*I.J. Intelligent Systems and Applications*, **2015, 02, 34-40**
Published Online January 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.02.05

# Dual Population Genetic Algorithm for Solving Constrained Optimization Problems

**A. J. Umbarkar**
Department of Information Technology, Walchand College of Engineering, Sangli, 416-416, India
Email: anantumbarkar@rediffmail.com

**M. S. Joshi**
Department of Computer Engineering, Jawaharlal Nehru Engineering College, Aurangabad, India
Email: madhuris.joshi@gmail.com

**P. D. Sheth**
Department of Information Technology, Walchand College of Engineering, Sangli, 416-416, India
Email: pranalisheth@gmail.com

***Abstract*—**Dual Population Genetic Algorithm is an effective optimization algorithm that provides additional diversity to the main population. It addresses the premature convergence problem as well as the diversity problem associated with Genetic Algorithm. Thus it restricts their individuals to be trapped in the local optima. This paper proposes Dual Population Genetic Algorithm for solving Constrained Optimization Problems. A novel method based on maximum constrains satisfaction is applied as constrains handling technique and Dual Population Genetic Algorithm is used as meta-heuristic. This method is verified against 9 problems from Problem Definitions and Evaluation Criteria for the Congress on Evolutionary Computation 2006 Special Session on Constrained Real-Parameter Optimization problem set. The results are compared with existing algorithms such as Ant Bee Colony Algorithm, Differential Evolution Algorithm and Genetic Algorithm that have been used for solving same problem set. Analysis shows that this technique gives results close to optimum value but fails to obtain exact optimum solution. In future Dual Population Genetic Algorithm can produce more efficient solutions using alternative constrains handling technique.

***Index Terms*—**Dual Population Genetic Algorithm, DPGA, Population Diversity, Metaheuristic Algorithms, Function Optimization, Constrained Optimization Problems, COPs.

## I. Introduction

Genetic algorithms (GAs) are population based search algorithms that obtain solution to optimization and search problems. The problem associated with GAs is that as the populations evolve, they lose diversity and their individuals are trapped in local optima, especially when involving complex problems which have a lot of peaks in the fitness landscape [1]. Literature terms this problem as premature convergence.

One of the proposed solutions for above problem is DPGA. DPGA uses a reserve population along with the main population. This provides additional diversity to the main population. Information exchange between the populations takes place through inter-population crossbreeding. This technique helps to solve the problem of premature convergence [1].

DPGA is used to solve optimization problems. Optimization problems can be classified into Unconstrained Optimization Problems (UOPs) as well as Constrained Optimization Problems (COPs). COPs are encountered in allocation, economics, location, VLSI, engineering and structural design problems [2]. If resources can be made available without any limitation, then there is no limit to the profit that can be achieved. However, in the face of real world complications, resources are most likely to be limited in the form of constraints imposed upon the optimization function. What constitute the difficulties of the constrained optimization problem are the various limits on the decision variables, the constraints involved, the interference among constraints, and the interrelationship between the constraints and the objective function, mass of memory and computation cost [3].

Evolutionary Algorithms (EAs) have been found effective in the solution of a wide variety of optimization and search problems. But EAs are unconstrained search technique therefore incorporating constraints into the fitness function of an EA is an open research area. There is wide scope available for the research regarding mechanisms that allow EAs to deal with equality and inequality constraints [4].

This paper applies a novel method to solve the above research problem. It solves COPs by applying Maximum Constrains Satisfaction Method (MCSM), using DPGA which is an EA. MCSM is a novel technique which tries to satisfy maximum constrains first and then it attempts to optimize objective function.

Section II gives a brief literature review of DPGA and COPs. Section III describes DPGA with implementation details such as fitness function for reserve population and evolutionary process. Section IV presents experimental results and discussion. Section V gives some conclusions and exhibits future scope.

## II. Literature Review

Dual Population Genetic Algorithm for solving COPs is new in the area of evolutionary algorithms for solving optimization problems. Therefore we have extensively searched in literature about evolution of DPGA. We have also studied nature of COPs and searched methods that solved COPs till date. We have provided brief literature review on DPGA followed by literature review on COPs.

Park and Ruy (2006) [5] proposed DPGA. It has two distinct populations with different evolutionary objectives: The prospect (main) population works like population of the regular genetic algorithm which aims to optimize objective function. The preserver (reserve) population helps to maintain the diversity. Park and Ruy (2007) [6] introduced DPGA-ED that is an improved design-DPGA. Unlike DPGA, the reserve population of DPGA-ED evolves by itself. Park and Ruy (2007) [7] proposed a method to dynamically adjust the distance between the populations using the distance between good parents. Park and Ruy (2007) [8] exhibited DPGA2 that uses two reserve populations. Park and Ruy (2010) [1] experimented DPGA on various classes of problems using binary, real-valued, and order-based representations. Umbarkar and Joshi (2013) [9] compared DPGA with OpenMP GA for Multimodal Function Optimization. The results show that the performance of OpenMP GA better than SGA on the basis of execution time and speed up.

The basic and classical constrained optimization methods include penalty function method, Lagrangian method [10] and Sequential Quadratic Programming (SQP) [11]. These are local search methods which can find a local optimal solution.

Recent trend is to make use of evolutionary algorithms to solve constrained optimization problems [12, 13]. Compared with the traditional nonlinear programming approach, evolutionary algorithms need less information such as gradient (derivatives), as well as it is a global searching approach. According to Koziel and Michalewicz [14], these algorithms can he grouped into four categories:

i. Methods based on preserving feasibility of solutions by transforming infeasible solutions to feasible solutions with some operators [15]
ii. Methods based on penalty functions which introduce a penalty term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained problem [16]
iii. Methods which make a clear distinction between feasible and infeasible solutions [17]
iv. Methods based on evolutionary algorithms [18, 19, 20]
v. other hybrid methods combining evolutionary computation techniques with deterministic procedures for numerical optimization [21]

## III. Dual Population Genetic Algorithm

DPGA starts with two randomly generated populations, the main population and the reserve population. The individuals of each population are evaluated by their own fitness functions. The evolution of each population is obtained by inbreeding between parents from the same population, crossbreeding between parents from different populations, and survival selection among the obtained offspring [1]. In the following paragraphs, the fitness function used for the reserve population, evolutionary process of simple DPGA and methodology applied by this paper are described

### A. Fitness Function for Reserve Population

The objective function of the problem works as the fitness function of the main population. Fitness function for the reserve population is defined in another way. An individual in the reserve population is given a high fitness value if its average distance from each of the individuals of the main population is large. Therefore the reserve population can provide the main population with additional diversity.

In the equation (1) of the fitness function, each individual of the reserve population can maintain a given distance $\delta$ from the individuals of the main population [1, 5, 6].

$$fr_{\delta}(x) = 1 - |\delta - d(M, x)| \qquad (1)$$

Where,

$d\ (M, x)$: average distance $(0 \le d(M, x) \le 1)$ between the main population M and individual $x$ of the reserve population

$\delta$: desired distance $(0 \le \delta \le 1)$ between two population

Assuming a binary representation for a chromosome, $d\ (M, x)$ is calculated using equation (2) [1, 5]

$$d(M, x) = \frac{1}{|M|}\sum_{m \in M} hd(m, x) = \frac{1}{l}\sum_{k=1}^{l} |\ f_{M,k} - x_k\ | \qquad (2)$$

Where,

$n$: size of main population

$m_i$: $i$th chromosome of the main population

$hd(m_{i,} x)$: Hamming distance between two chromosome vectors $m$ *and* $x$

$l$ : length of chromosome

$m_{i,\,k}$: $k$th gene on chromosome $m_{i\ and}\ x_k$

$f_{M,k}$: frequency of the $k$th gene value *'1'* of the main population M

$x_i$: frequency of $k$th gene value *'1'* of the chromosome vector x and is identical to the $k$th gene value of the chromosome

The definitions of $d(M, x)$ and δ enable $fr_{\delta}^{(x)}$ to take a maximum value of 1 when the distance from $x$ to the main population is $\delta$ and decreases linearly as $d(M, x)$ deviates from $\delta$ [1].

### B. Evolutionary Process

Firstly, the algorithm selects four parents - two from the main population and two from the reserve population. The parent selection from the main population depends on fitness value calculated by the objective function whereas parent selection from the reserve population depends on fitness value calculated by equation (1) [3].

From four parents, the algorithm generates six offspring by two inbreeding and one crossbreeding processes. Two offspring are generated by mating two parents selected from the main population; another two offspring are generated by mating two parents selected from the reserve population, and yet another two offspring are generated by mating one parent selected from the main population and one parent selected from the reserve population. The above procedure of generation of the new population is represented in the diagrammatic form in fig.1 [1, 6].

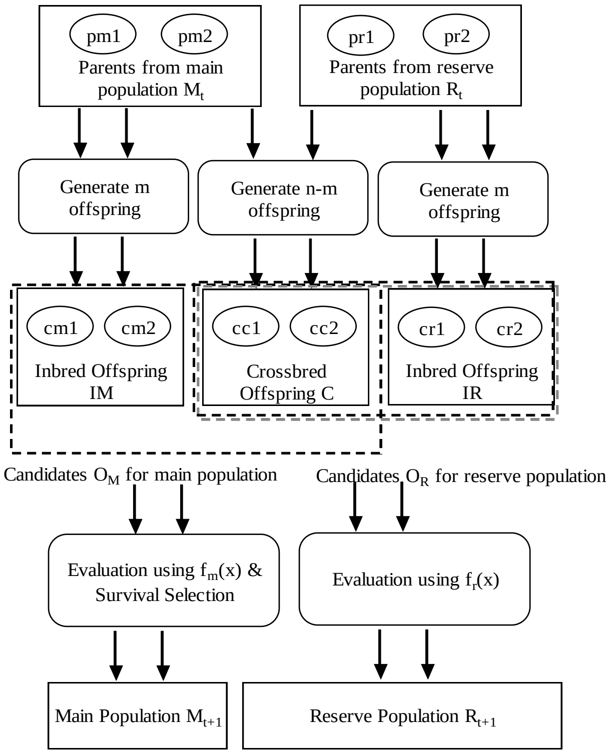
pm1
pm2
Parents from main population $M_t$
pr1
pr2
Parents from reserve population $R_t$
Generate m offspring
Generate n-m offspring
Generate m offspring
cm1
cm2
Inbred Offspring IM
cc1
cc2
Crossbred Offspring C
cr1
cr2
Inbred Offspring IR
Candidates $O_M$ for main population
Candidates $O_R$ for reserve population
Evaluation using $f_m(x)$ & Survival Selection
Evaluation using $f_r(x)$
Main Population $M_{t+1}$
Reserve Population $R_{t+1}$

Fig. 1. Procedure for producing new populations

### C. Methodology

The methodology for applying DPGA for COPs is described in this section. A detailed pseudo code is explained in fig.2 entitled DPGA_MCSM. MCSM is a novel technique. It is based on Deb's rules that in the category of methods searching for feasible solutions, any feasible solution is better than any infeasible one [20]. According to the first phase of MCSM, Pseudo Random Number Generator (PRNG) selects variables to prepare individuals which satisfy all constrains. As per the second phase of MCSM, DPGA meta-heuristic is applied for evolution of both populations via inbreeding. DPGA evolves till stopping criteria are met.

The algorithm starts with iinitialization of main population $M_0$, reserve population $R_0$, crossover rate, elitism rate, mutation rate, crossbreeding rate, tour size for tournament selection and max generation $t_{max.}$ The main population of size m and reserve populations of size n, where n > m are initialized using PRNG provided by stdlib.h of C programming language. We check satisfaction of all constraints for current COP in initialization. Both the populations initialize using same constraints on variables.

Firstly, main population undergoes through traditional evolution cycle of Genetic Algorithm. In the first Step I initialized population encoded using IEEE 754 algorithm from decimal value representation to binary value representation. In Step II fitness of main as well as reserve population is calculated using objective function and special fitness function (1), (2) for reserve population.

Step III and IV perform inbreeding of main and reserve population. For selection of the parents, individuals of the main population are sorted according to fitness value. Then according to crossover rate appropriate number of parents gets selected for reproduction. Multipoint crossover method is then used for generating new population called intermediate main population. For evolution of reserve population tournament selection is used. Multipoint crossover method is used to generate intermediate reserve population. Flip bit mutation is used to provide diversity to the intermediate main population and the intermediate reserve population.

In Step V DPGA performs crossbreeding between different populations. Offspring of size (n-m) is generated via crossbreeding on the main population and the reserve population. In crossbreeding (n-m) numbers of best candidates are selected from both populations. Intermediate main population, intermediate reserve population and offspring constitute a candidate set for the next generation of the main population and reserve population. Step VI decodes both populations using IEEE 754 algorithm from binary value representation to decimal value representation. Step VII evaluates generated candidate sets using objective function and fitness function for reserve population. Fitness values are used for survival selection in Step VIII using truncation mechanism. Since m is the number of inbred offspring, the crossbred offspring can survive only when they are better than at least one of the inbred offspring [1].

DPGA evolves till maximum number of generations specified are met or algorithm achieves optimal solution of current COP. These conditions work as stopping criteria for algorithm. Optimum value generated over the given maximum generations is taken as optimal solution for that run.

## IV. Experimental Results and Discussion

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

Standard problems are taken for experiments from "Problem Definitions and Evaluation Criteria for the Congress on Evolutionary Computation 2006 Special Session on Constrained Real-Parameter Optimization problem" (CEC 2006) [22]. In this report, 24 benchmark functions are described. Guidelines for conducting experiments, statistical parameters and its formula, performance evaluation criteria are given at the end of this report.

This test set contains following function characteristics that make our study reliable and robust-

1. Multimodal functions to verify the reliability of the algorithms
2. Scalable functions
3. High dimensional functions

```
Procedure DPGA_MCSM
begin
        Initialize main population $M_0$, reserve population $R_0$, crossover rate, elitism rate, mutation rate,
        crossbreeding rate, tour size, max generation $t_{max}$
        Initialize $M_0$ of size $m$
        Initialize $R_0$ of size $n$, $n>m$
        t: = 0
        Repeat
         Step I: Encode both population using IEEE 754 algorithm from decimal value representation to binary
               value representation
         Step II: Fitness Calculation
           a.  Evaluate $M_0$ using objective function $fm(x)$
           b.  Evaluate $R_0$ using fitness function for reserve population as per equation (1)  $fr(x)$
        Step III: Inbreeding of main population
           a.  Selection – using sorting select best fit parents from $M_t$
           b.  Crossover – using multipoint crossover method generate intermediate main population $O_m$
           c.  Mutation – flip bit mutation is used to add population diversity to $O_m$
        Step IV: Inbreeding of reserve population
           a.  Selection – using tournament selection select best fit parents from $R_t$
           b.  Crossover – using multipoint crossover method generate intermediate reserve population $O_r$
           c.  Mutation – flip bit mutation is used to add population diversity to $O_r$
        Step V: Crossbreeding
           a.  Offspring C of size (n-m) using best individuals from $O_m$ and $O_r$
           b.  Make $I_m$ = C U $O_m$ and $I_r$ = C U $O_r$
        Step VI: Decoding: Decode both population using IEEE 754 algorithm from binary value representation
                to decimal value representation
        Step VII: Evaluation
           a.  Evaluate $I_m$ using $fm(x)$
           b.  Evaluate $I_r$ using $fr(x)$
         Step VIII: Survival selection from $I_m$ of size m and from $I_r$ of size n
        t  = t + 1
        Until
         $fm(x)$ > = global optimal value or  t > $t_{max}$
End
Where, t : index of current generation                 $M_0$, $R_0$: main, reserve population
        $fm(x)$: objective function                      $O_m$, $O_r$: intermediate main, reserve population respectively
        $fr(x)$: fitness function for reserve population   $I_m$, $I_r$: constitute set of main, reserve population
respectively
```

Fig. 2. Pseudo code of DPGA_MCSM

Table I describes ten functions from CEC 2006 that we have implemented using DPGA_MCSM. It describes function name, dimension (D), type of function, no. of linear inequality constrains (LI), no. of non-linear inequality constrains (NI), no. of linear equality constrains (LE) and no. of non-linear equality constrains (NE).

Table II describes the parameter settings used for experimentation. Above parameter values are used for all functions. Consecutive 30 runs are taken for each function these parameter values. Size of main population is taken 100 where the size of reserve population size is taken 200. Crossover rate, elitism rate, mutation rate and crossbreed rate are kept same for both the main population and the reserve population.

Table III exhibits the optimal values found for the some functions using Ant Bee Colony Algorithm (ABC), Differential Evaluation (DE), and Genetic Algorithm (GA). ABC, DE, GA find exact optimum solution in 2, 40,000 function evaluations [37] where DPGA_MCSM gives solution close to optimal value in just 10, 000 average function evaluations.

Table IV shows statistical results obtained by DPGA _MCSM algorithm for 9 test functions from CEC 2006 over 30 independent runs using average 10,000 function evaluations. It contains function name, global optimum value, best value, mean value and worst value, standard deviation (S.D.) and standard error of mean (S.E.M.) obtained in 30 independent runs.

Table 1. Function Description

| Functi-on | D | Type | LI | NI | LE | NE |
|---|---|---|---|---|---|---|
| g01 | 13 | Quadratic | 9 | 0 | 0 | 0 |
| g02 | 20 | Nonlinear | 0 | 2 | 0 | 0 |
| g04 | 5 | Quadratic | 0 | 6 | 0 | 0 |
| g06 | 2 | Cubic | 0 | 2 | 0 | 0 |
| g07 | 10 | Quadratic | 3 | 5 | 0 | 0 |
| g08 | 2 | Nonlinear | 0 | 2 | 0 | 0 |
| g09 | 7 | Polynomial | 0 | 4 | 0 | 0 |
| g10 | 8 | Linear | 3 | 3 | 0 | 0 |
| g18 | 9 | Quadratic | 0 | 13 | 0 | 0 |
| g24 | 2 | Linear | 0 | 2 | 0 | 0 |

Table 2. Parameter Settings

| | | | |
|---|---|---|---|
| Crossover Rate | 0.80 | Elitism Rate | 0.20 |
| Mutation Rate | 0.09 | Crossbreed Rate | 0.10 |
| Main Pop. Size | 100 | Reserve Pop. Size | 200 |

Table 3. Optimal solutions for other algorithms

| Function | GA [20] | DE [20] | ABC [20] |
|---|---|---|---|
| g01 | -14.23 | −14.55 | −15.00 |
| g04 | -30590.45 | −30665.54 | −30665.53 |
| g06 | −6872.20 | – | −6961.81 |
| g07 | 34.98 | 24.31 | 24.47 |
| g08 | 0.096 | 0.096 | 0.095 |
| g09 | 692.06 | 680.63 | 680.64 |
| g10 | 10003.22 | 7147.33 | 7224.40 |

Table 4. Results of DPGA_MCSM for consecutive 30 runs

| Function | Global | Best | Mean | Worst | S.D. | S.E.M |
|---|---|---|---|---|---|---|
| g01 | -15.00 | -15.00000 | -10.87479 | -7.71976 | 1.58263 | 0.28895 |
| g04 | -30665.538671 | -30649.49587 | -30490.11261 | -30234.17525 | 108.74284 | 19.85364 |
| g06 | -6961.813875 | -6960.15972 | -6691.46685 | -6352.67120 | 142.49572 | 26.01604 |
| g07 | 24.306209 | 74.522052 | 127.71353 | 194.30380 | 30.64218 | 5.594 |
| g08 | -0.095825 | -0.094837 | -0.08918 | -0.07323 | 0.00459 | 0.00084 |
| g09 | 680.630057 | 677.228267 | 663.23351 | 602.43395 | 16.99565 | 3.10297 |
| g10 | 7049.24802 | 7013.121872 | 5876.105700 | 4667.755859 | 566.19879 | 103.37328 |
| g18 | -0.866025 | -0.835853 | -0.759260 | -0.622790 | 0.05414 | 0.00989 |
| g24 | -5.508013 | -5.508018 | -5.430250 | -5.372550 | 0.06961 | 0.01271 |

In order to analyze on which specific kind of problem, the DPGA_MCSM algorithm performs better or not, the results should be examined. DPGA_MCSM cannot find solution to the problems having equality constrains. It gives best solution almost equal to global optimum value for g01, g06 and g24 whereas algorithm gives best solution close to global optimum value for g04, g08, g09, g10 and g18. DPGA_MCSM cannot optimize g07.

## V. Conclusion

This paper proposes a novel technique for solving COPs using DPGA. The DPGA_MCMS algorithm initializes individuals satisfying all constrains. DPGA meta-heuristic is applied on these individuals and then several generations of DPGA optimize the objective function. This algorithm can solve problems that have only inequality constrains. Though the DPGA_MCMS algorithm fails to produce exact global optimal value, it succeeds in finding optimal solutions closer to global optima.

Though the DPGA_MCMS algorithm slightly deviates from global optima in future different constrains handling technique can effectively solve COPs using DPGA.

## Appendix A.

**g01:** Minimize

$$f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} {x_i}^2 - \sum_{i=5}^{13} x_i$$

Subject to

$g_1(\vec{x}) = 2x_1 + 2x_2 + 2x_{10} + 2x_{11} - 10 \leq 0$
$g_1(\vec{x}) = 2x_1 + 2x_2 + 2x_{10} + 2x_{11} - 10 \leq 0$
$g_2(\vec{x}) = 2x_1 + 2x_3 + 2x_{10} + 2x_{12} - 10 \leq 0$
$g_3(\vec{x}) = 2x_2 + 2x_3 + 2x_{11} + 2x_{12} - 10 \leq 0$
$g_4(\vec{x}) = -8x_1 + x_{10} \leq 0$
$g_5(\vec{x}) = -8x_2 + x_{11} \leq 0$
$g_6(\vec{x}) = -8x_3 + x_{12} \leq 0$
$g_7(\vec{x}) = 2x_4 - x_5 + x_{10} \leq 0$
$g_8(\vec{x}) = 2x_6 - x_7 + x_{11} \leq 0$
$g_9(\vec{x}) = 2x_8 - x_9 + x_{12} \leq 0$

where the bounds are 0 <= xi <= 1 (i=1,……,9), 0 <= xi <= 100 (i = 10, 11, 12) and 0 <= x13<=1 .

**g04:** Minimize

$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1 + x_5 + 37.293239x_1$
$-40792.141$

Subject to

$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4$

$$+0.0022053x_3x_5 - 92 \leq 0$$
$$g_2(\vec{x}) = -85..334407x_1 - 0.0056858x_2x_5$$
$$-0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$$
$$g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2$$
$$-0.0021813x_3^2 - 110 \leq 0$$
$$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2$$
$$-0.0021813x_3^2 + 90 \leq 0$$
$$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3$$
$$+0.0019085x_3x_4 - 25 \leq 0$$
$$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3$$
$$-0.0019085x_3x_4 + 20 \leq 0$$

where 78 <= $x_1$<= 102, 33 <= $x_2$<= 45 and 27 <= $x_i$<= 45 (i = 3, 4, 5).

**g06:** Minimize

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Subject to

$$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$
$$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$

where 13 <= x1<= 100 and 0 <= x2<= 100.

**g07:** Minimize

$$f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2$$
$$+4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2$$
$$+7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

Subject to

$$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$$
$$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$
$$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$$
$$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$$
$$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$$
$$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$$
$$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$$
$$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

where -10 <= xi<= 10 (i = 1,…., 10).

**g08:** Minimize

$$f(\vec{x}) = -\frac{sin^3(2\pi x_1)sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

Subject to

$$g_1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0$$
$$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$$

where 0 <= x1<=10 and 0 <= x2<= 10.

**g09:** Minimize

$$f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2$$
$$+x_5^7 - 4x_6x_7 - 10x_6 - 8x_7$$

Subject to

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$$
$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$$
$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$$
$$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

where- 10 <= xi<= 10 for (i = 1,….., 7).

**g10:** Minimize

$$f(\vec{x}) = x_1 + x_2 + x_3$$

Subject to

$$g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$$
$$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$$
$$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$$
$$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$$
$$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$$
$$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$$

where 100 <= $x_1$<= 10000, 1000 <= $x_i$<= 10000 (i = 2,3) and 10 <= $x_i$<= 1000 (i = 4,…,8).

**g18:** Minimize

$$f(\vec{x}) = -0.5\begin{pmatrix} x_1 x_4 - x_2 x_3 + x_3 x_9 - \\ x_5 x_9 + x_5 x_8 - x_6 x_7 \end{pmatrix}$$

Subject to

$$gg1\ (\vec{x}) = x2 + x2 - 1 \leq 0$$
$$gg_6\ (\vec{x}) = (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0$$
$$gg_7\ (\vec{x}) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \leq 0$$
$$gg_8\ (\vec{x}) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leq 0$$
$$gg_9\ (\vec{x}) = x^2 + (x_8 - x_9)^2 - 1 \leq 0$$
$$gg_{10}\ (\vec{x}) = x_2 x_3 - x_1 x_4 \leq 0$$
$$gg_{11}\ (\vec{x}) = -x_3 x_9 \leq 0$$
$$gg_{12}\ (\vec{x}) = x_5 x_9 \leq 0$$
$$gg_{13}\ (\vec{x}) = x_6 x_7 - x_5 x_8 \leq 0$$

where the bounds are $-10 \leq xi \leq 10$ ($i = 1, . . . , 8$) and $0 \leq x9 \leq 20$.

**g24:** Minimize

$$f(\vec{x}) = -x_1 - x_2$$

Subject to

$$g_1(\vec{x}) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \leq 0$$
$$g_2(\vec{x}) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \leq 0$$

where the bounds are 0 <= x1<= 3 and 0 <= x2<= 4.

## ACKNOWLEDGMENT

We express our sincere thanks to the all authors, whose papers in the area of DPGA and COPs are published in various conference proceedings and journals. Our special thanks to T. Park, R. Choe and K.R. Ryu for their work on Dual Population Genetic Algorithm published in various journals and conferences.

## 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] K. Tang, K. Mang et.al, “Genetic Algorithm and their applications”, *IEEE Signal Processing Magazine*, pp.213-216, 1996. DOI: 10.1109/79.543973

[3] Applications of Genetic Algorithms, http://www.informatics.indiana.edu/ fil/CAS/PPT/Davis/sld029.html, 12th August, 2013.

[4] http://www.scribd.com/doc/211328880/Ga-Constraint, 31st August, 2014.

[5] T. Park, K. Ruy, “A Dual-Population Genetic Algorithm for Balance Exploration and Exploitation”, *Acta Press Computational Intelligence*, 2006.

[6] 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.

[7] 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.

[8] 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

[9] 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

[10] D. Luenberger and Y. Ye, “Linear and nonlinear programming third edition”, New York, NY: Springer, 2007. ISBN 978-0-387-74503-9

[11] P. Boggs and J. Tolle, “Sequential quadratic programming,” Acta Numerica, vol.4, no.1, pp.1-51, 1995.

[12] C. Coello Coello, “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art,” Comput. Methods Appl. Mech. Eng., vol.191, no.11-12, pp.1245-1287, Jan. 2002. DOI: 10.1016/j.bbr.2011.03.031.

[13] H. Lu and W. Chen, “Self-adaptive velocity particle swarm optimization for solving constrained optimization problems”, J. Global Optimiz., vol.41, no.3, pp.427-445, Jul. 2008. DOI: 10.1007/s10898-007-9255-9.

[14] S. Koziel, Z. Michalewicz, “Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization”, Evolutionary Computation, vol.7, no.1, pp.19–44, 1999. DOI: 10.1162/evco.1999.7.1.19

[15] Z. Michalewicz, C.Z. Janikow, “Handling constraints in genetic algorithms, in: R.K. Belew, L.B. Booker (Eds.)”, In Proc. of the Fourth International Conference on Genetic Algorithms (ICGA-91), San Mateo, California, University of California, San Diego, Morgan Kaufmann Publishers, pp. 151–157, 1991.

[16] J. Joines, C. Houck, “On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs, in: D. Fogel (Ed.)”, In Proc. of the first IEEE Conference on Evolutionary Computation, IEEE Press, Orlando, Florida, pp. 579–584, 1994.

[17] K. Deb, “An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering”, vol.186, no.2–4, pp. 311–338, 2000. DOI: 10.1016/S0045-7825(99)00389-8.

[18] I. Sardou, M. Ameli, M. Sepasian, M. Ahmadian, “A Novel Genetic-based Optimization for Transmission Constrained Generation Expansion Planning", IJISA, vol.6, no.1, pp.73-83, 2014. DOI: 10.5815/ijisa.2014.01.09

[19] 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

[20] D. Karaboga, B. Akay, “A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems”, Applied Soft Computing, vol.11, no.3, pp. 3021-3031, 2011. DOI: 10.1016/j.asoc.2010.12.001

[21] J. Paredis, “Co-evolutionary constraint satisfaction”, In Proc. of the 3rd Conference on Parallel Problem Solving from Nature, Springer-Verlag, NewYork, 1994, pp. 46–55. DOI: 10.1007/3-540-58484-6_249

[22] J. Liang, T. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. Coello Coello, K. Deb, “Problem Definitions and Evaluation Criteria for the CEC 2006 special Session on Constrained Real-Parameter Optimization”, IEEE Congress on Evolutionary Computation (CEC-2006), IEEE, Vancouver, BC, Canada, 2006

## 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 completed his Bachelor of Engineering (BE) in Computer Engineering from PICT, Pune, MS, India and Master of Engineering (ME) from Computer Science and Engineering (CSE) from Walchand College of Engineering, at Sangli, MS, India.

He has 12 years of teaching experience at UG and 6 years at PG. His research interests include Parallel Genetic Algorithms, Parallel Evolutionary Algorithms and Parallel programming. He has published 11 research papers in Conferences and Journals.

**Madhuri S. Joshi** is presently working as an Professor, Department of Computer Engineering, Jawaharlal Nehru Engineering College, Aurangabad, M.S., India. She completed her B.E. (Electronics & Telecom) from College of Engineering, Pune (1985), M.Tech(CS) from IIT Madras (1993) and Ph.D. in 2008. Her research interests are Image Processing, Pattern Recognition, Data Mining, Operating Systems. She has published about 23 research papers in Conferences and Journals.

**Pranali Dilip 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, M. S. Joshi, P. D. Sheth,"Dual Population Genetic Algorithm for Solving Constrained Optimization Problems", International Journal of Intelligent Systems and Applications (IJISA), vol.7, no.2, pp.34-40, 2015. DOI: 10.5815/ijisa.2015.02.05