Boundary Inspection Approach for Constrained Multi-Objective Optimization

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 **Constraint handling is always a critical part in performance of optimization method. There exists conventional approaches like, substitution, penalty function, slack variable, Lagrangian multiplier, ignorance infeasible for constraint handling. Some of them are convenient to use with evolutionary algorithms. These exists some hybrid methods, algorithm and/or problem specific approaches for constraint handling in optimization. There has been a lack of efficient and generic constraint handling techniques. We are proposing a new generalized boundary inspection approach based constraint handling mechanism for population based evolutionary algorithms (EAs). The concept is general and can be used with any population based EAs, we demonstrate its implemented for Multi-Objective Optimization (MOO) in this work. A comparative study of the proposed algorithm with the augmented penalty function method and ignorance infeasible are presented in this work. We use Parallel universe Alien Genetic Algorithm (PUALGA) with non-dominated sorting as basic MOO algorithms and evaluate our proposed constraint handling mechanism. We have considered three benchmark test problems for evaluating the proposed mechanism. Though the proposed constraint handling method is demonstrated for PUALGA, it is very generalized and can be used with any evolutionary algorithm easily. The method proposed converts all infeasible solutions in to feasible solutions maintaining diversity in search space.**

Keywords **—Evolutionary Optimization, Constrained Optimization, Multi-Objective Optimization, Genetic Algorithm, Alien GA, Parallel Universe, PUALGA**

I. INTRODUCTION

During the last two decades Evolutionary Algorithms (EAs) have proved to become an important tool for solving complex engineering optimization problems. Most real-world problems are however constrained and a possible criticism of the EAs has been the lack of efficient and generic constraint handling techniques. It should be noted that the evolutionary optimization algorithms are unconstrained by nature and hence need additional mechanisms to handle the constraints. Three excellent review articles on existing constraint handling methods for EAs are presented by Coello, 2002 [1], Kramer, 2010 [2] and Mezura-Montes and Coello, 2011 [3]. Some of the popular constraint handling approaches for the EAs are penalty method, preservation of feasible solutions method, augmented lagrangian method and feasibility based rule. In the penalty method, penalty parameter is multiplied with the extent of constraint violation and is augmented with the objective function. While it is the simplest method of handling constraints, finding the appropriate penalty values is a challenging task. Preservation of feasible solutions method does not distinguish the extent of constraint violation and requires large number of generation to converge. This may not necessarily increase the extra objective function evaluations, but it certainly requires computing constraint functions for the infeasible members. When, the constraint function is computationally expensive, this method becomes very slow in convergence. All these methods address the issue of guiding the solution candidates from infeasible to feasible region. Moreover, the constraint handling mechanisms were not explicitly intended for enhancing the convergence property.

 DOI : 10.18231/2454-9150.2018.0946

Constraint handling becomes even more crucial and complex in multi-objective EAs. Singh et al., 2010 [4] extended simulated annealing for multi-objective constrained optimization problems. Yang et al., 2014 [5] used constrained method and adaptive operator selection in Multi-objective evolutionary algorithm based on decomposition (MOEAD). Yang and Deb, 2013 [6] proposed a new cuckoo search for multi-objective optimization under complex nonlinear constraints. A study on the constrained multi-objective optimization has been presented by Qu and Suganthan, 2011 [7]. They have investigated three constraint handling methods along with their ensemble of constraint handling methods [8]. They ensemble self-adaptive penalty, superiority of feasible solution, and e-constraint methods. While the fitness values are calculated for both, the feasible and infeasible members in the self-adaptive penalty method, only feasible members are evaluated for their fitness values in the method of the superiority of feasible solution method. e-constraint method employs fitness assignment process similar to the superiority of feasible solutions method, but with an adaptive relaxation in constraint violation for initial few generations. We use augmented penalty function and ignore infeasible methods for comparison with the new proposed algorithm in this work.

We in this work present a generalized constraint handling approach for population based EAs using Boundary Inspection (BI) approach. The BI approach converts every infeasible member to a feasible one during the evolution process. The algorithm attempts to move infeasible point in a direction joining an infeasible point and a feasible point such that we reach within feasible area. At every generation using this approach all infeasible members are converted to feasible members by moving towards randomly selected feasible point. The parameter deciding the location of the new point is used from a predefined pool of values based on its success history.

The BI approach for constraint handling is discussed in the next section. The BI approach for constraint handling is tested with a multi-objective evolutionary algorithm : Parallel Universe Alien Genetic Algorithm (PUALGA) proposed by the authors [9]. The PUALGA algorithm with the BI approach for constraint handling is discussed in section III. Performance measures for MOO is discussed in section-IV followed by test problem summary in section V. The results are presented in section VI and concluding remarks are drawn in section VI.

II. Boundary Inspection Approach for Constraint Handling

A randomly created population is classified in two groups, namely feasible and infeasible ones. For every member from the infeasible group, one member from feasible group is selected randomly. The BI approach can be applied using half moves as demonstrated in the figure (1). Point *R* is the worst point selected from infeasible group and point *S* is the corresponding point selected from feasible group. Point N_1 is located moving *R* towards *S* in the direction joining *R* and *S*, half the distance between point *R* and *S.* The point N_1 is not feasible, hence further half distance move from N_1 is carried out, reaching to N_2. That point is also not feasible hence we move to point N_3 moving half distance towards *S*, which is a feasible point. We apply this procedure to all infeasible point and convert them to feasible point at every generation of evolution.

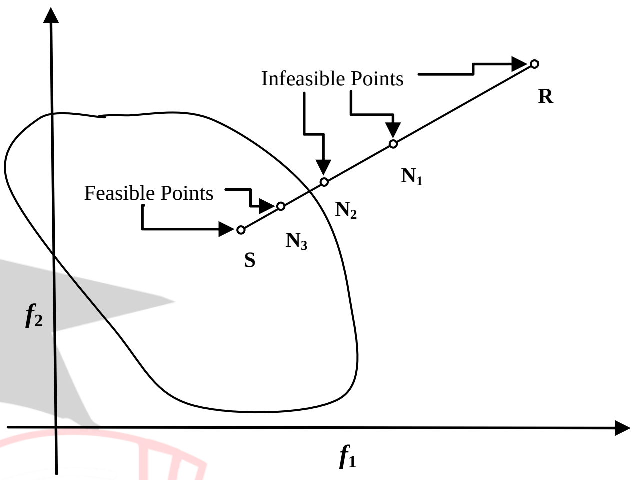


Figure 1: Boundary Inspection Approach by half moves

We propose to use an predefined ensemble of parameter λ to locate the new point on the line joining an infeasible point and the corresponding feasible point selected as shown in figure (2). Each value in the ensemble is given equal opportunity during initial learning period. The success count by each value in the learning period is converted to success probability, which is used in the next learning period. During the learning period the success probability is kept constant. Value of parameter λ to locate the new point is selected based on its success probability. Thus the value of parameter λ generating feasible point will automatically prefeed over the value failing. This will avoid the parameter tuning during evolution and problem specific tuning to the algorithm.

For each infeasible member *R*, one member, *S* from feasible population is selected randomly. A new point, *N* dividing the line joining point ***S*** and infeasible point, *R* in the λ:1 ratio is obtained such that it is feasible. The division ratio is selected from a predefined pool of λ values based of past performance history. An ensemble of possible values of ratio λ used are [-0.6, -0.3, 0.3, 0.6, 1, 1.5, 2].

 DOI : 10.18231/2454-9150.2018.0946

ISSN : 2454-9150 Vol-04, Issue-07, Oct 2018

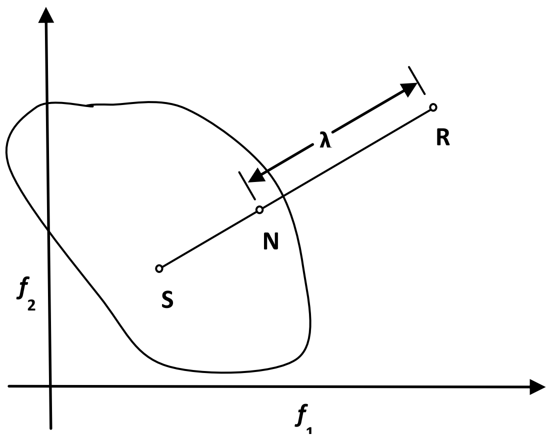


Figure 2: Boundary Inspection approach with predefined ensemble of parameter λ

III. PARALLEL UNIVERSE ALIEN GENETIC ALGORITHM (PUALGA) WITH BI APPROACH

We proposed the Hypothesis to use two subpopulations, one real coded and another, binary coded. (Parallel Universe) in Genetic Algorithm . We allow one or more best members from binary coded population known as Alien members to go to real coded population and take part in evolution. It will transfer the information from one sub-population to another. This approach provide robustness without any additional computational Burdon. 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 Hypothesis to use two subpopulations, one real coded and another, binary coded, Parallel Universe proposed as PUALGA improved convergence [8]. We now test the algorithm for constraint handling feature. We test commonly used constraint handling mechanisms; Feasibility rules (Ignore Infeasible) and Penalty functions [10] under PUALGA framework. We apply those concepts to both real coded populations. The overall PUALGA with BI approach for constraint handling is as follows:

Step(0): Initialization of GA Parameters

Step(1): Generation of binary and real coded population, fitness calculation and movement of infeasible points towards feasible members using BI approach.

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): Check constraint and move infeasible points towards feasible members using BI approach.

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.

IV. 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.[10]

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.[11] [12] 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

 DOI : 10.18231/2454-9150.2018.0946

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

and $f^{*(k)}_m$ 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 [12], maximum spread matrix by Zitzler 1999 [13] and spread matrix by Deb et al. 2000 [14].

From the obtained set of non-dominated solutions, we first identify the extreme solutions in the objective space. We calculate d_m^e , 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 $\bar{d}$ 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, Δ 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 $\bar{d}$ and would make $d_m^e = 0$ (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 Δ 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, non-dominated evaluation metric, and Inverted Generational Distance (IGD). IGD is a well known and widely accepted performance measure, which accounts convergence and distribution both [15]. Let P^* be a set of uniformly distributed true pareto optimal solutions and A is the obtained solution set, then IGD value is the average distance from P^* to A. Note that the smaller the IGD value, better is the performance of the MOO algorithm. We use IGD metric in this work for performance comparison of results obtained using different MOO algorithms.

V. TEST PROBLEMS

For testing the efficiency and effectiveness of the proposed BI approach for constraint handling with EAs, we use three two-objective constrained optimization test problems with known pareto optimal solutions. The three test problems are namely, Constr-Ex , BNH (Binh and Korn 1997) , OSY (Osyczka and Kundu 1995). All the problems have two objective functions, which are to be minimized. Each test function presents certain difficulties for constrained multi-objective optimisation. We use test problems with known sets of constrained Pareto-optimal solutions. The detailed discussion of the problem and its solution are available in Deb, 2001 [8]. For the convenience of the reader we briefly define the test problems here.

A. Test problem-1: Constr-Ex

A widely popular two variables and two objectives constrained optimization problem, namely Constr-Ex has the pareto optimal solution set in two regions, region A corresponding to $0.39 \le x_1 \le 0.67$ with $x_2 = 6 - 9\,x_1$ and the region B corresponding to $0.67 \le x_1 \le 1$ with $x_2 = 0$.

 DOI : 10.18231/2454-9150.2018.0946

ISSN : 2454-9150 Vol-04, Issue-07, Oct 2018

$$
\begin{aligned}
\text{min} \quad & f_1(\bar{x}) = x_1 \\
& f_2(\bar{x}) = \frac{1+x_2}{x_1} \\
\text{s.to} \quad & C_1(\bar{x}) = x_2 + 9x_1 \geq 6 \\
& C_2(\bar{x}) = -x_2 + 9x_1 \geq 1 \\
& 0.1 \leq x_1 \leq 5 \\
& 0 \leq x_2 \leq 5
\end{aligned}
$$

B. *Test problem-2: BNH*

We use another popular constrained optimization problem, namely BNH to validate the proposed concept. The pareto optimal set satisfying the constraints is available for this test problem in two regions, region A corresponding to $x_1 = x_2$ in [0-3] , while region B consisting of x_1 in [3-5] $x_2 = 3$.

$$
\begin{aligned}
\text{min} \quad & f_1(\bar{x}) = 4x_1^2 + 4x_2^2 \\
& f_2(\bar{x}) = (x_1-5)^2 + (x_2-5)^2 \\
\text{s.to} \quad & C_1(\bar{x}) = (x_1-5)^2 + x_2^2 \leq 25 \\
& C_2(\bar{x}) = (x_1-8)^2 + (x_2+3)^2 \geq 7.7 \\
& 0 \leq x_1 \leq 5 \\
& 0 \leq x_2 \leq 3
\end{aligned}
$$

C. *Test problem -3: OSY*

The OSY test problem has six variables and six constraints. This problem has only 3.25 % feasibility ratio as compared to 52.52 % for Constr-Ex and 93.61 % for BNH. The OSY problem definition is as follows,

$$
\begin{aligned}
\text{min} \quad & f_1(\bar{x}) = 25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 + (x_5-1)^2 \\
& f_2(\bar{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\
\text{s.to} \quad & C_1(\bar{x}) = x_1 + x_2 - 2 \geq 0 \\
& C_2(\bar{x}) = 6 - x_1 - x_2 \geq 0 \\
& C_3(\bar{x}) = 2 - x_2 + x_1 \geq 0 \\
& C_4(\bar{x}) = 2 - x_1 + 3x_2 \geq 0 \\
& C_5(\bar{x}) = 4 - (x_3-3)^2 - x_4 \geq 0 \\
& C_6(\bar{x}) = (x_5-3)^2 + x_6 - 4 \geq 0 \\
& 0 \leq x_1, x_2, x_6 \leq 10 \\
& 1 \leq x_3, x_5 \leq 5 \\
& 0 \leq x_4 \leq 6
\end{aligned}
$$

The pareto optimal solution set for this problem is available in five regions. Each of these five regions lies on one of the constraints' boundaries. The pareto optimal values for the variables x_1, x_2, x_3, and x_5 are summarized in table (1), while the optimum values of the remaining two variables are $x_4 = x_6 = 0$.

Table 1: Pareto optimal solutions for the OSY problem

Region	x_1	x_2	x_3	x_5
1	5	1	[1 - 5]	5
2	5	1	[1 - 5]	1
3	[4.056 - 5]	$(x_1 - 2)/3$	1	1
4	0	2	[1 - 3.732]	1
5	[0 - 1]	$2 - x_1$	1	1

VI. RESULTS AND DISCUSSION

PUALGA algorithm implemented in MATLAB using non-dominated sorting and elite survival selection operator for MOO is used to evaluate three constraint handling approaches. The population size is kept as 100 for all the test problems. We carried out twenty simulation runs for every test problem with distinct initial populations and a statistical analysis is presented for the comparison study of various algorithms.

Number of function evaluations (NFEs) and number of constraint evaluations (NCEs) are the two important measures for evaluating the computational expense of any constrained optimization algorithm. Performance metric IGD values are presented as the functions of NFEs and NCEs for the augmented penalty, ignore infeasible and boundary inspection to compare the computational performance.

The average of twenty runs in terms of IGD convergence profiles for the test problem Constr-Ex are presented in Fig. (3). The figure clearly indicates that the convergence of the proposed BCA constraint is better than the other two algorithms. The BI approach IGD values continues to decrease at a higher rate than both the other two algorithm, which indicates its better convergence capability. The BI approach converts infeasible members to feasible ones by projecting them through the feasible solutions. This mechanism creates possibilities of exploring guided search, which in turn improves the convergence.

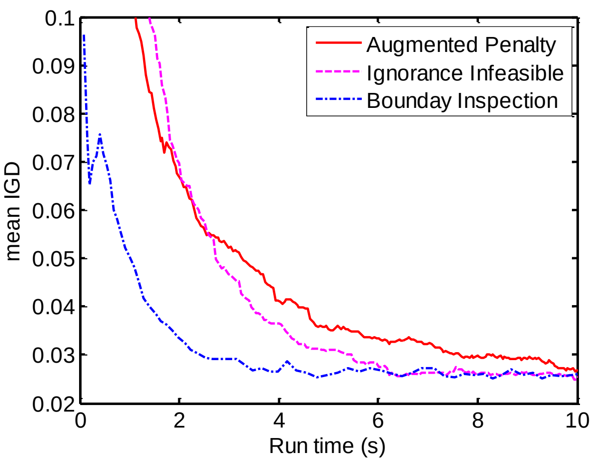


Figure 3: Average IGD values against Run time for ConstrEx test function

Since the BI implementation needs to evaluate constraints for all trial points, its convergence is also evaluated in terms of NCEs. The two IGD profiles, with respect to the NFEs and NCEs have similar trends among the three algorithms. The average IGD value convergence for Constr-Ex function is presented in terms of NFEs in Fig. (4) and NCEs in Fig. (5). The convergence profile became stagnant after 5,000 NFEs for BI approach and

 DOI : 10.18231/2454-9150.2018.0946

10,000 NFEs for ignorance infeasible .

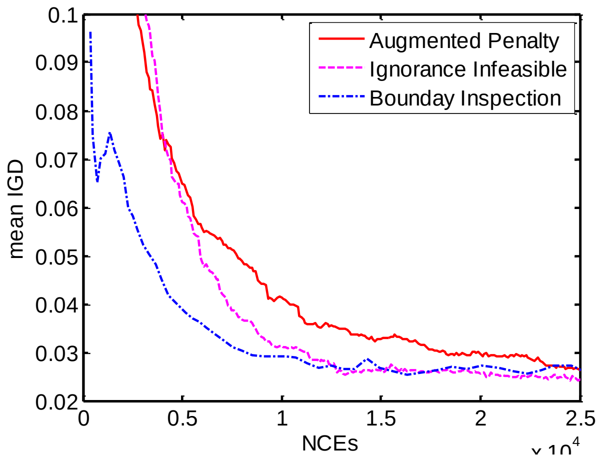


Figure 4: Average IGD values against NCE for ConstrEx test function

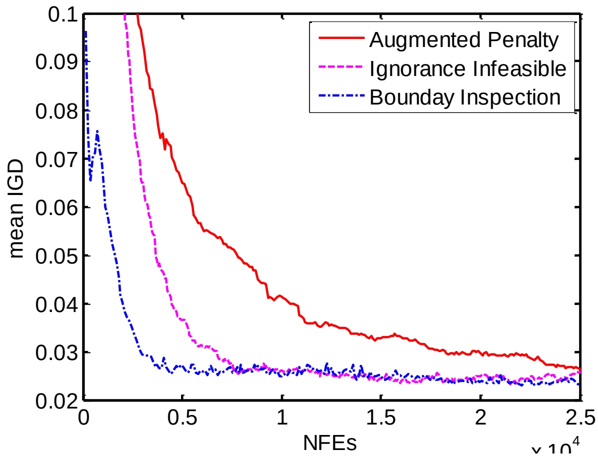
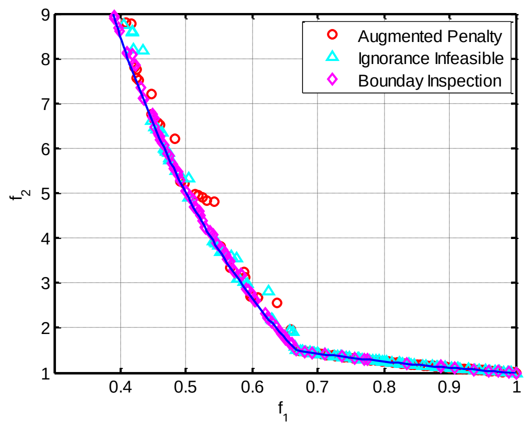
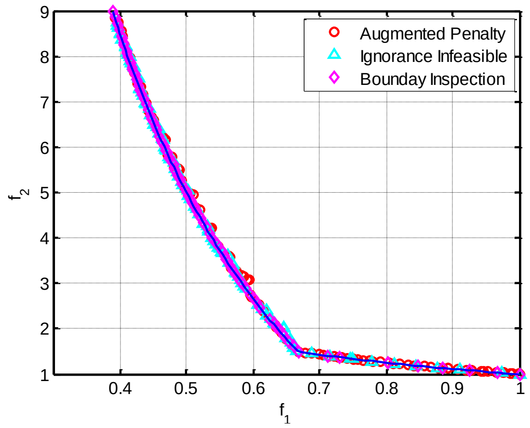


Figure 5: Average IGD values against NFE for ConstrEx test function

We represent pareto front obtained at the end of 25 generation and 100 generations for Constr-Ex test problem in Fig. (6). The pareto plot at 25 generations clearly indicate that the BI approach has uniform and better converged pareto.



(a) at 25 Generations



(b) at 100 Generations

Figure 6 : Pareto Front for ConstrEx test function

The convergence plots for the BNH test problem are shown in Fig. (7-9). The obtained results with this test problem are similar to the Constr-Ex problem. Since this test problem has high 93.61% feasibility ratio, the nature of convergence plots with respect to NFEs and NCEs are quite similar.

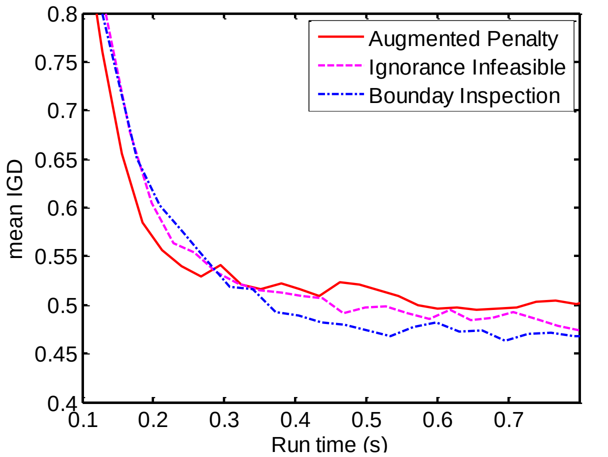


Figure 7: Average IGD values against Run time for BNH test function

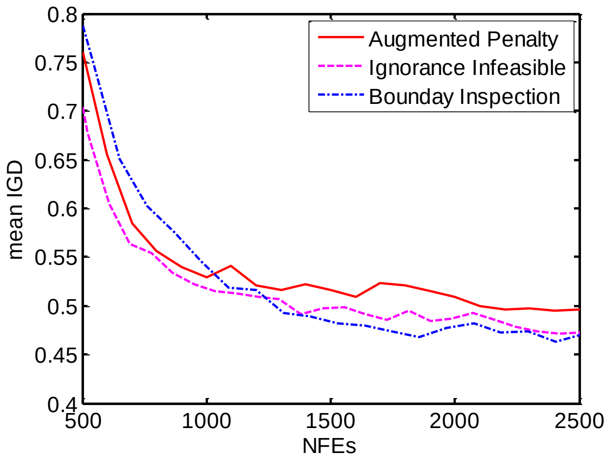


Figure 8: Average IGD values against NCE for BNH test function

 DOI : 10.18231/2454-9150.2018.0946

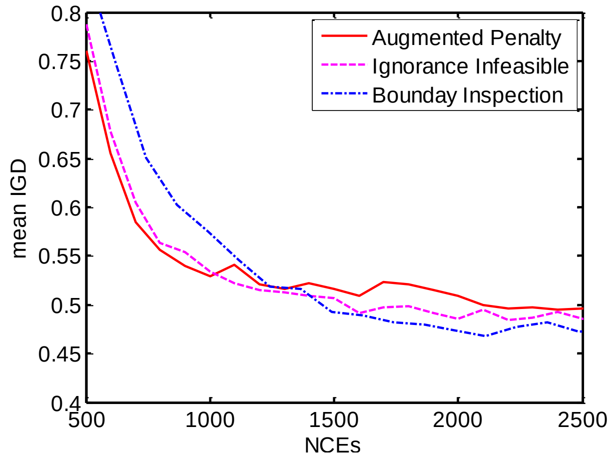


Figure 9: Average IGD values against NFE for BNH test function

We represent pareto front obtained at the end of 50 generation for BNH test problem in Fig. (10). Though all algorithm converge very close to true pareto front better uniformity of distribution of pareto optimal solutions is observed in BI approach.

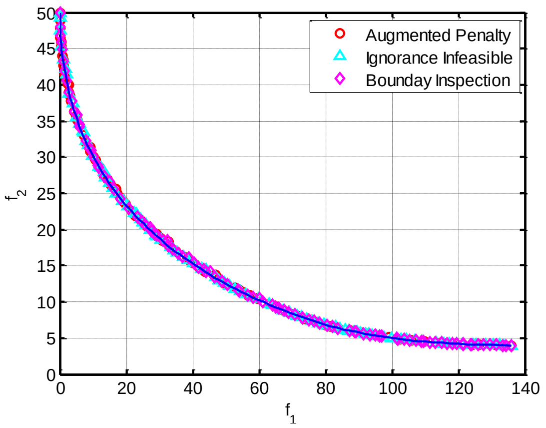


Figure 10 : Pareto Front for BNH test function 50 Generations

OSY test problem convergence plots are shown in Fig.(11-13). As this test problem has very low feasibility ratio of 3.25%, the nature of the convergence plots with respect to NFEs and NCEs are expected to be different. The other two algorithms show good performance in terms of NCEs compared BI approach due to low feasibility ratio.

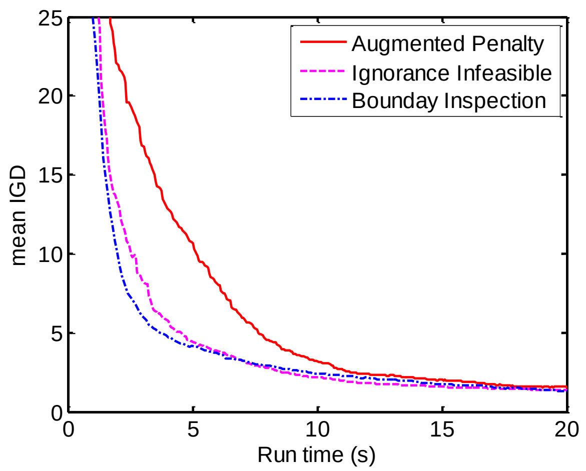


Figure 11: Average IGD values against Run time for OSY test function

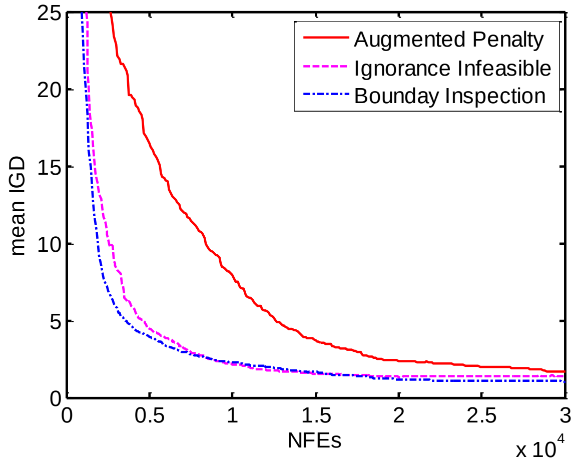


Figure 12: Average IGD values against NCE for OSY test function

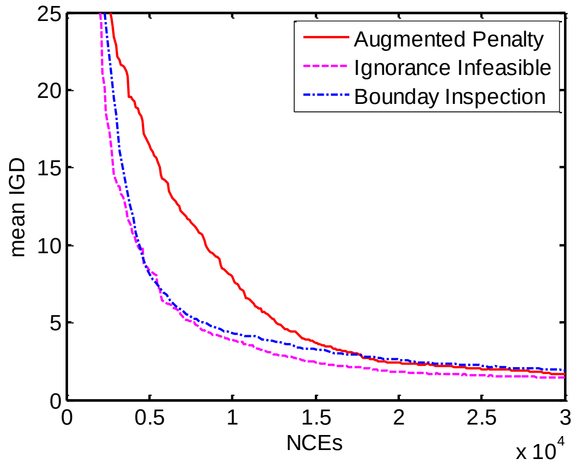
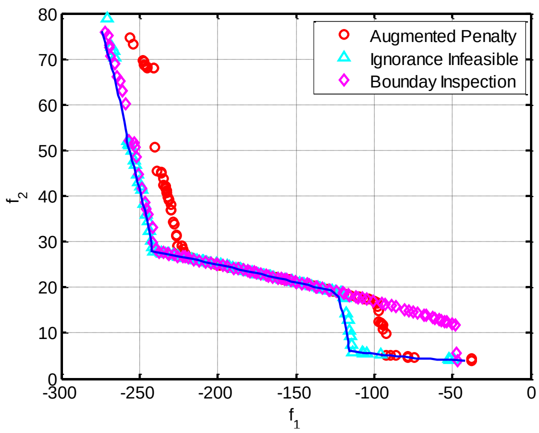


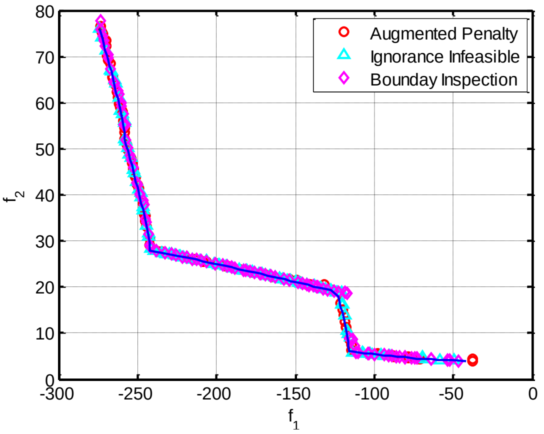
Figure 13: Average IGD values against NFE for OSY test function

We represent pareto front obtained at the end of 100 generation and 250 generations for OSY test problem in Fig. (14). Though all algorithm converge very close to true pareto front better uniformity of distribution of pareto optimal solutions is observed in BI approach. Augmented penalty approach obtained best coverage of pareto front covering both the end of the pareto front.



(b) at 100 Generations

 DOI : 10.18231/2454-9150.2018.0946



(b) at 250 Generations

Figure 14 Pareto Front for OSY test function

VII. Conclusion

Multi-objective constrained optimization problems are typically very difficult to solve. In this work, we have presented the constraint handling by BI approach in evolutionary algorithms. In the proposed algorithm, every infeasible member is projected through the randomly selected feasible member . This approach uses the original objective function values without any modification. The selection of parameter which locates the new point on the line joining infeasible and feasible point is based on success probability history, hence it is automated avoiding adaptive tuning during the evolution process. The efficacy of the BI approach is presented using multi-objective PUALGA algorithm and has been tested with three bench mark test functions. Statistical analysis of the performance measure, IGD is presented using 20 simulation runs for all the test problems. Further, the performance of the BI approach is compared with two popular constraint handling algorithms, namely augmented penalty function and ignore infeasible. Converge plot in terms of IGD are presented against run time, NFEs and NCEs to evaluate the comparative performance.

References

[1] Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer methods in applied mechanics and engineering, 191(11):1245–1287.

[2] Kramer, O. (2010). A review of constraint-handling techniques for evolution strategies. Appl. Comp. Intell. Soft Comput., 2010:3:1–3:19.

[3] Mezura-Montes, E. and Coello, C. A. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. Swarm and Evolutionary Computation, 1(4):173 - 194.

[4] Singh, H. K., Ray, T., and Smith, W. (2010). C-psa: Constrained pareto simulated annealing for constrained multi-objective optimization. Information Sciences, 180(13):2499 - 2513.

[5] Yang, Z., Cai, X., and Fan, Z. (2014). Epsilon constrained method for constrained multiobjective optimization problems: Some preliminary results. In Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion, GECCO Comp ’14, pages 1181 - 1186.

[6] Yang, X.-S. and Deb, S. (2013). Multiobjective cuckoo search for design optimization. Computers & Operations Research, 40(6):1616 - 1624.

[7] Qu, B. and Suganthan, P. (2011). Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods. Engineering Optimization, 43(4):403 - 416.

[8] Mallipeddi, R. and Suganthan, P. N. (2010). Ensemble of constraint handling techniques. IEEE Transactions on Evolutionary Computation, 14(4):561-579.

[9] Desai, R., Patel, N. and Puranik, S. A. (2018). Multi-Objective Optimization using Binary-Real Multi Population Hybridization: Parallel Universe Alien Genetic Algorithm (PUALGA). International Journal for Research in Engineering Application & Management (IJREAM), 04(6): 397 - 405.

[10] Deb, K. (2001). Multi-objective optimization using evolutionary algorithms, John Whiley & Sons Ltd., New York, 2001.

[11] D.A.V. Veldhuizen, Multi-objective evolutionary algorithms: classifications, analyses, and new innovations, Air Force Institute of Technology, 1999.

[12] E. Zitzler, Evolutionary algorithms for multi-objective optimization: Methods and applications, 1999.

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

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

[15] Zhao, S. Z., Suganthan, P. N., and Zhang, Q. (2012). Decomposition-based multi-objective evolutionary algorithm with an ensemble of neighborhood sizes. IEEE Transactions on Evolutionary Computation, 16(3) : 442–446.

DOI : 10.18231/2454-9150.2018.0946