# Multi-Objective Optimization using Binary-Real Multi Population Hybridization: Parallel Universe Alien Genetic Algorithm (PUALGA)

**Rupande N. Desai, Associate Professor, Rubber Technology Department, L D College of Engineering, Ahmedabad-380015. Gujarat, India. Email: rupandendesai@gmail.com**

**Dr. Narendra M. Patel, Associate Professor, Chemical Engineering Department, Vishwakarma Government Engineering College, Chandkheda, Ahmedabad-382424, Gujarat, India. Email: prof.nmpatel@gmail.com**

**Dr. S. A. Puranik, Director, Environmental Audit & Consultancy Cell, Atmiya Institute of Technology & Science [affiliated to Gujarat Technological University], Kalawad Road, Rajkot-360005, Gujarat, India. Email: sapuranik@aits.edu.in**

**Abstract** **Evolutionary Computation is becoming the most proven method for Global optimization of complex problems. Even with the developments in the computational powers of computers, solving the complex multi-objective problems requires very long time. There is always a need for development of robust and computationally efficient algorithms for engineering problems. Multi-Objective Optimization (MOO) is a class, which deals with multiple conflicting objectives simultaneously. MOO problems with conflicting objectives will have a set of solutions (representing trade-offs among the objectives), which are called pareto optimal solutions, of which none can be said to be better than the others with respect to all objectives. Usually the decision makers want a small set of solutions to make a choice among them. The challenge is to provide them with a set, as small as possible, that represents the whole set of choices, but to compute this set in an efficient way.**

**The efficiency of evolutionary algorithm can be increased by using subpopulations. This approach is used in this work and further extended by using two different encoding, real and binary for sub populations. Binary and real coded subpopulations exchange information through Aliens going from binary population to real population. This concept we present as Parallel Universe Alien Evolution; which we have implemented for Genetic Algorithm framework. This approach will improve robust ness of GA and maintain the diversity of population. The proposed algorithm is tested using nine benchmark multi-objective test problems and result show that the convergence and diversity of the final populations increase consistently.**

***Keywords — Alien GA, Evolutionary Algorithms, Genetic Algorithm, Engineering Optimization, Multi-Objective Optimization, Multi-Population GA.***

## I. INTRODUCTION

Efficiency in manufacturing and engineering activities is a result of optimization of design and operations. In particular for the efficiency of rubber processing, a scope still exists for optimizing the current industrial operations with the ever changing economic, energy and environmental landscape. Most engineering optimization problems are complex in nature and multi objective. Multi-objective optimization (MOO) is a class which deals with multiple and conflicting objectives simultaneously. When objectives are conflicting, achieving the optimum for one objective requires some compromise on one or more other objectives. Some examples of sets of conflicting objectives are: capital cost and operating cost, selectivity and conversion, quality and conversion, profit and environmental impact, and profit and safety cost. The relevance and importance of MOO in rubber technology is increasing due to increasing complexities in the design and operation of processes. The MOO implementation is being motivated by the availability of new and effective methods for solving multi-objective problems as well as increased computational capabilities.

 DOI : 10.18231/2454-9150.2018.0751 

MOO problems with conflicting objectives will have a set of solutions (representing trade-offs among the objectives), which are called Pareto optimal solutions (non-inferior solutions or effective solution), of which none can be said to be better than the others with respect to all objectives[1]. There are possibilities of multiple solutions in the Pareto front and all are equally important. Hence it is important to find as many Pareto-optimal solutions as possible in a problem. There are two main goals in a MOO, (a) to find a set of solutions as close as possible to the true optimal Pareto front and (b) to find a set of solutions as diverse as possible. First goal is common for any optimization problem whereas second goal is specific to multi-objective optimization problem.

Bhaskar et al. presented the background of MOO, different methods and their applications in chemical engineering until the year 2000.[2] He shows that there were around 30 journal publications covering different areas in chemical engineering on applications of MOO before year 2000. MOO applications in polymerization are included in the review of genetic algorithm applications in polymer science and engineering by Kasat et al. (2003) [3]. Applications of non-dominated sorting genetic algorithm (NSGA), NSGA-II and its jumping gene adaptations in chemical reaction engineering were reviewed by Nandasana et al. (2003) [4].

The first implementation of a real multi-objective evolutionary algorithm (vector-evaluated GA or VEGA) was suggested by David Schaffer in the year 1984 (Schaffer, 1984) [5] . Schaffer modified the simple genetic algorithm (with selection, crossover, and mutation) by performing independent selection cycles according to each objective. No significant study was performed for almost a decade after the pioneering work of Schaffer till a new non-dominated sorting procedure suggested by David Goldberg (1989) [6]. Goldberg proposed to use the concept of domination to assign more copies to non-dominated individuals in a population. Since diversity is another concern, he also suggested the use of a niching strategy among solutions of a non-dominated class. Getting this clue, researchers developed different versions of multi-objective evolutionary algorithms. Basically, these algorithms differ in the way fitness is assigned to each individual. Srinivas and Deb (1994) developed a non-dominated sorting GA (NSGA) [7], which is similar to the MOGA. NSGA differs from MOGA in two ways: fitness assignment and the way niching are performed.

Elitism was not considered in the early MOEAs. After the publication of the SPEA paper, most researchers in the field started to incorporate external populations in their MOEAs as their elitist mechanism. In 2001, a revised version of SPEA (called SPEA2) was introduced. SPEA2 has three main differences with respect to its predecessor (Zitzler et al., 2001 [8]): (1) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual, the number of individuals that dominate it and the number of individuals by which it is dominated; (2) it uses a nearest neighbour density estimation technique which guides the search more efficiently, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions.

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) was introduced as an upgrade of NSGA (Srinivas and Deb, 1994) [7], although it is easier to identify their differences than their similarities (Deb et al., 2002)[9]. In NSGA-II, for each solution one has to determine how many solutions dominate it and the set of solutions which it dominates. NSGA-II estimates the density of solutions surrounding a particular solution in the population by computing the average distance of two points on either side of this solution along each of the objectives of the problem. This value is the so-called crowding distance. During selection, NSGA-II uses a crowded-comparison operator which takes into consideration both the non-domination rank of an individual in the population and its crowding. NSGA-II does not implement an elitist mechanism based on an external archive. Instead, the elitist mechanism of NSGA-II consists of combining the best parents with the best off springs obtained. NSGA-II is computationally much more efficient than its predecessor, and its performance is so good that it has gained a lot of popularity in the last few years, becoming a benchmark against which other MOEAs are often compared.

Dipama et al 2010 uses a grid based multi-objective evolutionary algorithm for the optimization of power plants[10]. His algorithm uses new techniques to sustain convergence towards Pareto's front as well as population diversity as compared to other conventional algorithm. His method does not need ranking or crowding mechanisms to be invoked. Unlike classical evolutionary algorithms that promote non-dominated solutions at each generation, the present approach consists of emphasizing dominated and non-dominated ones to drive the searching process towards the boundaries of the feasible region.

We propose to use hybridization of binary coded and real coded GA in this work. Binary coded GA can explore search space reducing the accuracy of encoding. Two parallel populations are created and evolved exchanging information. Members from binary coded population go to real coded populations as Aliens and take part in evolution. This approach combines the strengths of binary coded and real coded GA along with benefits of parallel populations. We use non dominated sorting and crowding distance for selection. We discuss the proposed algorithm and show its working in next section. Followed to that we discuss about the performance measures used for MOO. Results are discussed in the next section followed by conclusion of the work.

 DOI : 10.18231/2454-9150.2018.0751 

## II. PROPOSED ALGORITHM

Most multi-objective optimization algorithms use the concept of dominance in their search. If a solution is better than the others with respect to all objectives, it is a dominating solution. For a decision vector, which is not dominated by any other decision vector, it is optimal if we cannot improve in any objective without causing degradation in at least one of the objective. Non-dominated solutions are ones within the search space whose corresponding objective vector components cannot be improved simultaneously. Such solutions are denoted as Pareto optimal, or sometimes called non-inferior solution. Many approaches exist and are suggested for finding the non-dominated set from a given population of solutions. Different approaches have different computational complexities. Few of them are Naive and slow, Jun Du Algorithm, Ding's Algorithm, Kung's Algorithm and Arena sort.

In addition to minimum computational efforts and performance for all types of problems, a good multi-objective optimization algorithm is expected to achieve the following conflicting goals:

- The best-known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto optimal set.
- Solutions in the best-known Pareto set should be uniformly distributed and diverse over of the Pareto front in order to provide the decision-maker a true picture of trade-offs.
- The best-known Pareto front should capture the whole spectrum of the Pareto front. This requires investigating solutions at the extreme ends of the objective function space.

The most popular and powerful algorithm NSGA-II attempts to achieve the goals by classifying the population members in non-dominated fronts and calculating crowding distance for selection of better ones within fronts. Deb, 2001 proposed algorithm to classify the chromosomes into fronts based on non-domination as follows:

1. Create new (empty) box, P', of size, $N_p$.
2. Transfer *i*th chromosome from P to P', starting with $i$=1.
3. Compare chromosome i with each member, say, *j*, already present in P', one at a time.
4. If *i* dominates (Deb, 2001) over *j* (i.e. *i* is superior to or better than *j* in terms of all objective functions), remove the *j*th chromosome from P' and put it back in its original location in P.
5. If *i* is dominated over by *j*, remove *i* from P' and put it back in its position in P.
6. If *i* and *j* are non-dominating (i.e. there is at least one objective function associated with *i* that is superior to/better than that of *j*), keep both *i* and *j* in P' (in sequence). Test for all *j* present in P'
7. Repeat for next chromosome (in the sequence, without going back) in P till all $N_p$ are tested. P' now contains a sub-box (of size ≤ Np) of non-dominated chromosomes (a subset of P), referred to as the first front or sub-box. Assign it a rank number, Irank, of 1
8. Create subsequent fronts in (lower) sub-boxes of P' using step 2 above (with the chromosomes remaining in P). Compare these members only with members present in the current sub-box, and not with those in earlier (better) sub-boxes.

Deb, 2001 proposed crowding distance calculation for selection of chromosomes with uniform distribution over pareto front. Evaluate the crowding distance for the ith chromosome in any front j, of P' using the following procedure:

a) Rearrange all chromosomes in front *j* in ascending order of the values of any one (say, the *q*th) of their several objective functions (fitness functions). This provides a sequence, and, thus, defines the nearest neighbours of any chromosome in front *j*.
b) Find the largest cuboid (rectangle for two fitness functions) enclosing chromosome *i* that just touches its nearest neighbours in the f-space.
c) Crowding distance; *Ii* ,*dist* =1/2 ×(sum of all sides of this cuboid)
d) Assign large values of *Ii*, *dist* to solutions at the boundaries (the convergence characteristics would be influenced by this choice)

NSGA-II algorithm proposed by Deb, 2001 assuming that we are minimizing all the objective functions is as given under.

1. Generate box, P, of $N_p$ parent chromosomes using a random-number code to generate the several binaries. These chromosomes are given a sequence (position) number as generated
2. Classify these chromosomes into fronts based on non-domination (Deb, 2001), as follows:
3. Spreading out: Evaluate the crowding distance, *Ii*, *dist*, for the *i*th chromosome in any front, *j*, of P' using the following procedure:
4. Make $N_p$ copies randomly (duplication permissible), of the better chromosomes from P' into a new box, P'' using:
   i. Select any pair, *i* and *j* , from P' (randomly, irrespective of fronts)
   ii. Identify the better of these two chromosomes. Chromosome *i* is better than chromosome *j* if:
   iii. *Ii*, *rank* ≠ *Ij*, *rank* : *Ii*, *rank* < *Ij*, *rank*
   iv. *Ii*, *rank* = *Ij*, *rank* : *Ii*, *dist* > *Ij*, *dist*
   v. Copy (without removing from P') the better of these two chromosomes in a new box, P"
   vi. Repeat till P" has $N_p$ members. Not all of P' need be in P". By this method, the better members of P' are copied into P" stochastically
5. Copy all of P" in a new box, D, of size Np Carry out crossover (using the stochastic remainder roulette-wheel selection procedure, Deb, 1995) and mutation (Deb, 1995) of chromosomes in D. This gives a box of Np daughter chromosomes.
6. Elitism: Copy all the Np best parents (P") and all the Np daughters with transposons (D) into box PD. Box PD has 2 Np chromosomes

 DOI : 10.18231/2454-9150.2018.0751 

i. Reclassify these 2 $N_p$ chromosomes into fronts (box PD') using only non-domination (as described in Step 2 above)
ii. Take the best $N_p$ from box PD' and put into box P'''

7. This completes one generation. Stop if appropriate criteria are met, e.g., the generation number>maximum number of generations (user specified).
8. Copy P''' into starting box, P. Go to Step 2 above.

B. Sankararao and S.K. Gupta, 2007 observed that for ZDT4 test function binary coded NSGA-II do not converge. They noted that " It may be mentioned here that though the binary coded NSGA-II fails to converge to the global optimal solution, for this test problem, the real coded NSGA-II does indeed, converge to the correct pareto solutions in 100,000 function evaluations." (Deb 2001). This observation initiated the thought of having two parallel populations evolving simultaneously, one binary coded and another, real coded; which may make the GA more robust.

Aimin Zhou et. al, 2011 highlights that competitive/co-operative co-evolution is one of approach to improve convergence and robustness for MOO. This supported our hypothesis of using two parallel populations. N Patel and N Padhiyar (2008) explored a new concept of Alien in their work, which we propose to use here to exchange information between the parallel populations.

We propose in this algorithm to use two subpopulations, one real coded and another, binary coded. (Parallel Universe) User specified best members from binary coded population known as Alien members will go to real coded population and take part in evolution. It will transfer the information from one sub-population to another. This approach increases robustness without any additional computational burden by combining the capacity of both, binary and real coded GAs. In fact, dividing the population in sub-population will reduce the calculations needed for sorting and selection and hence will increase the overall efficiency of the algorithm. The implementation of propose algorithm is as follows:

Step(0):Initialization of GA Parameters: Total Population Size (*nPopul*), Binary Population Fraction (*bFr*), Number of Generations (*nGen*), Number of Decision Variables (*nVar),* Maximum and Minimum Bounds on Decision Variables (*xMin*, *xMax*), Accuracy for Binary Encoding (*Acc*), Number of Aliens transferred every generation from binary population to real population (nAl)

Step(1):Generation of binary and real coded Population and fitness calculation.

Step(2):Selection for nPopul members for binary and (nPopul – nAl) members for real population. Add nAl members from binary to real population.

Step(3):Carry out Crossover and Mutation for each Population.

Step(4): Do Fitness calculation for each population.

Step(5):Do Elitism selection for each binary and real population.

Step(6):Alien member addition from binary to real coded population replacing the worst member in real coded population.

Step(7):Continuation of loop if maximum number of generations are not reached otherwise continue the loop; go to step 2.

As a part of our research study we propose to develop GA program with modifications in the existing algorithm to make it more robust and efficient. We propose to use two parallel populations using binary and real encoding, evolving separating and exchanging information through Aliens transporting across two universe. We name this concept as Parallel Universe Alien GA (PUALGA). We test the concept with bench mark test problems and compare its performance with existing well known algorithms.

## III. PERFORMANCE MEASURE FOR MOO

The aim of all multi-objective optimization algorithms is to find as many different solutions as possible in the Pareto optimal set. A multi-objective optimization algorithm has to perform two tasks, (i) to guide the search towards the global Pareto optimal region and (ii) to maintain the population diversity (in the objective space, in the parameters space or in both of them) in the current non-dominated front. The general performance criteria for the multi-objective optimization algorithms are:

- **Accuracy -** how close the generated non-dominated solutions are to the best known prediction.
- **Coverage -** how many different non-dominated solutions are generated and how well they are distributed.
- **Variance for every objective -** which is the maximum range of non-dominated front, covered by the generated solutions (fraction of the maximum range of the objective in the non-dominated region, covered by a non-dominated set).

The performance of the search algorithm is difficult to evaluate when, true Pareto optimal set is not known. Those results are generally presented using various performance measures for the search algorithms. Some tools for visual representations of non-dominated solutions are scatter-plot matrix, value path, bar chart, star coordinate and visual methods. Visual descriptions are now inadequate as the area of multi-objective optimization has become much popular and number of different algorithms and modifications are coming up. Performance metrics are important performance assessment measure, which also allow us to compare algorithms and to adjust their parameters for better results. Deb classifies them in three categories, metrics evaluating closeness to the Pareto optimal front, metrics evaluating diversity amongst non-dominated solutions and metrics evaluating closeness and diversity.[11]

 DOI : 10.18231/2454-9150.2018.0751 

### A. *Convergence to true pareto front*

The commonly used metrics for evaluating closeness to the pareto optimal front are error ratio, generational distance, maximum pareto optimal front error proposed by Veldhuizen 1999 and set convergence metric proposed by Zitzler 1999.[12] [13] Because of simplicity Zitzler 1999 have suggested generational distance matrix to evaluate closeness of solution found to the true solution and Deb 2000 and letter investigators have used this method. Generational distance is an average distance of the solutions fond by the algorithm to the true pareto front. For a set Q of N solutions from a known set of the pareto optimal set P*. Veldhuizen 1999 has defined average distance of Q from P*, the generational distance γ as:

$$\gamma = \frac{\left(\sum_{i=1}^{|Q|} d_i^{\,p}\right)^{1/p}}{|Q|}$$

where, $d_i = \min_{k=1}^{|Q|} \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f_m^{*(k)}\right)^2}$

and $f_m^{*(k)}$ is the m-th objective function value of the k-th member of $P^*$.

When there are large fluctuations in the distance values, GD doesn't represent the true distance. Variance of the matrix GD is also necessary in such cases. When objective function values are of different magnitude, they should be normalised before calculating distance measure. A large number of solutions uniformly distributed in the true pareto should be used to calculate γ matrix. The γ matrix measures the extent of convergence to a known set of pareto optimal solutions. Since, multi-objective algorithms would be tested on problems having a known set of Pareto-optimal set, the calculation of this metric is possible. But, realize that such a metric cannot be used for any arbitrary problem. Even when all solutions converge to the Pareto-optimal front, the above convergence metric does not have a value zero. The metric will be zero only when each obtained solution lies exactly on each of the chosen solutions. Although this metric alone can provide some information about the spread in obtained solutions, we need to define another metric to measure the spread in solutions obtained by an algorithm.

### B. *Matrix to measure distribution of solutions*

There exist many metrics to find diversity amongst the obtained non dominated solutions. Here the purpose is to represent span of true pareto front covered by the obtained solutions and its uniformity in the span covered. Few popular amongst them are spacing matrix, Chi-square like deviation measure matrix by Deb 1989, maximum spread matrix by Zitzler 1999 and spread matrix by Deb et al 2000 [13] [14] [15].

From the obtained set of non-dominated solutions, we first identify the extreme solutions in the objective space. We calculate dme , the distances between the extreme solutions and the boundary solutions of the obtained non-dominated solution set Q from the known end solutions of P*. The distance measure may be Euclidian distance, the sum of the absolute distance in the objective values or the crowing distance. The parameter di is the distance measure between the neighbouring solutions and d(bar) is the mean value of this distance measure. For a scenario with a large variance of the distances may have a numerator value greater than the denominator. The spread, (delta) is calculated as

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|} \left|d_i - \bar{d}\right|}{\sum_{m=1}^{M} d_m^e + |Q|\bar{d}}$$

The maximum value of the metric can be greater than one. But, a good distribution would make all distances di equal to d(bar) and would make dme = 0 (with existence of extreme solutions in the non-dominated set). Thus, for the most widely and uniformly spread-out set of non-dominated solutions, the numerator of A(delta) would be zero, making the metric to take a value zero. For any other distribution, the value of the metric would be greater than zero. Note that the above diversity metric can be used on any non-dominated set of solutions, including one which is not the Pareto-optimal set.

### C. *Matrix evaluating closeness and diversity*

There are some metrics which combinedly evaluates closeness and diversity. They are Hypervolume attainable surface based statistical metric weighted metric and non-dominated evaluation metric.

## IV. MOO TEST FUNCTIONS

One of the fundamental issues when designing an algorithm is to have a standard methodology to validate it. As part of this methodology, certain benchmark test functions are required. We used selected seven test functions to test the proposed algorithm. For evaluation of the proposed algorithm, we have used nine benchmark test functions in this work. The test problems are chosen from past studies in this area. The function SCH1 (from Schaffer's study), FON (from Fonseca-Fleming's study), POL (from Poloni's study), and KUR (from Kursawe's study) are used in this study. The number of variable and pareto front nature details of the selected MOO test functions are given in table 1. Table 2 describes the objective functions for selected MOO test problems. All the problems have two objective functions, which are to be minimized. Each test function presents certain difficulties for multi-objective optimization.

 DOI : 10.18231/2454-9150.2018.0751 

**Table 1 MOO Test Function Parameters**

| Problem | Number of variables, *n*, and the bounds | Pareto front nature and location |
|---|---|---|
| SCH1 | $n = 1$<br>$-1e^5 \le x \le 1e^5$ | convex<br>for x ∈[-2,2] |
| FON | $n = 3$<br>$-4 \le x_i \le 4$ | non-convex |
| POL | $n = 2$<br>$-\pi \le x_i \le \pi$ | non-convex, discontinuous |
| KUR | $n = 3$<br>$-5 \le \bar{x} \le 5$ | non-convex |

**Table 2 MOO Test Function Objectives**

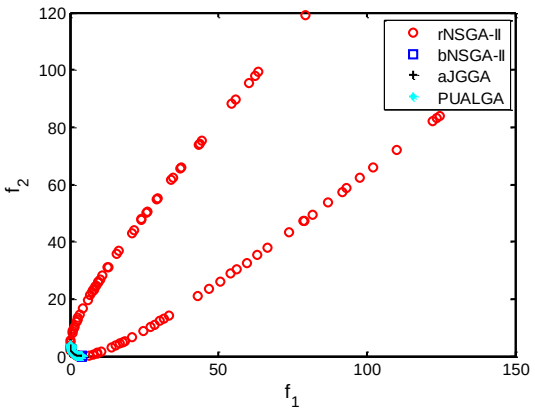
| Problem | Objective functions to be minimized |
|---|---|
| SCH1 | $f_1 = x^2$<br>$f_2 = (x-2)^2$ |
| FON | $f_1 = 1-\exp\left(-\sum_{i=1}^{i=3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$<br>$f_2 = 1-\exp\left(-\sum_{i=1}^{i=3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$ |
| POL | $f_1 = 1+(A_1-B_1)^2+(A_2-B_2)^2$<br>$f_2 = (x_1+3)^2+(x_2+1)^2$<br>$A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2$<br>$A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2$<br>$B_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2$<br>$B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2$ |
| KUR | $f_1 = \sum_{i=1}^{n-1}\left(-10\exp\left(-0.2\sqrt{x_i^2+x_{i+1}^2}\right)\right)$<br>$f_2 = \sum_{i=1}^{n}\left(\lvert x_i\rvert^{0.8} + 5\sin x_i^3\right)$ |

## V. RESULTS AND DISCUSSION

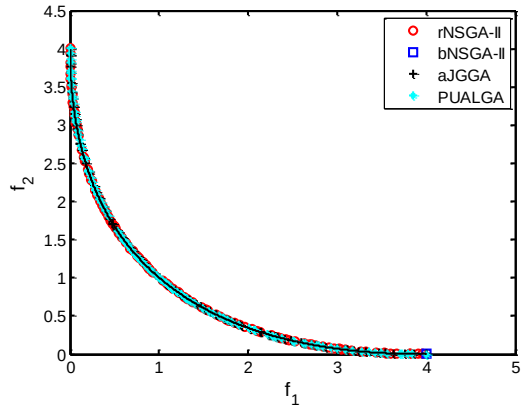
Though, the proposed algorithm can be used with any population based evolutionary optimization, we choose to use GA for its wide acceptance. Programs developed in MATLAB 2011 is used for all the cases in this work. It uses tournament selection, simulated binary crossover (SBX) and non-uniform mutation with elitism survival selection operators. The decision variables and their upper and lower limits for all the problems are given at table 1 along with the problem definitions. Population size is kept 100 for all runs. The number of generations used are 50 for SCH1, 70 for FON, 100 for POL, and 250 for KUR. Since GA is a stochastic optimization technique, it does not converge to the same solution every time even with the same initial population. Hence, we carried out ten simulation runs for every test problem with different initial population and average results are presented for the comparison of the algorithms.

### *A. Convergence to true Pareto front*

The metric value is a measure of average distance of the obtained solution from true front, hence smaller the value better is the convergence. Figure 1 to 4 represents pareto optimal solutions obtained with different algorithms along with true pareto front.

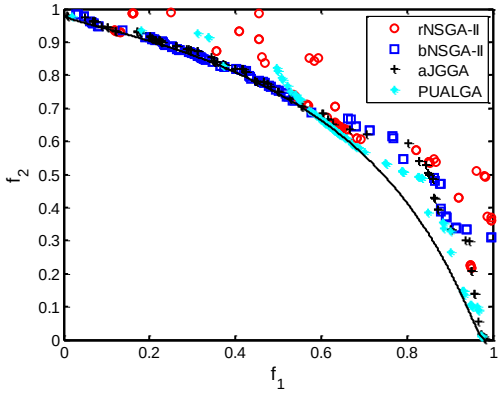



**(a) at 40 Generations**



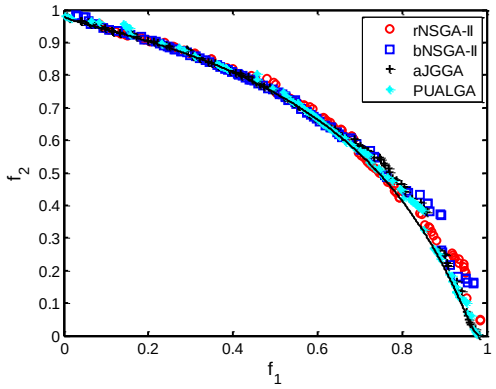

**(b) at 50 Generations**

**Figure 1 True pareto front and obtained optimal pareto solutions for SCH1 test function**



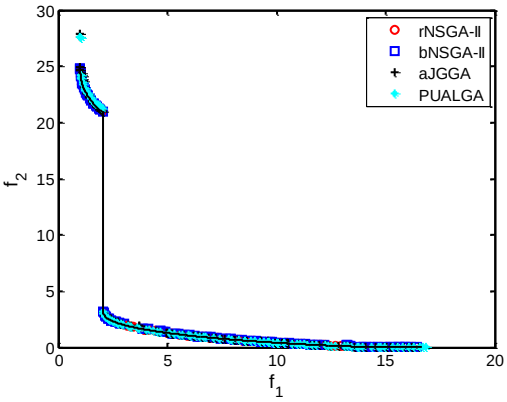

**(a) at 10 Generations**

 DOI : 10.18231/2454-9150.2018.0751 

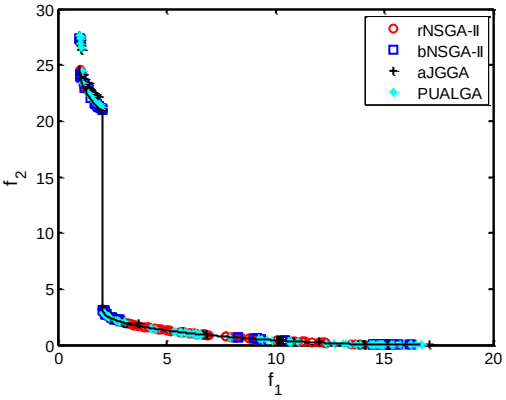



**(b) at 25 Generations**

**Figure 2 True pareto front and obtained optimal pareto solutions for FON test function**

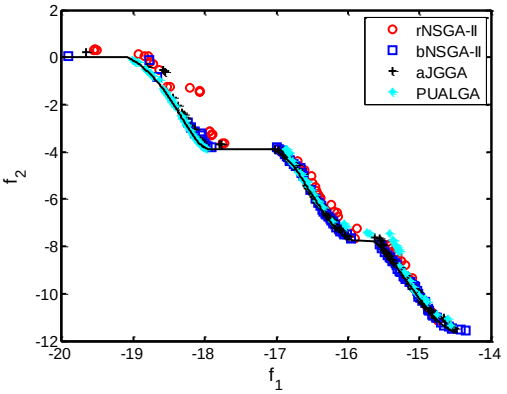



**(a) at 10 Generations**

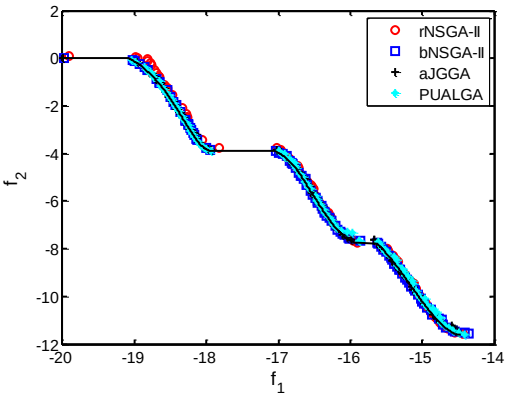



**(b) at 25 Generations**

**Figure 3 True pareto front and obtained optimal pareto solutions for POL test function**



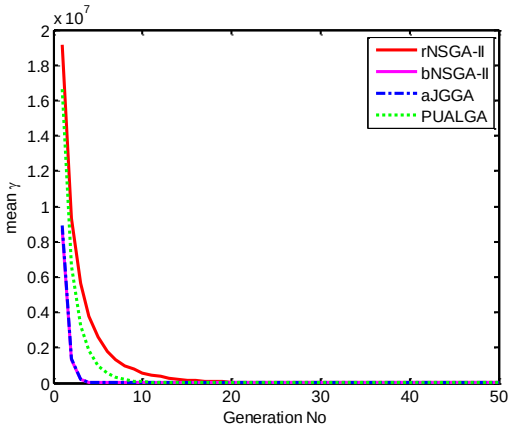

**(a) at 25 Generations**



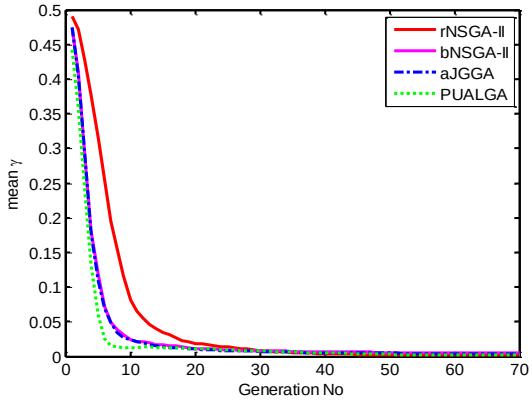

**(b) at 50 Generations**

**Figure 4 True pareto front and obtained optimal pareto solutions for KUR test function**

The generation wise convergence plot for all test functions are shown in figure 5. The diversity metric, Δ represents spread of solutions. It is a measure of distribution of solution along Pareto front. Zero value of the diversity metric indicates solutions are uniformly distributed covering full range of true front; smaller the value, better the spread. The generation wise progress of diversity metric, Δ is presented in figure 6.

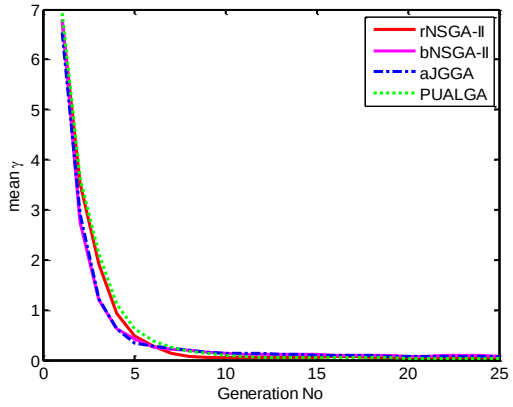



**(a) SCH1 *Function***

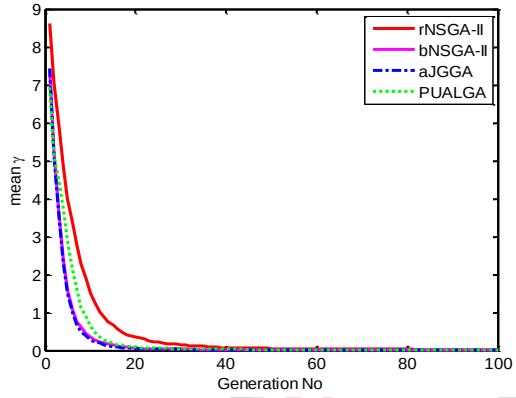



**(b) FON Function**

 DOI : 10.18231/2454-9150.2018.0751 

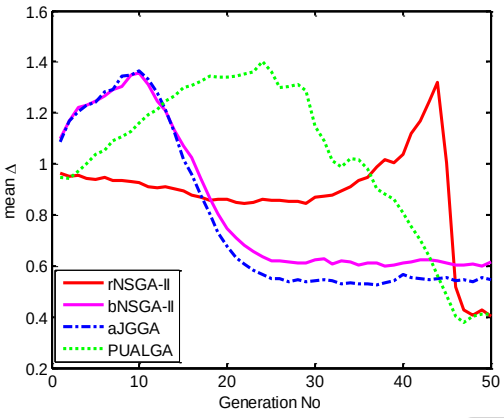



**(c) POL Function**

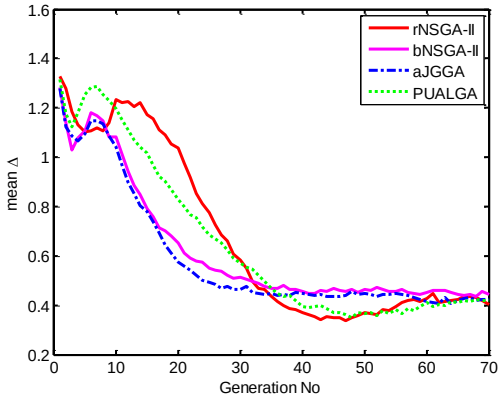



**(d) KUR *Function***

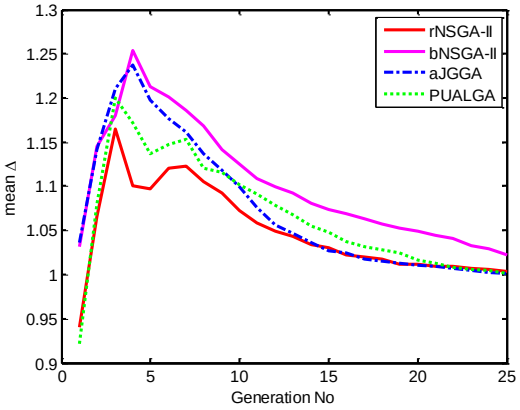
**Figure 5 Generation wise convergence for all test functions**



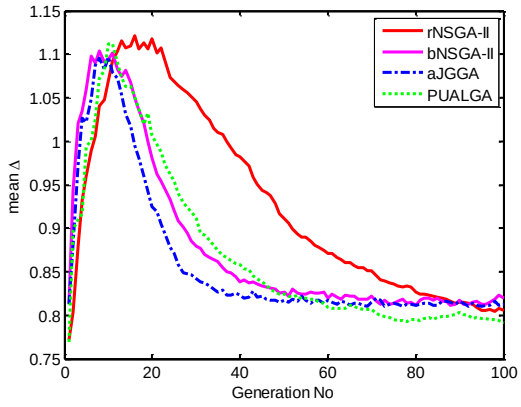

**(a) SCH1 Function**




**(a) FON Function**




**(c) POL Function**




**(d) KUR Function**

**Figure 6 Generation wise diversity for all test functions**

## VI. CONCLUSION

Proposed algorithm enhances the convergence rate and gets wide uniform spread of the Pareto solutions for multi objective optimization problems. This algorithm uses the concept of parallel population and combined binary and real GA. Non-dominated sorting is used in all algorithms for survival selection. The advantage of using two sub populations reduces the complexity of sorting and achieves better results with same computational efforts (Number of function Evaluations). Though, this concept can be applied for any population based MOGA, we show the results under GA framework in this study. The proposed PUALGA has been compared with widely accepted NSGA-II (binary and real coded) and Jumping Gene Adaptation of NSGA-II. Generation wise convergence rate by the proposed PUALGA has been observed to be very fast compared to the other approaches.

## REFERENCES

[1] R.E. Steuer, Multiple Criteria Optimization: Theory, Computation, and Application, Krieger Publishing Company, 1989.

[2] V. Bhaskar, S. Gupta K., A. Ray K., MODELING OF AN INDUSTRIAL WIPED FILM POLY(ETHYLENE TEREPHTHALATE) REACTOR, Polymer Reaction Engineering. 9 (2001) 71–99.

 DOI : 10.18231/2454-9150.2018.0751 

[3] R. Kasat B., A. Ray K., S. Gupta K., Applications of Genetic Algorithm in Polymer Science and Engineering, Materials and Manufacturing Processes. 18 (2003) 523–532.

[4] A. Nandasana D., A. Ray K., S. Gupta K., Dynamic Model of an Industrial Steam Reformer and Its Use for Multiobjective Optimization, Industrial & Engineering Chemistry Research. 42 (2003) 4028–4042.

[5] J.D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in: Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc., 1985: pp. 93–100.

[6] D. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: 1991: pp. 69–93.

[7] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (1994) 221–248.

[8] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation. 8 (2000) 173–195.

[9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, Evolutionary Computation, IEEE Transactions On. 6 (2002) 182–197.

[10] J. Dipama, A. Teyssedou, F. Aubé, L. Lizon-A-Lugrin, A grid based multi-objective evolutionary algorithm for the optimization of power plants, Applied Thermal Engineering. 30 (2010) 807–816.

[11] K. Deb, Multi-objective optimization using evolutionary algorithms, 1st ed, John Wiley & Sons, Chichester ; New York, 2001.

[12] D.A.V. Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, Air Force Institute of Technology, 1999.

[13] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications, 1999.

[14] K. Deb, Genetic Algorithms in Multimodal Function Optimization, 1989.

[15] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation. (2000).

 DOI : 10.18231/2454-9150.2018.0751 