I.J. Information Engineering and Electronic Business, 2018, 5, 40-45
Published Online September 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2018.05.06

Comparative Study of CEC'2013 Problem Using Dual Population Genetic Algorithm

A. J. Umbarkar
Department of Information Technology, Walchand College of Engineering Sangli, MS, India
Email: anantumbarkar@rediffmail.com

L. R. Moon
Department of Information Technology, Walchand College of Engineering Sangli, MS, India
Email: latikmoon@gmail.com

P. D. Sheth
Department of MCA
Government College of Engineering, Karad, 415-124, India
Email: pranalisheth@gmail.com

Received: 08 December 2017; Accepted: 19 January 2018; Published: 08 September 2018

***Abstract*—**Evolutionary Algorithms (EAs) are found to be effective for solving a large variety of optimization problems. In this Paper Dual Population Genetic Algorithm (DPGA) is used for solving the test functions of Congress on Evolutionary Computation 2013 (CEC'2013), by using two different crossovers. Dual Population Genetic Algorithm is found to be better in performance than traditional Genetic Algorithm. It is also able to solve the problem of premature convergence and diversity of the population in genetic algorithm. This paper proposes Dual Population Genetic Algorithm for solving the problem regarding unconstrained optimization. Dual Population Genetic Algorithm is used as meta-heuristic which is verified against 28 functions from Problem Definitions and Evaluation Criteria for the Congress on Evolutionary Computation 2013 on unconstrained set of benchmark functions using two different crossovers. The results of both the crossovers are compared with each other. The results of both the crossovers are also compared with the existing results of Standard Particle Swarm Optimization algorithm. The Experimental results showed that the algorithm found to be better for finding the solution of multimodal functions of the problem set.

***Index Terms*—**Dual Population Genetic Algorithm, DPGA, Genetic Algorithm, GA, Evolutionary Algorithm, EA, Function Optimization, CEC'2013, k-Point Crossover.

I. Introduction

Evolutionary Algorithms (EAs) is widely used for solving optimization problems. EAs are found to be useful from the last few decades to successfully solve the complex problems. Genetic algorithms (GAs) are population based stochastic evolutionary algorithms. It is based on the principal "survival of fittest". In the evolution of populations, GA loses the population diversity and gets trapped in local optima. This problem in EA is called as "premature convergence problem". It especially occurs in solving complex optimization problems, where search space has a lot of peaks and valleys in the fitness map [1].

The proposed solution to this problem in GA is using two populations instead of using only one. DPGA has two populations- main population and the reserve population. The job of the reserve population is to provide additional population diversity to the main population. The information between the main population and the reserve population is exchanged by means of inter-population crossbreeding. The crossbreeding technique helps to solve the problem of premature convergence.

In this paper, the unconstrained optimization problems defined in CEC'2013 are solved. Section II gives a brief literature review of DPGA. Section III describes DPGA with implementation details such as the crossover operators used in the experimentation. Section IV shows the experimental results, comparison of results and discussion. Section V gives conclusions and future scope.

II. Literature Review

This section firstly gives the literature of evolution of DPGA and then gives review of EAs, which are experimented on benchmark functions CEC'2013. The work provides a brief literature review on DPGA and the crossover operators used.

DPGA introduced by Park and Ruy in 2006 [1]. [2] gives the details review of DPGA. DPGA consists of two different populations with different evolutionary objectives. The objective of the main population is same as that of regular genetic algorithm which targets to optimize the objective function to its minimum or maximum as per required. The purpose of the reserved population is to maintain diversity. In 2007, Park and Ruy propose DPGA-ED [3]. Difference between a simple DPGA and DPGA-ED is that DPGA-ED evolves by itself. Park and Ruy unveiled DPGA2 that uses two reserve populations instead using only one population for providing diversity to the main population.

[4] proposes the approach of adjusting the distance between the main population and reserve population of DPGA. [5] applied DPGA to non-stationary Optimization.

Umbarkar, Joshi and Hong (2014) [6] improves the performance of DPGA by parallelizing it using multithreads. By using this technique they also solve the problem of population diversity and premature convergence.

Zambrano-Bigiarini, Clerc and Rojas (2013) [7] uses Standard Particle Swarm Optimization algorithm to solve the problem set of CEC'2013.

Elsayed, Sarker and Essam [10] applied GA on CEC 2013. [11] accelerate the Particle Swarm Optimization with Diversity-Guided Convergence and Stagnation Avoidance. [12] proposed the Diversity Enhanced Differential Evolution

III. Dual Population Genetic Algorithm

DPGA is a variant of GA which consists of two populations, the main population and the reserve population. Both the populations were initialized with random numbers. Individuals of both the populations were evaluated using their own fitness functions. The new generation of each population is obtained by inbreeding between the parents of the same population (Crossover operator), Crossbreeding between the parents of different populations, and the survival selection among the obtained offspring. The fitness function used for the reserve population, evolutionary process of simple DPGA and crossover operator used in it, are described as follows:

A. Crossover Operators

The main searching operator in this algorithm is the crossover operator while mutation and crossbreed are considered as a variation or diversity operator. In this paper, we have used two crossovers i.e. *k*-point crossover and Discrete TPX [8]. To provide multiple combinations of selected parents it selects more than one crossover points.

A.1 k-point Crossover Operators

The *k*-point crossover randomly selects *k* crossover points cp1 to cpk-1 in the selected parents. Two offspring are created by combining the parents at crossover points. The algorithm for *k*-point crossover is given below:

ALGORITHM 1:PSEUDO-CODE FOR k-POINT CROSSOVER

```
Select two parents A^(t) and B^(t)
Create two offspring C^(t+1) and D^(t+1)
Randomly select k crossover points cp_1,…,cp_k ϵ {1,…,n-1}
for i=1 to cp1 do
  c_i^(t+1) = a_i^(t)
  d_i^(t+1) = b_i^(t)
end do
switch = 0
for j = 2 to k do
  if switch = 0 then
    for i = cp_(j-1) + 1 to cp_j do
      c_i^(t+1) = b_i^(t)
      d_i^(t+1) = a_i^(t)
    end do
    switch = 1
  else
    for i = cp_(j-1) + 1 to cp_j do
      c_i^(t+1) = a_i^(t)
      d_i^(t+1) = b_i^(t)
    end do
    switch = 0
  end if
  if switch = 0 then
    for i = cp_(j-1) + 1 to cp_j do
      c_i^(t+1) = b_i^(t)
      d_i^(t+1) = a_i^(t)
    end do
  else
    for i = cp_(j-1) + 1 to cp_j do
      c_i^(t+1) = a_i^(t)
      d_i^(t+1) = b_i^(t)
    end do
  end if
```

In the example shown below the points 2nd and 3rd, 5th and 6th, 8th and 9th and 10th and 11th gene are selected as crossover points where value of *k* is 4:

Parent A :	1 0 | 1 0 1 | 1 0 1 | 1 1 | 0 0
Parent B :	1 1 | 1 0 1 | 0 1 1 | 0 0 | 0 1
Offspring C :	1 0 | 1 0 1 | 1 0 1 | 0 0 | 0 0
Offspring D :	1 1 | 1 0 1 | 0 1 1 | 1 1 | 0 1

A.2 Discrete TPX

The Discrete TPX is the combination of two crossover operators, binary encoded discrete crossover and real valued three parent crossover. Using three parents for crossover will provide the operator with more exploration. The algorithm for discrete three parent crossover is given below.

ALGORITHM 1:PSEUDO-CODE FOR DISCRETE TPX

Select three parents $A^{(t)}$, $B^{(t)}$ and $C^{(t)}$
Create two offspring $X^{(t+1)}$, $Y^{(t+1)}$ and $Z^{(t+1)}$
Randomly select k crossover points $cp_1,\dots,cp_k \in \{1,\dots,n-1\}$
for i=1 to n **do**
  select a real random number $u \in \{0,1\}$
  if $u \geq 0$ && $u < 0.33$ **then**
    $x_i^{(t+1)} = a_i^{(t)}$
    $y_i^{(t+1)} = b_i^{(t)}$
    $z_i^{(t+1)} = c_i^{(t)}$
  else if $u \geq 0.33$ && $u < 0.66$ **then**
    $x_i^{(t+1)} = b_i^{(t)}$
    $y_i^{(t+1)} = c_i^{(t)}$
    $z_i^{(t+1)} = a_i^{(t)}$
  else
    $x_i^{(t+1)} = c_i^{(t)}$
    $y_i^{(t+1)} = a_i^{(t)}$
    $z_i^{(t+1)} = b_i^{(t)}$
  end if
end do

B. *Evolutionary Process*

The initialization of the main population with size m and reserve population with size n, fitness is calculated using their own fitness functions. As set of m offspring are generated by inbreeding between the parents of main population and reserve population respectively using the operators like crossover and mutation. Then (*n-m*) offspring are generated by crossbreeding between the one parent from the main population and other from reserve population for each individual again by using crossover and mutation operator.

The newly generated individuals are evaluated using the objective function for the main population and only m individuals are selected among them on the basis of their fitness values for the next generation. As algorithm already has m offspring generated by the process of inbreeding, the crossbreed offspring can only survive if they are better than at least one of the inbreed offspring in terms of their fitness values. The newly generated individuals of the reserve population are evaluated by the fitness function of the reserve population. All of them are selected to constitute the next generation of the reserve population.

IV. Results and Discussion

Standard problems are taken for experiments from CEC'2013 Real-Parameter Optimization problem [9]. In this report, 28 benchmark functions are described. The performance of the proposed algorithm is analyzed in this section by solving the benchmark functions introduced in CEC 2013 [9]. The brief introduction to the set of problems is given as below:

Table 1. CEC'2013 Functions

Name	**Function**	**Optimum Value**
Unimodal Functions		
F01	Sphere Function	-1400
F02	Rotated High Conditioned Elliptic Function	-1300
F03	Rotated Bent Cigar Function	-1200
F04	Rotated Discus Function	-1100
F05	Different Power Function	-1000
Basic Multimodal Functions		
F06	Rotated Rosenbrock's Function	-900
F07	Rotated Schaffers F7 Function	-800
F08	Rotated Ackley's Function	-700
F09	Rotated Weierstrass Function	-600
F10	Rotated Griewank's Function	-500
F11	Rastrign's Function	-400
F12	Rotated Rastrign's Function	-300
F13	Non-Continuous Rotated Rastrign's Function	-200
F14	Schwefel's Function	-100
F15	Rotated Schwefel's Function	100
F16	Rotated Katsuura Function	200
F17	Lunacek Bi_Rastrigin Function	300
F18	Rotated Lunacek Bi_Rastrigin Function	400
F19	Expanded Griewank's plus Rosenbrock's Function	500
F20	Expanded Scaffer's F6 Function	600
Composition Function		
F21	Composition Function 1 (n=5,Rotated)	700
F22	Composition Function 2 (n=3,Unrotated)	800
F23	Composition Function 3 (n=3,Rotated)	900
F24	Composition Function 4 (n=3,Rotated)	1000
F25	Composition Function 5 (n=3,Rotated)	1100
F26	Composition Function 6 (n=5,Rotated)	1200
F27	Composition Function 7 (n=5,Rotated)	1300
F28	Composition Function 8 (n=5,Rotated)	1400

The results are taken on the AMD FX(tm)-8320 Eight-Core Processor with 3.51 GHz clock speed. The experiments are carried on system with 16GB RAM and hard disk of capacity 500GB with operating system CentOS 6.5.

Comparison between any two algorithms is done on the basis of student *t*-test value. The *t*-test value can be calculated by using following equation 1.

$$t = \frac{X_1 - X_2}{\sqrt{\frac{(n_1-1)S_1{}^2 + (n_2-1)S_2{}^2}{n_1+n_2-2} \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

Where, in equation (1) X_1 and X_2 are the mean of algorithm 1 and algorithm 2, n_1 and n_2 are the number of sample tested for the results, and S_1 and S_2 are the standard deviation of algorithm for a particular problem. If the value of t is found to be less than 0, will show that the first algorithm is better for solving the problem otherwise second one would be better.

A. Comparison between DPGA using k-point crossover and Discrete TPX for lower dimensions

Table 2 gives the comparative result of DPGA using *k*-point crossover versus DPGA using discrete TPX. It is clear from the *t*-test value that the Discrete TPX gives better results than *k*-point crossover for lower dimensions. Discrete TPX proves itself better for almost the functions except F17, F21, F23, F24, F25, F26 and F28 functions.

Table 2. Comparison between K-point crossover and Discrete TPX for lower dimension

Fn. No.	K-Point Crossover		Discrete TPX		t-Test
	M	SD	M	SD	
F1	-1.39E+03	3.66E+00	-1.40E+03	1.31E+00	0.339491
F2	-4.57E+02	4.03E+02	-9.38E+02	1.68E+02	0.397825
F3	-7.76E+02	3.11E+02	-8.35E+02	1.25E+02	0.062367
F4	3.24E+02	7.40E+02	-5.19E+02	1.59E+02	0.379786
F5	-9.96E+02	2.90E+00	-9.97E+02	2.13E+00	0.007903
F6	-9.00E+02	2.72E-01	-9.00E+02	6.94E-02	0.281849
F7	-7.92E+02	2.21E+00	-7.98E+02	1.43E+00	0.818658
F8	-6.95E+02	2.80E+00	-6.95E+02	2.85E+00	0.006369
F9	-5.99E+02	1.47E-01	-6.00E+02	8.60E-02	1.616449
F10	-4.98E+02	1.49E+00	-4.99E+02	9.65E-01	0.387548
F11	-3.98E+02	6.99E-01	-3.99E+02	5.57E-01	0.472732
F12	-2.98E+02	6.52E-01	-3.00E+02	3.23E-01	1.151011
F13	-1.97E+02	1.12E+00	-2.00E+02	0.00E+00	0.756002
F14	-8.87E+01	9.77E+00	-9.11E+01	5.60E+00	0.083637
F15	1.13E+02	7.41E+00	1.08E+02	5.94E+00	0.201622
F16	2.03E+02	1.82E+00	2.00E+02	0.00E+00	0.468688
F17	3.03E+02	6.07E-01	3.03E+02	4.43E-01	-0.0237
F18	4.04E+02	8.73E-01	4.03E+02	6.82E-01	0.082222
F19	5.00E+02	2.84E-01	5.00E+02	5.78E-02	0.43412
F20	6.00E+02	1.89E-01	6.00E+02	8.80E-02	0.589632
F21	7.25E+02	1.88E+01	7.65E+02	1.47E+01	-0.69501
F22	8.31E+02	1.75E+01	8.22E+02	7.97E+00	0.167756
F23	9.44E+02	3.44E+01	9.84E+02	2.29E+01	-0.38725
F24	1.01E+03	2.19E+00	1.02E+03	7.88E+00	-1.96993
F25	1.12E+03	1.29E+01	1.17E+03	3.17E+01	-1.17709
F26	1.20E+03	2.88E+00	1.21E+03	2.62E+00	-0.26336
F27	1.43E+03	3.41E+01	1.40E+03	2.16E+01	0.295389
F28	1.45E+03	2.53E+01	1.48E+03	2.43E+01	-0.43603

B. Comparison between DPGA using k-point crossover and PSO

Table 3 shows the comparison of the results of DPGA using *k*-point crossover versus DPGA using discrete TPX. For higher dimensions the *k*-point crossover is found to be better than proposed discrete TPX except F2, F4, F14, F15, F16, F22, F24, F26 and F26.

Table 3. Comparison between K-point crossover and Discrete TPX for higher dimension

Fn. No.	K-Point Crossover		Discrete TPX		t-Test
	M	SD	M	SD	
F1	1.50E+05	1.21E+04	1.57E+05	1.20E+04	-0.19349
F2	5.83E+09	7.36E+08	5.19E+09	1.33E+09	0.286763
F3	9.84E+18	1.40E+19	1.29E+20	2.04E+20	-1.54493
F4	4.73E+09	2.36E+09	4.23E+05	1.55E+05	0.668036
F5	6.35E+04	1.44E+04	6.88E+04	1.91E+04	-0.1223
F6	2.28E+04	3.11E+03	2.86E+04	4.21E+03	-0.61451
F7	3.27E+06	2.51E+06	3.40E+06	2.88E+06	-0.01785
F8	-6.83E+02	4.82E+00	-6.79E+02	6.62E-02	-0.27857
F9	-5.43E+02	3.04E+01	-5.19E+02	1.94E+00	-0.26418
F10	2.24E+04	2.83E+03	2.36E+04	2.75E+03	-0.14357
F11	2.08E+03	3.14E+02	2.21E+03	1.51E+02	-0.1364
F12	1.99E+03	1.67E+02	2.08E+03	2.39E+02	-0.19098
F13	2.04E+03	1.66E+02	2.20E+03	1.60E+02	-0.3274
F14	1.60E+04	5.63E+02	1.58E+04	3.07E+02	0.105333
F15	1.70E+04	3.34E+02	1.66E+04	6.26E+02	0.377395
F16	2.07E+02	7.34E-01	2.00E+02	0.00E+00	3.08635
F17	4.92E+03	4.85E+02	5.45E+03	3.35E+02	-0.36572
F18	5.16E+03	4.01E+02	5.23E+03	5.40E+02	-0.05528
F19	2.04E+07	5.98E+06	2.50E+07	4.12E+06	-0.25786
F20	6.25E+02	0.00E+00	6.25E+02	0.00E+00	Inf
F21	1.23E+04	6.12E+02	1.31E+04	8.44E+02	-0.41176
F22	1.79E+04	4.27E+02	1.78E+04	4.21E+02	0.065736
F23	1.80E+04	2.53E+02	1.80E+04	2.86E+02	-0.07589
F24	1.52E+03	9.39E+01	1.48E+03	4.95E+01	0.116111
F25	1.51E+03	6.19E+00	1.51E+03	5.90E+00	-0.04611
F26	1.73E+03	1.08E+01	1.72E+03	7.90E+00	0.159713
F27	3.93E+03	6.03E+01	3.87E+03	8.80E+01	0.381049
F28	1.64E+04	1.56E+03	1.69E+04	1.71E+03	-0.10427

C. Comparison between DPGA using k-point crossover and PSO

Table 4 shows the results of comparison between PSO and DPGA using *k*-point crossover. The *k*-point crossover is found to be good for only F8 function. For all the other functions SPSO is better.

Table 4.Comparison between PSO[7] and DPGA using k-point crossover

Fn. No.	PSO [7]		K-Point Crossover		t-Test
	M	SD	M	SD	
F1	-1.40E+03	3.18E-13	1.50E+05	1.21E+04	-92.4752
F2	6.79E+05	1.87E+05	5.83E+09	7.36E+08	-58.6657
F3	4.37E+08	9.47E+08	9.84E+18	1.40E+19	-5.1836
F4	4.99E+04	8.72E+03	4.73E+09	2.36E+09	-14.8382
F5	-1.00E+03	5.41E-05	6.35E+04	1.44E+04	-33.2812
F6	-8.57E+02	2.41E+01	2.28E+04	3.11E+03	-52.2599
F7	-7.14E+02	1.53E+01	3.27E+06	2.51E+06	-9.64366
F8	-6.79E+02	4.25E-02	-6.83E+02	4.82E+00	5.279006
F9	-5.46E+02	6.74E+00	-5.43E+02	3.04E+01	-0.06841
F10	-5.00E+02	2.38E-01	2.24E+04	2.83E+03	-59.7295
F11	-1.70E+02	4.18E+01	2.08E+03	3.14E+02	-7.54086
F12	-6.52E+01	4.87E+01	1.99E+03	1.67E+02	-5.94846
F13	2.28E+02	6.22E+01	2.04E+03	1.66E+02	-4.10339
F14	7.16E+03	8.53E+02	1.60E+04	5.63E+02	-1.46975
F15	8.02E+03	1.14E+03	1.70E+04	3.34E+02	-1.10923
F16	2.02E+02	3.87E-01	2.07E+02	7.34E-01	-1.7531
F17	6.11E+02	6.62E+01	4.92E+03	4.85E+02	-9.12257
F18	6.91E+02	6.24E+01	5.16E+03	4.01E+02	-10.0598
F19	5.37E+02	1.20E+01	2.04E+07	5.98E+06	-25.1922
F20	6.23E+02	1.19E+00	6.25E+02	0.00E+00	-0.27242
F21	1.54E+03	3.04E+02	1.23E+04	6.12E+02	-5.01516
F22	9.72E+03	1.40E+03	1.79E+04	4.27E+02	-0.82251
F23	1.13E+04	1.35E+03	1.80E+04	2.53E+02	-0.70149
F24	1.34E+03	1.69E+01	1.52E+03	9.39E+01	-1.44154
F25	1.50E+03	2.05E+01	1.51E+03	6.19E+00	-0.07841
F26	1.63E+03	9.06E+01	1.73E+03	1.08E+01	-0.15822
F27	2.98E+03	1.64E+02	3.93E+03	6.03E+01	-0.82792
F28	1.80E+03	1.30E+03	1.64E+04	1.56E+03	-1.57995

D. Comparison between DPGA using Discrete TPX and PSO[7]

Table 5 represents the comparison between the results of DPGA using discrete TPX and the PSO [7]. Discrete TPX is found to be good for only F16 function. For all the other functions the PSO is better than DPGA.

Table 5. Comparison between DPGA using Discrete TPX and PSO

Fn. No.	PSO [7]		Discrete TPX		t-Test
	M	SD	M	SD	
F1	-1.40E+03	3.18E-13	1.57E+05	1.20E+04	-97.669
F2	6.79E+05	1.87E+05	5.19E+09	1.33E+09	-28.9258
F3	4.37E+08	9.47E+08	1.29E+20	2.04E+20	-4.67981
F4	4.99E+04	8.72E+03	4.23E+05	1.55E+05	-5.72792
F5	-1.00E+03	5.41E-05	6.88E+04	1.91E+04	-27.046
F6	-8.57E+02	2.41E+01	2.86E+04	4.21E+03	-49.7189
F7	-7.14E+02	1.53E+01	3.40E+06	2.88E+06	-8.74376
F8	-6.79E+02	4.25E-02	-6.79E+02	6.62E-02	-0.80331
F9	-5.46E+02	6.74E+00	-5.19E+02	1.94E+00	-0.57387
F10	-5.00E+02	2.38E-01	2.36E+04	2.75E+03	-64.9629
F11	-1.70E+02	4.18E+01	2.21E+03	1.51E+02	-8.03473
F12	-6.52E+01	4.87E+01	2.08E+03	2.39E+02	-6.21523
F13	2.28E+02	6.22E+01	2.20E+03	1.60E+02	-4.47684
F14	7.16E+03	8.53E+02	1.58E+04	3.07E+02	-1.44028
F15	8.02E+03	1.14E+03	1.66E+04	6.26E+02	-1.06132
F16	2.02E+02	3.87E-01	2.00E+02	0.00E+00	0.731805
F17	6.11E+02	6.62E+01	5.45E+03	3.35E+02	-10.3047
F18	6.91E+02	6.24E+01	5.23E+03	5.40E+02	-10.15
F19	5.37E+02	1.20E+01	2.50E+07	4.12E+06	-44.8977
F20	6.23E+02	1.19E+00	6.25E+02	0.00E+00	-0.27242
F21	1.54E+03	3.04E+02	1.31E+04	8.44E+02	-5.36642
F22	9.72E+03	1.40E+03	1.78E+04	4.21E+02	-0.81399
F23	1.13E+04	1.35E+03	1.80E+04	2.86E+02	-0.70756
F24	1.34E+03	1.69E+01	1.48E+03	4.95E+01	-1.1733
F25	1.50E+03	2.05E+01	1.51E+03	5.90E+00	-0.08436
F26	1.63E+03	9.06E+01	1.72E+03	7.90E+00	-0.1501
F27	2.98E+03	1.64E+02	3.87E+03	8.80E+01	-0.76773
F28	1.80E+03	1.30E+03	1.69E+04	1.71E+03	-1.63318

V. Conclusion

DPGA is a diversity based technique using two populations. Two crossover operators are experimented on CEC'2013 problem set. DPGA can successfully solve the CEC'2013 problems with smaller dimensions but it is observed that the algorithm is suffering by the curse of dimensionality i.e. as the dimension increases from 2 to 50 the algorithm loses its accuracy to find the optimum solution.

DPGA using discrete TPX and it found better than k-point crossover only for the problems having lower dimensions. But for higher dimension problems k-point crossover is able to maintain consistency for obtaining an optimal solution. On the basis of t-test evaluation, the results of both types of the crossover are also compared with the results of PSO on the same problem. PSO has found better than DPGA to solve the functions of CEC'2013.

In the future, the results of the DPGA algorithm for CEC'2013 could be improved by using better survival selection and crossover, mutation operators.

Further, DPGA performance can be improved by adding more reserve populations. A performance comparison of DPGA could be done with other metaheuristic. DPGA can be tested on the latest test bed of CEC.

ACKNOWLEDGMENT

We express our sincere thanks to all the authors, whose papers in the area of Dual Population EA and published in various conference proceedings and journals.

REFERENCES

[1] T. Park, K. Ruy, "A Dual-Population Genetic Algorithm for Adaptive Diversity Control", *IEEE transactions on evolutionary computation*, vol. 14, no. 6, 2010 DOI:10.1109/TEVC.2010.2043362.

[2] A.J. Umbarkar, M.S. Joshi, P.D. Sheth, "Diversity-Based Dual-Population Genetic Algorithm (DPGA): A Review" *In Proceedings of Fourth International Conference on Soft Computing for Problem Solving*, (AISC, vol 335). 2015. DOI https://doi.org/10.1007/978-81-322-2217-0_19

[3] T. Park and K. Ruy, "A dual population genetic algorithm with evolving diversity", in *Proc. IEEE Congr. Evol. Comput.*, pp. 3516–3522, 2007. DOI: 10.1109/CEC.2007.4424928.

[4] T. Park and K. Ruy, "Adjusting population distance for dual-population genetic algorithm," in *Proc. of Aust. Joint Conf. Artif. Intell.*, pp. 171–180, 2007. DOI: 10.1109/TEVC.2010.2043362.

[5] T. Park, R. Choe, K. Ruy, "Dual-population Genetic Algorithm for Nonstationary Optimization", in Proc. GECCO'08 ACM, 2008, pp.1025-1032. DOI: 10.1145/1389095.1389286.

[6] A. Umbarkar, M. Joshi, "Dual Population Genetic Algorithm (GA) versus OpenMP GA for Multimodal Function Optimization", *International Journal of Computer Applications, 64(19):29-36*, February 2013. DOI: 10.5120/10744-5516.

[7] M. Zambrano-Bigiarini, M. Clerc and R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements," *Evolutionary Computation (CEC), 2013 IEEE Congress on* , vol., no., pp.2337,2344, 20-23 June 2013. doi: 10.1109/CEC.2013.6557848.

[8] M. Mehra, M. Jayalal, A. John Arul, S. Rajeswari, K. Kuriakose, S. Murty, "Study on Different Crossover Mechanisms of Genetic Algorithm for Test Interval Optimization for Nuclear Power Plants", IJISA, vol.6, no.1, 2014, pp.20-28. DOI: 10.5815/ijisa.2014.01.03.

[9] J. Liang, B. Qu, P. Suganthan, A. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization", Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 2013, vol. 201212.

[10] S. Elsayed, R. Sarker and D. Essam, "A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization," *Evolutionary Computation (CEC), 2013 IEEE Congress on* , vol., no., pp.356,360, 20-23 June 2013. DOI: 10.1109/CEC.2013.6557591.

[11] C. Worasucheep, "A Particle Swarm Optimization with Diversity-Guided Convergence Acceleration and Stagnation Avoidance", *in Proc. of 8th International Conference on Natural Computation (ICNC 2012)*, pp.733-738, 2012. DOI: 10.1109/ICNC.2012.6234647.

[12] B. Qu, P. Suganthan, "Constrained Multi-Objective Optimization Algorithm with Diversity Enhanced Differential Evolution", *IEEE conference on evolutionary computation (CEC)*, 2010. DOI: 10.1109/CEC.2010.5585947.

Authors' Profiles

A. J. Umbarkar is presently working as an Assistant Professor in Information Technology at Walchand College of Engineering, at Sangli, MS, India. He has received his Bachelor of Engineering (BE) in Computer Engineering from PICT, Pune, MS, India, his Master of Engineering (ME) from Computer Science and Engineering (CSE) from Walchand College of Engineering, at Sangli, MS, India and received PhD from Government College Engineering, Aurangabad, from University: BAMU Aurangabad, MS, India. He has 15 years of teaching experience. His research interests include Parallel Genetic Algorithms, Parallel Evolutionary Algorithms and Parallel programming. He has published about 32 research papers in Conferences and Journals.

Latik R. Moon has completed Bachelor of Technology (B. Tech. 2013) in Computer Science and Engineering from SGGSIE&T, Nanded, MS, India; and Master of Technology (M. Tech. 2015) in Information Technology from Walchand College of Engineering, Sangli, MS, India. His research interests are Parallel Programming, High Performance Computing, GPU Programming and Evolutionary Algorithms.

Pranali D. Sheth has completed Bachelor of Technology (B. Tech- 2007) and Master of Technology (M. Tech - 2014) in Information Technology from Walchand College of Engineering, Sangli, MS, India. Her research interests are Parallel Evolutionary Programming, High Performance Computing, Data Mining and Image Processing.

How to cite this paper: A. J. Umbarkar, L. R. Moon, P. D. Sheth," Comparative Study of CEC'2013 Problem Using Dual Population Genetic Algorithm", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.10, No.5, pp. 40-45, 2018. DOI: 10.5815/ijieeb.2018.05.06