01 0 | 3.37E+01 0 | 5.20E+0 0 |
| 10 | 1.29E+00 0 | 3.97E+0 0 | 9.35E+0 0 | 9.42E+00 0 | 9.18E+00 0 | 1.52E+01 0 | 9.62E+00 0 | 1.81E+0 1 |
| 20 | 1.41E+00 0 | 3.30E+0 0 | 8.84E+0 0 | 9.59E+00 0 | 8.78E+00 0 | 1.30E+01 0 | 9.07E+00 0 | 1.78E+0 1 |
| 30 | 2.15E+00 0 | 3.35E+0 0 | 8.81E+0 0 | 9.44E+00 0 | 8.64E+00 0 | 1.25E+01 0 | 8.80E+00 0 | 1.74E+0 1 |
| 40 | 2.18E+00 0 | 3.29E+0 0 | 8.94E+0 0 | 9.44E+00 0 | 8.68E+00 0 | 1.22E+01 0 | 8.55E+00 0 | 1.68E+0 1 |
| 50 | 1.22E+00 0 | 3.22E+0 0 | 8.62E+0 0 | 9.13E+00 0 | 8.49E+00 0 | 1.22E+01 0 | 8.72E+00 0 | 1.64E+0 1 |
| 100 | 1.70E+00 0 | 2.96E+0 0 | 8.54E+0 0 | 8.81E+00 0 | 8.22E+00 0 | 1.15E+01 0 | 8.54E+00 0 | 1.50E+0 1 |
| 200 | 1.12E+00 0 | 2.86E+0 0 | 8.28E+0 0 | 8.21E+00 0 | 7.89E+00 0 | 1.13E+01 0 | 8.19E+00 0 | 1.40E+0 1 |

TABLE III: OFFLINE ERROR FOR F=1000

| Peak count | Cellular DPSABC | CABC | CPSO | Adaptive MQSO | Adaptive CPSO | MQSO 10(10+10q) | MQSO 10(5+5q) | RPSO |
|------------|-----------------|--------------|--------------|---------------|---------------|-----------------|---------------|---------------|
| 1 | 1.27E+0 0 | 6.77E+0 0 | 6.77E+0 0 | 6.60E+0 0 | 5.83E+0 0 | 2.17E+01 0 | 1.86E+0 0 | 2.40E+00 1 |
| 10 | 1.25E+0 0 | 2.79E+0 0 | 5.19E+0 0 | 5.64E+0 0 | 5.29E+0 0 | 7.66E+00 0 | 5.71E+0 0 | 1.58E+01 0 |
| 20 | 1.53E+0 0 | 2.95E+0 0 | 5.23E+0 0 | 5.95E+0 0 | 5.49E+0 0 | 7.25E+00 0 | 5.85E+0 0 | 1.56E+01 0 |
| 30 | 1.26E+0 0 | 2.37E+0 0 | 5.33E+0 0 | 5.97E+0 0 | 5.42E+0 0 | 7.19E+00 0 | 5.81E+0 0 | 1.49E+01 0 |
| 40 | 1.09E+0 0 | 2.28E+0 0 | 5.61E+0 0 | 6.12E+0 0 | 5.32E+0 0 | 7.19E+00 0 | 5.70E+0 0 | 1.45E+01 0 |
| 50 | 1.10E+0 0 | 1.34E+0 0 | 3.86E+0 0 | 3.56E+0 0 | 3.30E+0 0 | 4.36E+00 0 | 3.63E+0 0 | 1.20E+01 0 |
| 100 | 1.56E+0 0 | 2.08E+0 0 | 5.57E+0 0 | 5.78E+0 0 | 2.08E+0 0 | 6.94E+00 0 | 5.83E+0 0 | 1.25E+01 0 |
| 200 | 1.03E+0 0 | 2.00E+0 0 | 5.50E+0 0 | 5.54E+0 0 | 2.00E+0 0 | 6.98E+00 0 | 5.54E+0 0 | 1.16E+01 0 |

TABLE IV: OFFLINE ERROR FOR F=2500

| Peak count | Cellular DPSABC | CABC | CPSO | Adaptive MQSO | Adaptive CPSO | MQSO 10(10+10q) | MQSO 10(5+5q) | RPSO |
|------------|-----------------|--------------|--------------|---------------|---------------|-----------------|---------------|--------------|
| 1 | 9.60E-03 0 | 1.74E+0 0 | 4.15E+0 0 | 2.48E+0 0 | 2.00E+0 0 | 9.88E+00 0 | 7.64E+0 0 | 1.00E+0 0 |
| 10 | 5.60E-02 0 | 1.85E+0 0 | 2.82E+0 0 | 2.91E+0 0 | 3.03E+0 0 | 4.38E+00 0 | 3.12E+0 0 | 1.37E+0 1 |
| 20 | 3.20E-01 0 | 1.96E+0 0 | 3.41E+0 0 | 3.40E+0 0 | 3.17E+0 0 | 4.34E+00 0 | 3.58E+0 0 | 1.39E+0 1 |
| 30 | 1.12E+00 0 | 1.37E+0 0 | 3.62E+0 0 | 3.47E+0 0 | 3.22E+0 0 | 4.36E+00 0 | 3.63E+0 0 | 1.30E+0 1 |
| 40 | 1.24E+00 0 | 1.31E+0 0 | 3.84E+0 0 | 3.56E+0 0 | 3.32E+0 0 | 4.37E+00 0 | 3.00E+0 0 | 1.25E+0 1 |
| 50 | 1.71E+00 0 | 2.25E+0 0 | 5.55E+0 0 | 5.98E+0 0 | 5.25E+0 0 | 7.14E+00 0 | 5.87E+0 0 | 1.42E+0 1 |
| 100 | 1.15E+00 0 | 1.29E+0 0 | 4.10E+0 0 | 3.53E+0 0 | 3.35E+0 0 | 4.21E+00 0 | 3.58E+0 0 | 1.04E+0 1 |
| 200 | 1.13E+00 0 | 1.26E+0 0 | 3.97E+0 0 | 3.37E+0 0 | 3.29E+0 0 | 4.04E+00 0 | 3.30E+0 0 | 9.63E+0 0 |

VII. CONCLUSION

In this paper, a hybrid algorithm of ABC, PSO and cellular automata was proposed. In this algorithm the hive consists of a number of cells and the bees randomly put in the cells of cellular automata, in a way that for each cell of cellular automata is allotted a bee. The Bees which are distributed among the cells, try to improve their position by local interaction with the neighboring cells. One factor to prevent the algorithm's trapping in the local optimal is fine parameter in which the bee that is chosen to seek and can't improve its position will be fined and at the end of each generation the bees which are fined up to fine threshold, greedily are replaced with their best neighbors. The algorithm tested by the series of typical problem instances. Because the cellular automata model is inherently parallel, this parallelism potentially is inherited to the bee colony, so the proposed algorithm is capable of finding the answer in the same time as standard algorithm. The simulation results shows that the proposed algorithm has higher converging velocity for functions in all dimensions.

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. Baradarani Hashemi and M. R. Meybodi, "A Multi-Role Cellular PSO for Dynamic Environments," in *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.



Fariborz Mahmoudi received his B.S. degree from the Shahid Beheshti University, Tehran, Iran. The M.S. degree from Amirkabir University and his Ph.D. degree in Software engineering from Azad University of Tehran, Iran. Dr. Mahmoudi is currently a faculty member at the Faculty of Electronic and Computer Engineering at Qazvin Azad University. His research has focused on Learning Systems, Software Engineering, Soft Computing and Grid Computing.



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.