Probabilistic Optimization Algorithms for Numerical Function Optimization Problems

M.-H. Tayarani-N. and M.-R. Akbarzadeh-T.[1]
Islamic Azad University of Mashhad
Iran
tayarani@ieee.org, akbarzadeh@ieee.org

***Abstract*— This paper proposes a novel optimization algorithm called Cellular Probabilistic Optimization Algorithms (CPOA) based on the probabilistic representation of solutions for real coded problems. In place of binary integers, the basic unit of information here is a probability density function. This probabilistic coding allows superposition of states for a more efficient algorithm. This probabilistic representation enables the algorithm to climb the hills in the search space. Furthermore, the cellular structure of the proposed algorithm aims to provide an appropriate tradeoff between exploitation and exploration. The proposed algorithm is tested on several numeric benchmark function optimization problems. Experimental results show that the performance of CPEA is improved when compared with other evolutionary algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). Furthermore, this improvement becomes particularly more significant for problems with higher dimension.**

***Keywords*— Evolutionary Algorithms, Probabilistic Evolutionary Algorithms, Optimization.**

I. INTRODUCTION

Preserving diversity in population is a well known approach for improving the performance of evolutionary algorithms. A diversity maintaining mutation is proposed in [1] to improve the performance of GA. A fuzzy controller for adjusting crossover rate and mutation rate is proposed in [2] that maintain the diversity in the population. For solving a closed-loop time optimal path planning problem, [3] uses a multi-objective diversity controlled genetic algorithm. Reference [4] uses a chaotic mutation in genetic algorithm for training artificial neural networks. The chaotic mutation is very efficient for maintaining the population diversity during the evolutionary process of GA.

Reference [5] presents a genetic algorithm with population diversity handling to maximize user satisfaction during composition of web services. An improved multi-objective diversity control oriented genetic algorithm which is chromosome representation independent is proposed in [6]. For the optimization of frequency response masking FIR digital filters over the double base number system multiplier coefficient space, [7] presents a novel diversity controlled genetic algorithm. In another work In order to overcome premature convergence in GA, [8] proposes a novel adaptive genetic algorithm with adaptive crossover probability based on diversity maintaining. An especial population selection policy is proposed in [9] based on the combination of population diversity handling and simulated annealing that enhance the convergence of genetic algorithm. Diversity based fuzzy adaptive search method for parallel GA is proposed in [10] to improve the performance of GA. For decreasing the computation complexity of GA, [11] uses a small population for GA. In order to improve the diversity in small populations they reinitialize the converged individuals.

The diversity of the population is also effective on the performance of other population based optimization algorithms such as the particle swarm optimization (PSO) as reported in [12]. In [13] a novel PSO is proposed that reinitializes the particles with poorer fitness to maintain the diversity in the population. Another approach for maintaining diversity in the population is using quantum particles. Reference [14] proposes a method of controlling the diversity in quantum behaved PSO to enable the particles to escape the sub optima more easily.

Quantum Evolutionary Algorithm (QEA) is an approach in which chromosomes are coded after quantum states of electrons in a probabilistic fashion. The resulting architecture is highly suitable to preserve diversity, i.e. each chromosome consists of m Q-bits that is equivalent to 2^m states. In quantum informatics, the basic carrier of information is not a bit but a quantum system with two states such as in an atom, an ion or a photon with two polarized directions, or the Q-bit. A Q-bit is in a linear superposition state and is used to specify the amplitudes of two states. In [15, 16] quantum-inspired evolutionary algorithms are investigated for a class of combinatorial optimization problems in which quantum rotation gates act as update operators. This quantum rotation gate is also used in a novel parallel quantum GA for hierarchical ring model and infinite impulse response (IIR) digital filter design [17]. Reference [18] proposes quantum evolutionary algorithm for multi-objective optimization and quantum rotation gate.

While the above algorithms have reported good success in application of QEA and its variants on several problems, this paradigm is inherently defined for problems with binary representation. For real coded problems, the problem in real domain must first be mapped to a binary coding before optimization by QEA. This approximation can introduce undesirable limitations and errors for QEA on real coded problems.

The advantage of QEA is in the superposition of its states, i.e. each quantum chromosome can represent any superposition

[1]Author is also currently affiliated with the department of electrical engineering and computer engineering, Ferdowsi University of Mashhad

978-1-4244-1674-5/08 /$25.00 ©2008 IEEE

of states simultaneously. This characteristic increases the diversity of the population and improves the exploration of the algorithm without loss of exploitation. Imitating the concept of superposition of states, this paper proposes a novel optimization algorithm for real coded problems that, similar to QEA, has a probabilistic structure that aims to take advantage of superposition of states. Furthermore, a cellular interaction architecture is proposed that promotes local interaction/leadership for better exploitation/exploration of local neighborhoods.

The reminder of this paper is organized as follows. In Section II, the proposed cellular probabilistic optimization algorithm (CPOA) is explained. In Section III the best parameters for the proposed algorithm is found. The experimental results are discussed in Section IV, and finally we conclude the proposed algorithm in Section V.

II. The Proposed Cellular Probabilistic Optimization Algorithm

The advantage of QEA is the superposition of states i.e. each quantum individual can represent any superposition of states simultaneously. But QEA is suitable only for the binary coded problems, and for the problems with real parameters this algorithm may not work well due to approximation error during binary-real conversions. This paper proposes a probabilistic optimization algorithm for real coded problems that, similar to QEA, has a probabilistic representation.

A. Representation

In evolutionary algorithms to date, the representation of solutions onto individuals can be classified as: binary, numeric, symbolic and recently Q-bit. The proposed Probabilistic Optimization Algorithm uses a new probabilistic representation for solutions called P-value. This representation aims to preserve better diversity. A P-value individual is a string of P-values. A P-value is defined as:

$$\begin{bmatrix} \mu \\ \sigma \end{bmatrix}$$

Where μ is the mean and σ is the standard deviation of a Gaussian probability density function. The value of μ is in the range of $l \le \mu \le u$ where l and u are the lower and the upper bounds of the search space, respectively. A P-value individual is defined as a string of P-values:

$$p = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_k & \cdots & \mu_m \\ \sigma_1 & \sigma_2 & \cdots & \sigma_k & \cdots & \sigma_m \end{bmatrix} \quad (1)$$

Where k=1,2,…,m. And m is the dimension of the search space. This probabilistic representation makes the diversity of the population higher than in other non-probabilistic representations and hence each individual is able to represent all of values of the entire search space simultaneously.

B. Cellular POA

Cellular POA is a structured evolutionary algorithm in which the individuals are located in a lattice like environment and each individual interacts only with its neighbors. In CPOA, the connections among neighbors helps the algorithm to exploit possible solutions of the algorithm, and the overlapped small neighborhoods help the algorithm to explore the search space because the induced slow diffusion of solutions through the population provides a kind of exploration [23]. So the importance of cellular structure for CPOA is that the fitness and genotype diversity in the population is preserved for a long number of generations. The structure of the CPOA which is used in this paper is shown in Fig. 1.

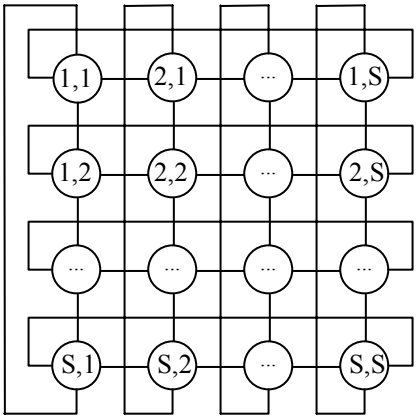

Figure 1. The structure of the Cellular CPOA for a lattice with the size of S

The pseudo code of CPOA is considered as:

```
Procedure CPOA
begin
      t=0
   1. initialize P(0) in a cellular structure.
   2. make X(0) by observing the values of
       P(0).
   3. evaluate X(0).
   4. for all real solutions x_ij^0 in X(0) do
       begin
   5.    find N_ij in X(0).
   6.    select real-valued solution x with
          best fitness in N_ij and store it in
          B_ij
       end
   7. while not termination condition do
      begin
         t=t+1
   8.    make X(t) by observing the values of
            P(t-1)
   9.    evaluate X(t)
   10.   update P(t) based on B_ij and X(t)
            using update operator.
   11.   for all real solutions x_ij^t in X(t) do
          begin
   12.      find N_ij in X(t).
   13.      select possible solution x with
             best fitness in N_ij.
   14.      if x is fitter than B_ij save x in B_ij.
   15.      if R(0,1)<r_μ reinitialize μ_ij^t randomly.
   16.      if R(0,1)<r_σ reinitialize σ_ij^t randomly.
          end
      end
end
```

CPOA has a cellular population of the probabilistic individuals. The population of probabilistic individuals is represented as:

$$P(t) = \left\{ p_{ij}^{t} \mid i, j = 1,2,..., S \right\}$$

Where t is generation number and S is the size of lattice-like population. The probabilistic individual's p_{ij}^{t} are defined as:

$$p_{ij}^{t} = \begin{bmatrix} \mu_{ij,1}^{t} & \mu_{ij,2}^{t} & \cdots & \mu_{ij,k}^{t} & \cdots & \mu_{ij,m}^{t} \\ \sigma_{ij,1}^{t} & \sigma_{ij,2}^{t} & \cdots & \sigma_{ij,k}^{t} & \cdots & \sigma_{ij,m}^{t} \end{bmatrix} \quad (2)$$

Where m is the number of P-values in the probabilistic individual, i.e., the string length of the P-valued individuals, i,j=1,2,…,S, S is the size of lattice-like population, i,j shows the location of the q-individual in the lattice-like population and t is the generation number of the evolution.

The procedure of CPOA is described as follows:

1. In the initialization step we set:

$$\begin{bmatrix} \mu_{ij,k}^{0} \\ \sigma_{ij,k}^{0} \end{bmatrix} = \begin{bmatrix} R(l_k, u_k) \\ a \end{bmatrix}$$

for i,j=1,2,…,S and k=1,2,…,m. Where S is the size of lattice-like population, m is the size of probabilistic individual i.e. the dimension of search space, l_k, u_k are the lower bound and the upper bound of k-th dimension of search space and a is a constant number.

2. In the observe step in iteration 0, the real valued solutions $X(0) = \{x_{ij}^{0} \mid i, j = 1,2,...S\}$ are made by observing the probabilistic individuals $P(0) = \{p_{ij}^{0} \mid i, j = 1,2,..., S\}$. The real valued individual x_{ij}^{0} is a possible solution for the problem. The observing operation is performed as:

$$x'^{t}_{ij,k} = G(\mu_{ij,k}^{t}, \sigma_{ij,k}^{t}) \quad (3)$$

$$x_{ij,k}^{t+1} = \begin{cases} l_k & x'^{t}_{ij,k} \le l_k \\ x'^{t}_{ij,k} & l_k < x'^{t}_{ij,k} < u_k \\ u_k & u_k \le x'^{t}_{ij,k} \end{cases}$$

Where G(.,.) is a Gaussian random number generator. Here each real valued individual x_{ij}^{t} is a string of real values which is a possible solution for the real coded problem.

3. In this step all the real valued individual's x_{ij}^{0} are evaluated with fitness function.
4. In this step the "for" loop is run for all real solutions x_{ij}^{0}.
5. The neighbors of real solution x_{ij}^{0} are found and stored in N_{ij}.

Suppose that the size of lattice is S and the probabilistic individual located at (i,j) is represented as p_{ij} (i,j=1,2,…,S). The neighbors of the individual p_{ij} are then defined as:

$$N_{ij} = \left\{ p_{i'j}, p_{ij'}, p_{i''j}, p_{ij''}, p_{ij} \right\}$$

Where:

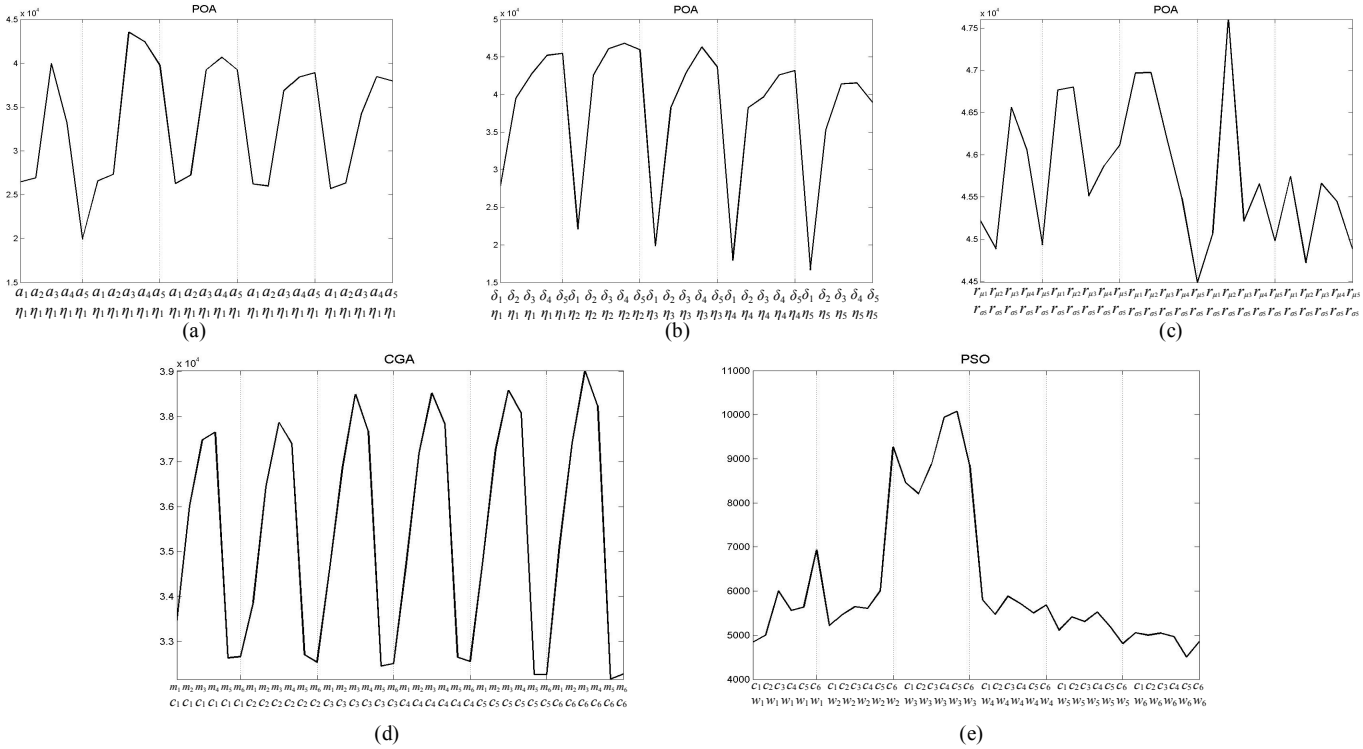

Figure 2. Parameter setting for CPOA, GA and PSO for Generalized Rastrigin's Function. a) The parameters of CPOA for a and η are set to $a_1 \ldots a_5$=(0,0.01,0.1,1,2,5), $\eta_1 \ldots \eta_5$= (0.995,0.99,0.98,0.95,0.9). b) The parameters of CPOA for δ and η are set to $\delta_1 \ldots \delta_5$=(0,0.01,0.1,1,2,5), $\eta_1 \ldots \eta_5$= (0.995,0.99,0.98,0.95,0.9). c) The parameters of CPOA for r_μ and r_σ are set to $\delta_1 \ldots \delta_5$=(0,0.01,0.1,1,2,5), $\eta_1 \ldots \eta_5$= (0.995,0.99,0.98,0.95,0.9). d) The mutation rate and crossover rate of GA is $m_1 \ldots m_6$=(0.001,0.002,0.004,0.008,0.016,0.02), $c_1 \ldots c_6$=(0.1,0.2,0.4,0.6,0.8,1). e) The parameters of PSO is set to $c_1 \ldots c_6$=(0.002,0.005,0.01,0.05,0.1,0.4) and $w_1 \ldots w_6$=(0.2,0.5,1,1.2,1.5,2). All results are averaged over 50 runs.

$$i' = \begin{cases} i-1 & i \neq 1 \\ S & i = 1 \end{cases}, \qquad j' = \begin{cases} j-1 & j \neq 1 \\ S & j = 1 \end{cases}$$

$$i'' = \begin{cases} i+1 & i \neq S \\ 1 & i = S \end{cases}, \qquad j'' = \begin{cases} j+1 & j \neq S \\ 1 & j = S \end{cases}$$

6. The best real solution x in N_{ij} is found and stored to B_{ij}.
7. The algorithm runs until termination condition is satisfied. Termination condition here is when maximum number of iterations is reached.
8. In this step real valued individuals $X(t)$ are made by observing the probabilistic individuals $P(t\text{-}1)$.
9. The real solutions are evaluated by fitness function.
10. In this step $P(t)$ is updated based on the values of $B(t)$ and $X(t)$. The update operator is performed as:

$$\begin{bmatrix} \mu_{ij,k}^{t+1} \\ \sigma_{ij,k}^{t+1} \end{bmatrix} = \begin{bmatrix} \delta \times (u_k - l_k) \\ 0 \end{bmatrix} \times \mathrm{R}(0,1) + \begin{bmatrix} \mu_{ij,k}^{t} \\ \eta \times \sigma_{ij,k}^{t} \end{bmatrix} \quad (4)$$

for i,j=1,2,…,S and k=1,2,…,m. Where R(.,.) is a uniform random number generator, $0<\eta<1$ and δ is considered as Table. I.

11. In this step the "for" loop is running for all real solutions x_{ij}^{t}.
12. The neighbors of real solutions x_{ij}^{t} are found and stored in N_{ij}.
13. The best real solution x in N_{ij} is found.
14. If x is fitter than B_{ij} store it to B_{ij}.

15,16. In CPOA we consider the reinitialization operator for exploring the search space. In these two steps if reinitialization condition is satisfied we reinitialize the μ_{ij}^{t} and/or σ_{ij}^{t}.

The probabilistic representation of the CPOA enables it to climb the hills in the search space. Suppose that a probabilistic individual p_{ij}^{t} is located around a local optimum in the search space. According to the probabilistic representation of the individuals, an observation of the p_{ij}^{t} makes a possible solution x_{ij}^{t} near the probabilistic individual p_{ij}^{t}. If the observed individual x_{ij}^{t} has an inferior situation than B_{ij} it is ignored; on the other hand if it has a better situation than B_{ij}, the B_{ij} is changed to x_{ij}^{t}. After this changing, the probabilistic individual p_{ij}^{t} moves gradually to B_{ij}. This process continues until the probabilistic individual reaches the local optimum. This characteristic of CPOA makes each probabilistic individual able to perform a local random search in the search space, it means that the observation and updating operators have the roles of exploration of the search space; on the other hand, the restrictive connectivity between the individuals in the cellular structured population performs a global search on the search space. The cellular structure of CPOA joint to the reinitialization operator, make CPOA a powerful algorithm for exploring the search space. So in the proposed algorithm, the observing operator in conjunction with updating operator have the exploitation role, and the restrictive connectivity and re-initialization operators have the exploration role.

TABLE I. LOOKUP TABLE OF $\Delta\mu$. $f(x)$ IS THE FITNESS OF REAL SOLUTION x_{ij}^{t} AND $f(b)$ IS THE FITNESS OF B_{ij}

$x_{ij,k} > B_{ij,k}$	$f(x) \geq f(b)$	$\Delta\mu$
False	*False*	δ
False	*True*	$-\delta$
True	*False*	$-\delta$
True	*True*	δ

III. PARAMETER SETTING

For evaluating the proposed algorithm CPOA is compared with PSO and GA. In CPOA, GA and PSO there are some parameters that must be found. For comparing these algorithms at first the best parameters for these algorithms are found. This Section tries to find the best parameters for these algorithms.

In this paper the PSO algorithm is considered as [19]:

$$v_i^{t+1} = w v_i^t + c_1 \mathrm{R}(0,1)\left(\mathrm{pbest}_i^t - x_i^t\right) + c_2 \mathrm{R}(0,1)\left(\mathrm{gbest}^t - x_i^t\right) \quad (5)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Where c_1 and c_2 are two positive constants, R(.,.) is a uniform random number generator, w is the inertia weight, pbest_i^t is the best position which particle i has achieved, gbest^t is the best particle position which the overall swarm has achieved and t is the iteration number of the algorithm.

There are some parameters in CPOA, PSO and GA that must be tuned; they are crossover rate and mutation rate in GA, c_1, c_2 and w in PSO and a, δ, η and reinitialization probability in CPOA. For finding these parameters 14 numeric benchmark functions are used (see appendix A). It is obvious that the best parameters for each algorithm are problem dependent, so this paper finds the best parameters for each benchmark function independently.

The mutation in GA is considered as: for each allele of each chromosome, mutate the allele with probability of mutation.

TABLE II. THE BEST PARAMETERS FOR GA, PSO AND CPOA

	GA		PSO		CPOA				
	m	c	c_1,c_2	w	a	δ	η	r_μ	r_σ
Schwefel 2.26	0.004	1	0.1	1	1	0.02	0.098	0.002	0.002
Rastrigin	0.004	1	0.01	0.2	0.1	0.005	0.095	0.002	0.002
Ackley	0.008	1	0.05	0.2	0.1	0.005	0.995	0.002	0.002
Griewank	0.004	1	0.01	0.2	0.1	0.005	0.99	0.002	0.002
Penalized 1	0.004	1	0.005	0.5	0.01	0.005	0.095	0.02	0.002
Penalized 2	0.004	1	0.01	0.2	0.1	0.005	0.995	0.002	0.002
Michalewicz	0.004	1	0.4	0.5	0.1	0.002	0.98	0.002	0.002
Goldberg	0.004	1	0.005	0.5	1	0.005	0.99	0.002	0.002
Sphere Model	0.004	1	0.01	0.2	1	0.005	0.98	0.002	0.002
Schwefel 2.22	0.004	1	0.005	0.5	0.1	0.005	0.995	0.002	0.002
Schwefel 2.21	0.008	0.8	0.01	0.5	0.1	0.01	0.995	0.002	0.002
Dejong	0.008	0.1	0.005	0.5	0.1	0.005	0.995	0.002	0.002
Rosenbrock	0.004	1	0.005	0.2	0.1	0.005	0.99	0.002	0.002
Kennedy	0.008	0.1	0.005	1	2	0.005	0.9	0.002	0.002

TABLE III. EXPERIMENTAL RESULTS OF THE FOURTEEN NUMERICAL FUNCTION OPTIMIZATION PROBLEMS. THE NUMBER OF RUNS WAS 50. MEAN AND STD REPRESENT THE MEAN BEST STANDARD DEVIATION OF 50 RUNS RESPECTIVELY

	$m=100$							
	CPOA		GA		PSO		QEA	
	MEAN	**STD**	**MEAN**	**STD**	**MEAN**	**STD**	**MEAN**	**STD**
Schwefel	4.52×10^{4}	4.12×10^{3}	3.90×10^{4}	4.99×10^{2}	9.74×10^{3}	1.47×10^{3}	3.39×10^{4}	2.80×10^{3}
Rastrigin	-92.93	13.28	-136.68	12.09	-822.30	42.45	-2.05×10^{3}	2.62×10^{2}
Ackley	0.51	0.11	-5.25	0.40	-8.69	0.47	-17.245	0.11
Griewank	0.73	0.02	-0.36	0.06	-1.00	0.17	-39.39	6.64
Penalized 1	61.87	5.20	33.74	27.14	-671.41	653.10	-1.70×10^{5}	2.37×10^{3}
Penalized 2	-13.43	3.95	-192.87	35.66	-523.10	125.36	-3.94×10^{4}	4.69×10^{3}
Michalewicz	18.71	2.00	78.02	2.03	13.78	1.94	22.58	2.63
Goldberg	91.64	1.38	92.57	0.54	57	2.09	39.43	3.37
Sphere Model	-3.73×10^{1}	2.63	-3.99×10^{3}	4.21×10^{2}	-1.07×10^{4}	1.74×10^{3}	-4.65×10^{5}	5.66×10^{3}
Schwefel 2.22	-5.98×10^{-2}	5.45×10^{-3}	-3.19×10^{-1}	2.41×10^{-2}	-8.26×10^{-1}	7.09×10^{-2}	-5.072	0.47
Schwefel 2.21	-6.63	0.75	-53.65	3.11	-26.30	3.89	-176.04	5.18
Dejong	1.37×10^{1}	5.52	-1.20×10^{4}	4.85×10^{3}	-1.37×10^{4}	5.40×10^{3}	-2.83×10^{7}	5.88×10^{6}
Rosenbrock	-9.94×10^{1}	0.49	-7.11×10^{2}	1.04×10^{2}	-6.01×10^{2}	8.97×10^{1}	-1.03×10^{5}	2.5×10^{4}
Kennedy	-1.15×10^{-5}	1.04×10^{-7}	-1.30×10^{-1}	1.88×10^{-2}	-1.20×10^{1}	3.85	-1.96	1.44

	$m=250$			
	CPOA	GA	PSO	QEA

The changing operation is performed as changing the value of allele to a random number between the lower and the upper bound of the search space. The pseudo code of mutation operator is considered as:

```
Procedure mutation
 begin
 1.  for all individuals x^t_i in X^t do
 2.    for all alleles x^t_{i,k} in x^t_i do
 3.      if R(0,1)<mutation rate
 4.        x^t_{i,k}=R(l_k,u_k)
 end
```

Where R(.,.) is a uniform random number generator, $x^t_{i,k}$ is k-th allele of i-th individual in the population in generation t and l_k and u_k are the lower bound and the upper bound of the search space respectively.

Fig. 2 shows the parameter setting of CPOA, GA and PSO for Generalized Rastrigin's Function. In this figure the horizontal axis is the parameters of the algorithm and the vertical axis is the best results averaged over 50 runs. Table II summarizes the best parameter setting for the 14 benchmark functions. The method of finding the best parameters for the algorithms for all benchmark functions is the same as Fig 2. We use a three stage process for finding the best parameters for CPOA. At first we determine the best parameters for a and η by considering constant numbers of δ=0.005 and r_μ and r_σ equal to zero. After finding the best values for a and η we determine the best parameters for δ and η by considering a equal to the best parameter found in the first stage and r_μ and r_σ equal to zero. And finally in the third stage we determine the best parameters for r_μ and r_σ. In this stage we consider the constant values for a, δ and η that are equal to the best parameters found in previous stages.

IV. EXPERIMENTAL RESULTS

In this paper the dimension of the problems is set to m=100, 250, 500 and 1000. The population size for all of the experiments is set to 25, and maximum generation termination condition is used. All results are averaged over 50 runs. The parameters of the algorithms (a, δ, η and reinitialization probability for CPOA, mutation rate and crossover rate for GA and c_1, c_2 and w in PSO) are considered as the best values that are found in Section III. The structure of QEA is considered as the structure proposed in [15].

Table III summarizes the experimental results of CPOA, GA and PSO for m=100, 250, 500 and 1000 for 14 benchmark functions. As indicated by all of the benchmark functions, the proposed algorithm consistently has a better performance than PSO. The functions which are discussed are proposed for minimization and have a global minimum with some local minima. Because the algorithms are designed for maximization, we redefine them to maximize $-f(x)$. As it seems in most of benchmark functions the performance of CPOA is

better than GA, except in Michalewicz Function. Also the performance of CPOA in all problems is better than PSO. The performance of the CPOA shows that the convergence speed of this algorithm on the problems with small numbers of local optima is very high. This is because the probabilistic representation of the algorithm performs a local random search on the local optima. The proposed algorithm has a better performance than QEA because QEA is designed for binary coded problems [15].

V. CONCLUSION

The proposed Probabilistic Evolutionary Algorithms, similar to Quantum EA, aim to increase population diversity by their probabilistic nature; but unlike QEA, PEAs are applicable to the real-coded problems. Furthermore, the cellular architecture promotes better exploitation of local neighborhoods and avoids premature convergence. The proposed approach shows superior performance when applied to 14 benchmark problems and compared with GA and PSO. There are some open questions in this algorithm, such as the optimum (δ, η and reinitialization probability) parameters for CPOA. It is obvious that the optimality of the parameters is problem dependent, but what are the optimal parameters for a specific problem? Second is if the algorithm remains superior in other real coded problems such as training neural network and fuzzy systems.

APPENDIX

There are some benchmark numerical functions for testing the optimization algorithms. Here we used 14 benchmark functions for testing the algorithms:

Generalized Schwefel's Problem 2.26 [21], Generalized Rastrigin's Function [21], Ackley's function [21], Generalized Griewank Function [21], Generalized Penalized Function 1 [21], Generalized Penalized Function 2 [21], Michalewicz Function [20], Goldberg & Richardson Function [22], Sphere Model [21], Schwefel's Problem 2.22 [21], Schwefel's Problem 2.21 [21], Dejong Function 4 [20], Rosenbrock Function [22], and Kennedy multimodal function generator [22].

These functions have some local minima and a global minimum. Since they are used for maximization, $-f(x)$ is used as fitness value.

REFERENCES

[1] ZhiMing Liu, JiLiu Zhou, ZhiCheng Wei, Hang Lu Li Tao, "A study on novel genetic algorithm with sustaining diversity," *IEEE International Conference on Signal Processing Proceedings*, China 2000.

[2] Kejun Wang, , "A new fuzzy genetic algorithm based on population diversity," *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2001.

[3] Weerayuth, N. Chaiyaratana, N, "Closed-loop time-optimal path planning using a multi-objective diversity control oriented genetic algorithm," *IEEE International Conference on Systems, Man and Cybernetics*, oct 2002.

[4] Qingzhang Lü, Guoli Shen and Ruqin Yu, "A chaotic approach to maintain the population diversity of genetic algorithm in network training," *Elsevier, Computational Biology and Chemistry* Vol 27, July pp 363-371, 2003.

[5] Chengwen Zhang, Sen Su, and Junliang Chen, "Efficient Population Diversity Handling Genetic Algorithm for QoS-Aware Web Services Selection," *Springer Lecture Notes in Computer Science* Volume 3994, pp 104-111, May 10, 2006.

[6] Theera Piroonratana, and Nachol Chaiyaratana, "Improved Multi-Objective Diversity Control Oriented Genetic Algorithm," *Springer Lecture Notes in Computer Science* Volume 4029, pp 430-439, Jul 1, 2006.

[7] Sai Mohan Kilambi Nowrouzian, B, "A Diversity Controlled Genetic Algorithm for Optimization of FRM Digital Filters over DBNS Multiplier Coefficient Space," *IEEE International Symposium on Circuits and Systems*, New Orleans, May 2007.

[8] Xing, Xiaojun Jia, Qiuling Ling, Zhigang Yuan, Dongli, "A Novel Genetic Algorithm Based on Individual and Gene Diversity Maintaining and Its Simulation," *IEEE International Conference on Automation and Logistics*, Jinan, China Aug. 2007.

[9] Chengwen Zhang, Sen Su and Junliang Chen, "DiGA: Population diversity handling genetic algorithm for QoS-aware web services selection," *Elsevier, Advances in Computer Communications,* Vol 30, pp 1082-1090, 8 March 2007,.

[10] Yoichiro Maeda and Qiang Li, "Fuzzy Adaptive Search Method for Parallel Genetic Algorithm Tuned by Evolution Degree Based on Diversity Measure," *Springer Lecture Notes in Computer Science* Volume 4529, pp 677-687, Jul 2, 2007.

[11] Kenji Kobayashi, Tomoyuki Hiroyasu and Mitsunori Miki, "Mechanism of Multi-Objective Genetic Algorithm for Maintaining the Solution Diversity Using Neural Network," *Springer Lecture Notes in Computer Science* Volume 4403, pp 216-226, May 20, 2007.

[12] Daniel N. Wilke, Schalk Kok , Albert A. Groenwold, "Comparison of linear and classical velocity update rules in particle swarm optimization: notes on diversity," *International Journal for Numerical Methods in Engineering* Vol 70, Issue 8 , pp 962–984, 17 Nov 2006.

[13] Fang Wang Naiqin Feng Yuhui Qiu, "An adaptive diversity strategy for particle swarm optimization," *Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering*, Nov 2005.

[14] Jun Sun, Wenbo Xu and Wei Fang, "Quantum-Behaved Particle Swarm Optimization Algorithm with Controlled Diversity," *Springer Lecture Notes in Computer Science* Volume 3993, pp 847-854, May 10, 2006.

[15] K. Han, J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 6, 2002.

[16] K.H Han, K.H Park, C.H Lee, J.H Kim, "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem," *Proceeding of the IEEE congress on Evolutionary Computation* Seoul, Korea, 2001.

[17] G Zhung, W Jin, L Hu, "Novel parallel quantum GAs," *Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies*, PDCAT,'2003.

[18] G Zhung, W Jin, L Hu, "Quantum evolutionary algorithm for multi-objective optimization," *Proceedings of the 2003 IEEE International Symposium on Intelligent Control Houston*, Texas October 5-8.2003.

[19] M. R Alrashidi and M. E. EI-Hawary "A Survey of Particle Swarm Optimization Applications in Electric Power Systems" *IEEE Trans. Evol. Comput*. vol. 9, pp.

[20] A.-R. Khorsand, M.-R. Akbarzadeh-T. "Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm". *IEEE International Conference on Systems, Man and Cybernetics*, 2005.

[21] W. Zhong, J. Liu, M. Xue, L. Jiao, "A Multi-agent Genetic Algorithm for Global Numerical Optimization," *IEEE Trans. Sys, Man and Cyber.* vol. 34, pp. 1128–1141, 2004.

[22] V. K. Koumousis, C. P. Katsaras. "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," *IEEE Trans. Evol. Comput*. vol. 10, pp. 19–28, 2006.

[23] E. Alba, B. Dorronsoro," The Exploration/ Exploitation Tradeoff in Dynamic Cellular Genetic Algorithms," *IEEE Trans. Evol. Comput*. vol. 9, pp. 126–142, 2005.