NON-DOMINATED RANKED GENETIC ALGORITHM FOR SOLVING CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

[1]OMAR AL JADAAN, [2]LAKSHMI RAJAMANI, [3]C. R. RAO

[1]Dept. CSE, EC. Osmania University, Hyderabad 500-007, India.

[2]Prof. Dept. CSE, EC. Osmania University, Hyderabad 500-007, India.

[3]Prof. Dept. CSE, EC. Osmania University, Hyderabad 500-007, India.
E-mail: o_jadaan@yahoo.com, lakshmiraja@yahoo.com, crrcs@uohyd.ernet.in

ABSTRACT

Evolutionary algorithms are becoming increasingly valuable in solving large-scale, realistic engineering multiobjective optimization problems, which typically require consideration of conflicting and competing design issues. A criticism of Evolutionary Algorithms might be the lack of efficient and robust generic methods to handle constraints. The most widespread approach for constrained search problems is to use penalty methods, because of their simplicity and ease of implementation. Penalty function is generic and applicable to any type of constraint (linear or nonlinear). Nonetheless, the most difficult aspect of the penalty function approach is to find appropriate penalty parameters. In this paper, a method combining the new Non-dominated Ranked Genetic Algorithm (NRGA), with a parameterless penalty approach are exploited to devise the search to find Pareto optimal set of solutions, alleviate the above difficulties. The parameterless penalty approach that does not require any penalty parameter where penalty parameters assignment among feasible and infeasible solutions are made with a view to provide a search direction towards the feasible region. The new Parameterless Penalty and the Non-dominated Ranked Genetic Algorithm (PP-NRGA) continuously find better Pareto optimal set of solutions. This new algorithm have been evaluated by solving five test problems, reported in the multi-objective evolutionary algorithm (MOEA) literature. Performance comparisons based on quantitative metrics for accuracy, coverage, and spread are presented.

Keywords: *Multi-Objective Optimization, Pareto Optimal Solutions, Constrained Optimization, Penalty Functions, Ranking*.

1. INTRODUCTION

Trade-off information in the form of a Pareto optimal set of solutions is important in considering competing design objectives when making decisions associated with most engineering problems. The Presence of multiple objectives in engineering problems, in principle, gives rise to a Pareto set of optimal solutions, instead of a single optimal solution. In the absence of any further information, one of these Pareto optimal solutions cannot be said to be better than the other. This demands a user to find as many Pareto optimal solutions as possible.

Classical optimization methods (including the multi-criterion decision-making methods) suggest converting the multi-objective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto optimal solution at a time. When such a method is to be used for finding multiple solutions, it has to be applied many times, hopefully finding a different solution at each simulation run. A number of multi-objective evolutionary algorithms (MOEAs) have been suggested, mainly because of their ability to find multiple Pareto optimal solutions in one single simulation run. As evolutionary algorithms offer a relatively more flexible way to analyze and solve realistic engineering design problems, their use in multi-criterion decision making is becoming increasingly important. A number of multi-objective evolutionary algorithms (MOEAs) have been reported since the early eighties. Detailed summaries of the state-of-the-art in MOEA were discussed [1][2].

In the following, the general representation of a standard multi-objective problem and definitions of Pareto optimality are presented, followed by a brief overview of existing MOEAs. The details of the proposed parameterless penalty dominated ranking GA (PP-NRGA) approach PP-NRGA are then described, followed by its application to a five test problems. An extensive performance comparison of PP-NRGA and NSGA-II in solving the test problems is presented. Finally, concluding remarks are made with a brief discussion of PP-NRGA's strengths and weaknesses.

II. A STANDARD MULTI-OBJECTIVE OPTIMIZATION PROBLEM

A general multi-objective optimization problem consisting of k competing objectives and m constraints defined as functions of decision variable set x can be represented as follows:

$$Minimize\ \ f(x) = \{f_1(x), f_2(x), \ldots, f_k(x)\} \quad (1)$$

Subject to:

$$g_i(x) \le 0, \forall_i = \{1, 2, \ldots, m\} \quad (2)$$

$$x \in X \quad (3)$$

Where $x = \{x_j : j = 1, 2, \ldots n\}$ represents the decision vector, x_j is the j^{th} decision variable, X represent the decision space, $g_i(x), i \in \{1, \ldots, m\}$ are constraints, which include all equality constraints after transforming them to inequality constraints using (4)

$$|h(x)| - \varepsilon \le 0 \quad (4)$$

Where ε is a small tolerance. Since the algorithm that will be discussed does not use gradient information, it does not mater if equality constraint (4) is non-differentiable. F(x) is the multi-objective vector, and $f_l(x)$ is the l^{th} objective function.

III. PARETO OPTIMAL

Pareto optimal, which is also referred as non-dominance, of a set of solutions is formally defined as follows Cohon [3]:

A feasible solution to a multi-objective problem is Pareto optimal if there exist no other feasible solution that will yield an improvement in one objective without causing degradation in at least one other objective. Van Veldhuizen and Lamont [1] and Zitzler [29] provide a more rigorous definitions of this and related multi-objective terminology. Based on the definitions by Van Veldhuizen and Lamont and notations used in Equations (1-3), the following are defined:

Pareto Dominance: A multi-objective vector $u = (u_1, u_2, \ldots, u_k)$ is said to dominate $\upsilon = (\upsilon_1, \upsilon_2, \ldots, \upsilon_k)$ (denoted by $u \succ \upsilon$) if and only if u is partially more than υ, i.e. $\forall_i = \{1, 2, \ldots, k\}, u_i \le \upsilon_i \wedge \exists i = \{1, 2, \ldots k\}, u_i < \upsilon_i$

Pareto Optimality: A solution $x \in X$ is said to be Pareto optimal with respect to X if and only if there exists no $x \in X$ for which $\upsilon = F(x)$ dominates $u = F(x)$.

Pareto Optimal Set: For a given multi-objective problem $F(x)$, the Pareto optimal set $P*$ is a set consisting of Pareto optimal solutions. $P*$ is a subset of all the possible solutions in X. Mathematically, $P*$ is defined as follows:

$$p* = \{x \in X \mid \neg \exists x* \in X : F(x*) \succ F(x)\} \quad (5)$$

Pareto Front: The Pareto front, $PF*$ is the set that contains the evaluated objective vectors of $P*$. Mathematically $PF*$ is defined as:

$$PF* = \{u = F(x) \mid x \in P*\} \quad (6)$$

IV. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAs)

Over the past two decades, a number of different EAs were suggested to solve multi-objective optimization problems. Of them, VEGA [25] MOGA-III [14], SPEA2 [13], NSGA-II [18], Srinivas and Deb's NSGA [27], and Horn NPGA [17], Fonseca and Fleming's [15], for detailed information about other MOEA algorithms readers are encouraged to refer to [1],[17].

The [15],[27] and [17] algorithms demonstrated the necessary additional operators for converting a simple EA to a MOEA. Two common features on all three operators were the following:

- Assigning fitness to population members based on non-dominated sorting.
- Preserving diversity among solutions of the same non-dominated front.

Although they have been shown to find multiple non-dominated solutions on many test problems, and a number of engineering design problems, researchers realized the need of introducing more useful operators (which have been found useful in single-objective EA's) to solve multi-objective optimization problems better. Particularly, the interest has been to introduce elitism to enhance the convergence properties of a MOEA. Reference [29] showed that elitism helps in achieving better convergence in MOEAs. Among the existing elitist MOEAs, Zitzler and Thiele's SPEA [30], [31], Knowles and Corne's PAES [5], MOMGAIII [6], PAES-II [4], NSGA-II [18] are well studied. Readers are encouraged to refer to the original studies. Since evolutionary algorithms (EAs) work with a population of solutions, a simple EA can be extended to maintain a diverse set of solutions. With an emphasis for moving toward the true Pareto optimal region, an EA can be used to find multiple Pareto optimal solutions in one single simulation run.

V. PARAMETERLESS PENALTY- NON-DOMINATED RANKING GENETIC ALGORITHM (PP-NRGA)

Over the years, the main criticisms of the multi-objective evolutionary algorithms (EAs) might be the lack of efficient and robust generic methods to handle constraints. The most widespread approach for constrained search problems is to use penalty methods, because of their simplicity and ease of implementation. The penalty function approach is generic and applicable to any type of constraint (linear or nonlinear). Nonetheless, the most difficult aspect of the penalty function approach is to find an appropriate penalty parameters needed to guide the search towards the constrained optimum.

In this paper, a multi-objective GA approach (Non-dominated Ranking Genetic Algorithm NRGA) and an adaptive penalty function using ranks as penalty parameters are exploited to devise the new approach to find feasible Pareto front solutions. The new approach is called Parameterless Penalty Non-dominated Ranking Genetic Algorithm PPNRGA.

The adaptive penalty parameters assignments among feasible and infeasible solutions are made with a view to provide a search direction towards the feasible region. Two tires Rank-based Roulette Wheel selection operator (RRWS) is used along with the new adaptive penalty allow NRGA to continuously find better feasible Pareto optimal solutions, gradually leading the search near the true Pareto optimum solutions. NRGA with this constraint handling approach have been tested on five benchmarks problems commonly used in the literature. In all cases, the proposed approach has been able to repeatedly find feasible Pareto optimal solutions closer to the true Pareto optimal solutions than NSGA-II other MOEA.

VI. PROPOSED CONSTRAINT HANDLING METHOD

A. Penalty Method

The introduction of the penalty term enables us to transform constrained optimization problem equations (1-3) into an unconstrained one, such as the one given by (7):

$$Minimize\ \left\{\Psi(y) = \psi_1(y), \psi_2(y), \ldots, \psi_k(y)\right\} \quad (7)$$

Where $\psi_i = f_i(x) + r_g\,\phi\big(g_j(x)\big)$,

$j = 1,\ldots,m; i = 1,\ldots.,k$ and $\phi \ge 0$ is a real-valued function which imposes a penalty controlled by a sequence of *penalty coefficients*.

This transformation (i.e. (7)) has been used widely in evolutionary constrained optimization [19], [26]. The penalty function method may work quite well for some problems; however, deciding an optimal (or near optimal) value for r_g turns out to be a difficult optimization problem itself! If r_g is too small, an infeasible solution may not be penalized enough. Hence, an infeasible solution may be evolved by an evolutionary algorithm. If r_g is too large, a feasible solution is very likely to be found, but could be of very poor quality. A large r_g discourages the exploration of infeasible regions, even in the early stages of evolution. This is particularly inefficient for problems where feasible regions in the whole search space are disjointed. In this case, it may be difficult for an evolutionary algorithm to move from one feasible region to another unless they are very close to each other. Reasonable exploration of infeasible regions may act as bridges connecting two or more different feasible regions. The critical issue here is how much exploration of infeasible regions (i.e., how large r_g is) should be considered as reasonable. The answer to this question is problem dependent. Even for the same problem, different stages of evolutionary search may require different r_g values.

According to (7), different r_g values define different fitness functions. A fit individual under one fitness function may not be fit under a different fitness function. Finding a near optimal r_g adaptively is equivalent to ranking individuals adaptively in a population. Hence, the issue becomes how to rank individuals according to their objective and penalty values. A novel method for ranking individuals without specifying a r_g value is proposed. Experimental studies test the

effectiveness and efficiency of this method which can be regarded as an adaptive penalty approach.
It has been widely recognized that neither under-penalization nor over-penalization is a good constraint handling technique, and there should be a balance between preserving feasible individuals and rejecting infeasible ones [16]. In other words, ranking should be dominated by a combination of objective and penalty functions, and so the penalty coefficient r_g should be within the bounds $r_g^l < r_g < r_g^u$. It is worth noting that the two bounds are not fixed. They are problem dependent, and may change from generation to generation as they are also determined by the current population.

VII. Implementation of the Evolutionary Algorithm for Constrained Optimization

This section focuses on the detailed implementation of the evolutionary algorithm PP-NRGA for constrained optimization; also, the outcomes of testing the new approach on five benchmark problems are highlighted. The results are also compared with NSGA-II other evolutionary algorithm.

A. Ranking

To overcome the difficulty of determining the optimal r_g a different approach is suggested in this section to balance between the objective and penalty functions. The following fitness function is introduced:

$$\psi_i(x) = f_i(x) + rank_{f_i} + rank_g * \sum_{j=1}^{m} \phi(x) \quad (8)$$

Where $i = 1,2,...,k$, $rank_{f_i}$ is the rank of the objective function values, which takes values in the range of [1 − *population size*]. $rank_g$ Is the rank of the sum of the constraints violation for each solution, which takes values from [(*population size*+1)−(2 * *population size*)]. What (8) above mounts to is that minimum fitness value and less constraints violation inevitably leads to best fitness value. By using rank-based roulette wheel selection [21], [24], self-adapting is achieved without any extra computational cost. More importantly, the motivation of ranking comes from the need for balancing objective and penalty functions directly and implicitly in optimization. Equation (8) provides a convenient way of balancing in a ranked set.

B. Sorting Algorithm

In this study the fast non-dominated sorting approach from [18] is used because of the comparison with NSGA-II, and because it requires only $O\left(MN^2\right)$ computations. Whereas any fast sorting algorithm can be used.

C. Diversity Mechanism

Along with convergence to the Pareto optimal set, it is desired that an EA maintains a good spread of solutions in the obtained set of solutions. In the proposed PP-NRGA, crowding distance is used, which does not require any user defined parameter for maintaining diversity among population members. Readers are encouraged to refer to [18] for more information about the crowding distance.

D. Ranked Based Roulette Wheel Selection

The authors of [22], and [23] use modified roulette wheel selection algorithm where each individual is assigned a fitness value equal to its rank in the population; the highest rank has the highest probability to be selected (in case of maximization). The probability is calculated as illustrated in the following equation:

$$P_i = \frac{2 * Rank}{N * (N+1)} \quad (9)$$

Where N is the number of individuals in the front when dealing with individuals, and number of fronts when dealing with fronts. In this study the individuals in a front are ranked based on their crowding distance, and the fronts ranked based on the non-dominance rank.

E. Main Loop

Initially, a random parent population P is created, objectives and constraints values are evaluated, ranks of each objective are calculated, and the ranks of the sum of the constraints violation are computed. The constrained problem is converted to unconstrained one using the equation (8) for each objective function, then the population is sorted based on the non-domination. Each solution is assigned a fitness (or rank) equal to its non-domination level (1 is the best level, 2 is the next-best level, and so on). Thus, minimization of fitness is assumed. The crowding distance is computed for each solution in each front. At first, the usual Ranked based Roulette Wheel Selection (RRWS), recombination, and mutation operators are used to create offspring population Q of size N. Since elitism is introduced by comparing current population with previously found best non-dominated solutions, the procedure is different after the initial generation. We first describe the l^{th} generation of the proposed algorithm as shown in algorithm 1.

The step-by-step procedure shows that PP-NRGA algorithm is simple and straight forward. First, a combined population $P \cup Q$ is formed of size 2*N*. Then, objectives and constraints values are evaluated, ranks of each objective are calculated, and the ranks of the sum of the constraints violation are computed. After that the constrained problem is converted to unconstrained one using the equation (8) for each objective function, the combined population is sorted according to non-domination, and the crowding distance is computed. Since all previous and current population members are included in the combined population elitism is ensured. Now, reduction (elitism) procedure will select *N* solutions out of 2*N*, it will select the solutions with minimum non-domination rank first, the remaining solutions which belongs to same non-domination rank will be selected based on the crowding distance values. The new population of size *N* is now used for selection (RRWS), crossover, and mutation to create a new population *Q* of size *N*.

The two tiers ranked based roulette wheel selection [22], and [23] is used, The first tier will select the front based on the non-domination rank, here the solutions belonging to the best non-dominated set F_1 (best front) have the largest chance to be selected in the combined population. Thus, solutions from the set F_2 are chosen with less chance than solutions from the set F_1 and so on. Then second tier will select a solution inside the front based on the crowding distance value, the solution with larger crowding distance value will have larger chance to be selected.

The overall complexity of the algorithm is $O\left(MN^2\right)$, which is governed by the non-dominated sorting part of the algorithm. If performed carefully, the complete population of size 2*N* need not be sorted according to non-domination. As soon as the sorting procedure has found enough number of fronts to have members in P_{t+1}, there is no reason to continue with the sorting procedure.

TABLE I
PARAMETERS USED IN THIS STUDY

Param./Prob.	CTP1	CTP2	CTP3	CTP4	CTP5
No. Obj.	2	2	2	2	2
No. Var.	2	2	2	2	2
No. Const.	2	1	1	1	1
Pop. Size	100	100	50	200	200
Max. Gen.	200	200	200	800	400
Var. Bound	[0 1]	[0 1]	[0 1]	[0 1]	[0 1]

The diversity among non-dominated solutions is introduced by using the crowding distance procedure, which is used by the ranked based roulette wheel selection during the population selection phase. Since solutions compete with their crowding-distance (a measure of density of solutions in the neighborhood), no extra niching parameter (such as σ_{share} needed in the NSGA [27]) is required. Although the crowding distance is calculated in the objective function space, it can also be implemented in the parameter space, if so desired [9]. However, in all simulations performed in this study, the objective function space niching is used.

F. *Survival Selection (elitism):*

After evaluating the offspring's fitness (non-dominated rank, crowding distance), parents and offspring fight for survival as Pareto dominance is applied to the combined population of parents and offspring. Then the least dominated *N* solution vectors survive to make the population of the next generation.

VIII. TESTING AND EVALUATION OF PP-NRGA

In this section, first the test problems used to compare the performance of PP-NRGA with NSGA-II are described. For both NRGA and NSGA-II, the same parameter values have been chosen and have not made any effort in finding the best parameter setting. We leave this task for a future study.

A. Test Problems

Test problems are chosen from the literature. In 2001, Deb and his students [12] have developed test problems for constrained multi-objective optimization and suggested eight test problems. Five of those eight problems are chosen here and call them CTP1, CTP2, CTP3, CTP4, and CTP5. All problems have two objective functions and constraints. These test problems are described in the appendix. The simulated binary crossover (SBX) operator and polynomial mutation [10] are used. The crossover probability of $P_c = 0.9$ and a mutation probability of $P_m = 0.1$ are used. Distribution indexes [10] for crossover and mutation operators as $\eta_c = 15$ and $\eta_m = 20$, respectively are specified. The table I shows the remaining parameters values used in this study.

B. Performance Measures

Unlike in single-objective optimization, there are two goals in a multi-objective optimization:
1) Convergence to the Pareto optimal set and
2) Maintenance of diversity in solutions of the Pareto optimal set.

These two tasks cannot be measured adequately with one performance metric. Many performance metrics have been suggested [1], [7], [31], [28]. Here, two running performance metrics to understand the behavior of the algorithm from [11] are used in evaluating each of the above two goals in a solution set obtained by a multi-objective optimization algorithm. The first metric 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 solutions, the calculation of this metric is possible. First, we find a set of H = 600 uniformly spaced solutions from the true Pareto optimal front in the objective space. For each solution obtained with an algorithm, the minimum normalized Euclidean distance of it from H chosen solutions on the Pareto optimal front is computed. The average of these distances is used as the first metric CP^t (the convergence metric). In order to keep the convergence metric within [0, 1] once the metric values are calculated for all generations, normalize $C(P^t)$ by it maximum value (usually $C(P^0)$). When all obtained solutions lie exactly on chosen solutions, this metric takes a value of zero. Even when all solutions converge to the Pareto optimal front, the above convergence metric does not have a value of zero. The metric will yield 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, different metric to measure the spread in solutions obtained by an algorithm is defined (diversity metric). The second metric measures the extent of spread achieved among the obtained solutions. Here, we are interested in getting a set of solutions that spans the entire Pareto optimal region. For each objective, we calculate a diversity value as follows: the obtained non-dominated points at each generation are projected on suitable hyper-plane. The plane is divided into a number of small grids (or $M-1$ dimensional boxes). Depending on whether each grid contains an obtained non-dominated point or not, a diversity metric is defined. If all grids are represented with at least one point, the best possible (with respect to the chosen number of grids) diversity measure is achieved. If some grids are not represented by a non-dominated point, the diversity is poor. The parameters required from the user are the direction cosine of the reference plane, the number of grids (G_i) in each $(M-1)$ dimension, and the target (or reference) set of points. Because of lake of space reader recommended to refer to the original study [11] For more details about the running metrics used. In the experiments the number of grids is equal to the population size and $f_2 = 0$ plane to project the points.

Algorithm 1 PP-NRGA

1: Initialize Population P
2: Generate random population − size N
3: Evaluate objectives values and constraints
4: Calculate the rank of objectives values, $R_{f_i}, (i = 1, 2, ..., k)$ for each solution, $R_{f_i} \in [1 - N]$
5: Calculate the rank of the sum of the constraints violation R_g for each solution $R_g \in [(N+1) - (2N)]$
6: Convert the constrained problem to unconstrained one using the equation (8) for each objective function, for each solution in P
7: Assign Rank (level) Based on Pareto dominance – sort
8: Calculate the crowding distance between members on each front
9: Generate offspring Population Q from P
10: {Ranked based Roulette Wheel Selection
11: Recombination and Mutation
12: Evaluate objectives values and constraints}
13: **for** $g = 1$ to G **do**
14: **for** each member of the combined population ($P \cup Q$)
15: Calculate the rank of objectives values, $R_{f_i}, (i = 1, 2, ..., k)$ for each solution in the combined population $P \cup Q, R_{f_i} \in [1 - 2N]$
16: Calculate the rank of the sum of the constraints violation R_g for each solution in the combined population $P \cup Q, R_g \in [(2N+1) - (4N)]$
17: Convert the constrained problem to unconstrained one using the equation (8) for each objective function, for each solution in P
18: Assign Rank (level) based on Pareto – sort
19: Calculate the crowding distance between members on each front
20: **end for**
21: (elitist) Select the members of the combined population based on least dominated N solution to make the population P of the next generation. Ties are resolved by taking the less crowding distance.
22: Calculate the rank of objectives values, $R_{f_i}, (i = 1, 2, ..., k)$ for each solution, $R_{f_i} \in [1 - N]$.
23: Calculate the rank of the sum of the constraints violation R_g for each solution $R_g \in [(N+1) - (2N)]$.
24: Convert the constrained problem to unconstrained one using the equation (8) for each objective function, for each solution in P.
25: Assign Rank (level) Based on Pareto dominance sort.
26: Calculate the crowding distance between members on each front.
27: Q = Create next generation from P { .
28: | { Ranked based Roulette Wheel Selection.
29: Recombination and Mutation.
30: Evaluate objective values and constraints } | .
31: **end for**

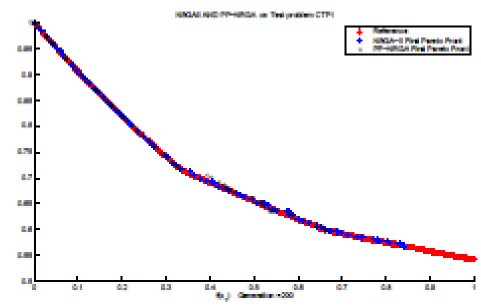

Fig. 1. CTP1 Final Generation

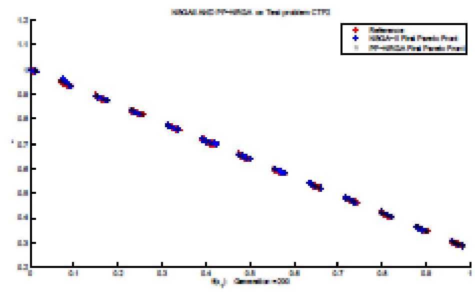

Fig. 2. CTP2 Final Generation

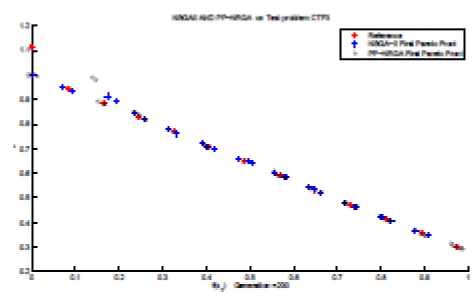

Fig. 3. CTP3 Final Generation

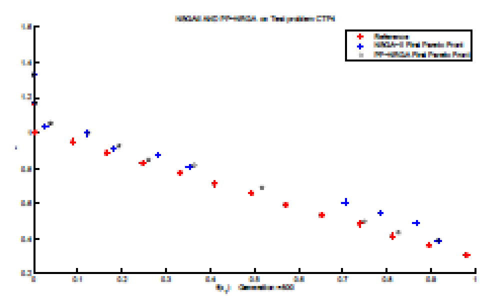

Fig. 4. CTP4 Final Generation

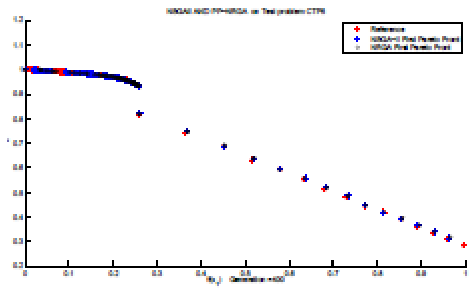

Fig. 5. CTP5 Final Generation

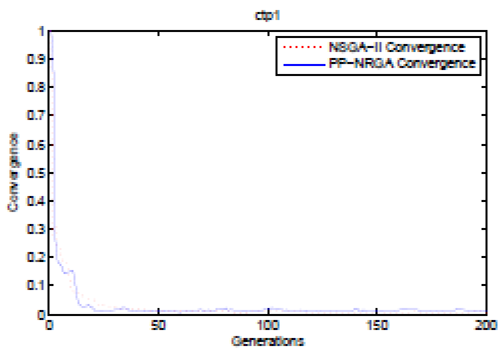

Fig. 6. CTP1 Convergence

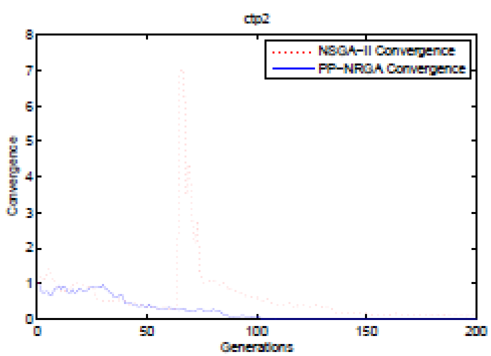

Fig. 7. CTP2 Convergence

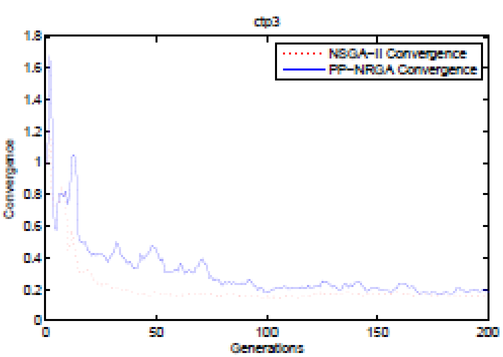

Fig. 8. CTP3 Convergence

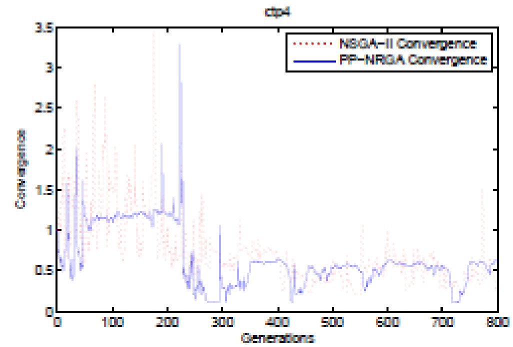

Fig. 9. CTP4 Convergence

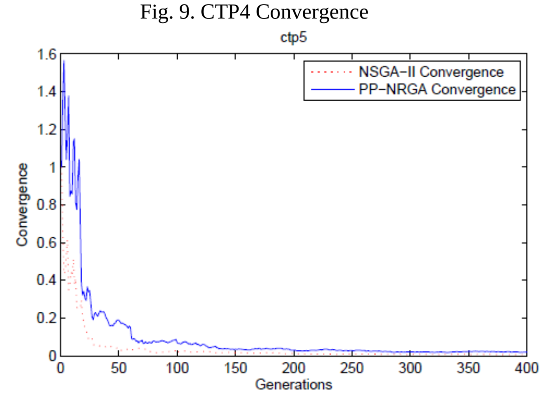

Fig. 10. CTP5 Convergence

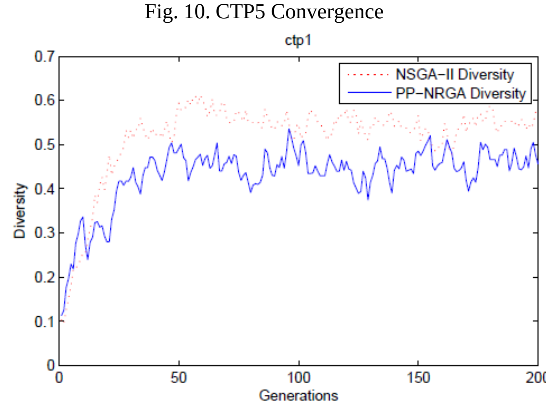

Fig. 11. CTP1 Convergence

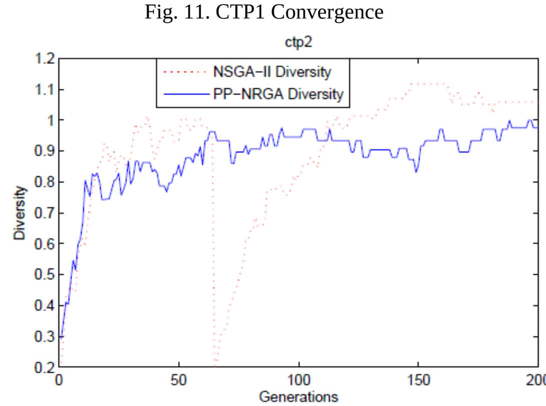

Fig. 12. CTP2 Convergence

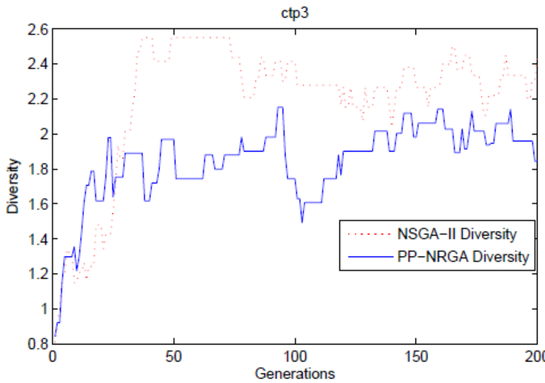

Fig. 13. CTP3 Convergence

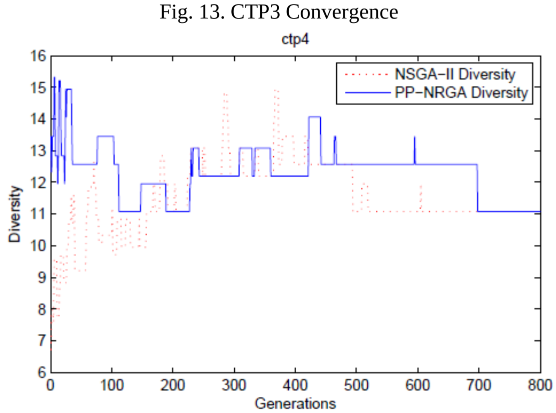

Fig. 14. CTP4 Convergence

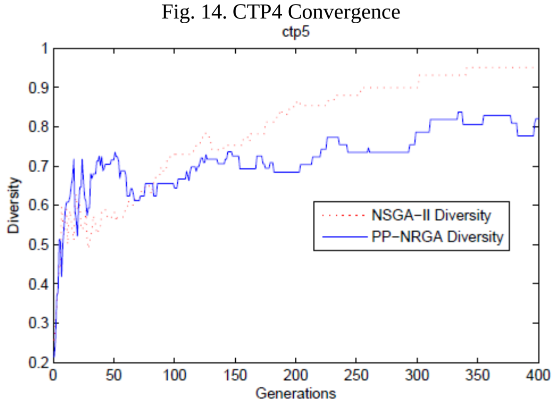

Fig. 15. CTP5 Convergence

C. Discussion of the Results

Figures 1-5 show the final generation but these will not give a clear picture about the behavior of the algorithms, for that the figures 6-10 illustrate the convergence behavior of PP-NRGA and NSGA-II algorithms. Figure 6 shows that the convergence metric of PP-NRGA on problem CTP1, quickly moves to zero faster than NSGA-II, thereby implying that starting from a random set of solutions PP-NRGA quickly approach the feasible Pareto optimal front faster than NSGA-II. A value of zero the convergence metric implies that all non-dominated solutions match the chosen Pareto optimal solutions. After about 20 generations, the PP-NRGA population comes very close to the Pareto optimal front, whereas NSGA-II took too much to get closer to the Pareto optimal front. The same is happened in the figures 7, and 9, where in CTP3 and CTP5 NSGA-II was ahead. Figures 11-15 show the diversity metric graphs obtained using the two algorithms. Figure 11 explains that the diversity metric increases exponentially till 50 generations in PP-NRGA and till 60 in NSGA-II after that the diversity remain more or less the same. Although the obtained solutions are very close to the chosen Pareto optimal front, the diversity metric oscillates near a stable value. The same behavior of the diversity metric in the remaining figures, except in figure 14 and figure 15 where PP-NRGA performs better than NSGA-II.

From above PP-NRGA out perform NSGA-II in most of the test problems. On most of the problems, PP-NRGA is able to find a better spread and faster convergence of solutions than NSGA-II algorithm. Fig. 1 shows all non-dominated solutions obtained after 200 generations with PP-NRGA and NSGA-II on CTP1 problem. The feasible Pareto optimal region is also shown in the figure. This figure demonstrates the abilities of PP-NRGA in converging to the feasible true front and in finding diverse solutions in the front. In the convergence aspect PP-NRGA performed better than NSGAII in this problem. But NSGA-II is shows better spread over the feasible Pareto front in the final GA population. Next, the non-dominated solutions on the problem CTP2 is shown in Fig. 2. This problem has a discontinued piece wise Pareto optimal front. the performance of PP-NRGA is better than NSGA-II. Although PP-NRGA get closer to the true front than NSGA-II, also PP-NRGA have found a better spread and more solutions in the entire Pareto optimal region than PP-NRGA. The problem CTP3 has vertexes Pareto optimal front. Figure 3 shows that the overall performance of PPNRGA is same NSGA-II, but the convergence and ability to find a diverse set of solutions are definitely better with PPNRGA. Finally, in figure 5 and figure 4 show that PP-NRGA finds a better converged set of non-dominated solutions in CTP4, CTP5 compared to NSGA-II algorithm.

IX. Conclusions and Future Work

This paper has proposed a new constraint multi-objective evolutionary algorithm called Parameterless Penalty Non-dominated Ranking Genetic Algorithm (PP-NRGA). The new elitist MOEA (PP-NRGA) have been tested on five different difficult test problems borrowed from the literature. The balance between the objective and penalty function is achieved through rank used in fitness function (8). The introduction of rank and using that rank in the fitness function enables the algorithm to bias toward the global Pareto front. The proposed method could get solutions closer to the global Pareto front on the five test problems. The PP-NRGA does not introduce any specialized variation operators, and does not require a priori knowledge about a problem since it uses parameterless penalty coefficient r_g in a penalty function. The procedure presented in this paper for handling constraints can be integrated to any evolutionary algorithm framework. The proposed PP-NRGA was able to maintain a better spread of solutions and converge better in the obtained feasible non-dominated front compared to NSGA-II. However, in all problems, PP-NRGA was able to converge closer to the true Pareto optimal front. PP-NRGA maintains diversity among solutions by controlling dynamic and parameterless crowding approach. However, the diversity preserving mechanism used in NSGA-II (which used in PPNRGA) is found to be the best with PP-NRGA.

With the properties of parameterless penalty function, two tiers ranked based roulette wheel selection procedure, a fast non-dominated sorting procedure, and an elitist strategy, PP-NRGA can be used for solving constraint real world problems, and should find increasing attention and applications in the near future.

REFERENCES

[1] Gary B. Lamont C. A. Coello and David A. Van Veldhuizen. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, second edition edition, 2007.

[2] C. A. C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information System, 1(3):269–308, 1999.

[3] J.L. Cohon. Multiobjective programming and planning. Mathematics in Science and Engineering, 140, 1978.

[4] David W. Corne, Nick R. Jerram, Joshua D. Knowles, and Martin J. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001, pages 283–290. Morgan Kaufmann Publishers, 2001.

[5] David W. Corne, Joshua D. Knowles, and Martin J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 839–848, Paris, France, 2000. Springer.

[6] R. O. Day. Explicit Building Block Multiobjective Evolutionary Computation: Methods and Application. PhD thesis, Air Force Institute of Technology, AFIT/ENG, BLDG 642, 2950 HOBSON WAY, WPAFB (Dayton) OH 45433-7765, USA, June 2005.

[7] K. Deb. Optimization for Engineering Design: Algorithms and Examples. Prentice-Hall, New Delhi, 1995.

[8] K. Deb. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons LTD, New York, NY, 2001.

[9] Kalyanmoy Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation, 7:205–230, 1999.

[10] Kalyanmoy Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. Complex syst., 9:115–148, April 1995.

[11] Kalyanmoy Deb and Sachin Jain. Running performance metrics for evolutionary multi-objective optimization. Technical Report 2002004, Kanpur Genetic Algorithm Labratory (KanGAL), 2002.

[12] Kalyanmoy Deb, Amrit Pratap, and T. Meyarivan. Constrained test problems for multi-objective evolutionary optimization. In First International Conference on Evolutionary Multi-Criterion Optimization EMO-2001, pages 284–298. Springer-Verlag, 2001.

[13] Marco Laumanns Eckart Zit. Ler and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001), pages 95–100, Athens, Greece, 2001.

[14] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, the Fifth International Conference on Genetic

Algorithms, pages 416–423, San Mateo, CA, 1993. Morgan Kauffman.

[15] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, the Fifth International Conference on Genetic Algorithms, pages 416–423. Morgan Kaufmann Publishers, 1993.

[16] M. Gen and R. Cheng. Genetic Algorithms and Engineering Design. Wiley, New York, 1997.

[17] N. Nafploitis J. Horn and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In Z. Michalewicz, editor, the First IEEE Conference on Evolutionary Computation, pages 82–87, Piscataway, NJ, 1994. IEEE Press.

[18] Sameer Agarwal Kalyanmoy Deb, Amrit Pratap and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans on Evolutinary Computation, 6(2):182–197, April 2002.

[19] S. Kazarlis and V. Petridis. Varying fitness functions in genetic algorithms: Studying the rate of increase in the dynamic penalty terms. In Parallel Problem Solving from Nature, pages 211–220, 1998.

[20] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary Computation, 8(2):149–172, 2000.

[21] C. R. Rao Omar Al Jadaan, Lakshmi Rajamani. Improved selection operator for ga. Journal of Theoretical and Applied Information Technology, 4(4):269277, 2008.

[22] Lakishmi Rajamani Omar Al Jadaan, C.R. Rao. Ranked based roulette wheel selection method. In International Symposium on Recent Advances in Mathematics and its Applications: (ISRAMA 2005), Calcutta Mathematical Society at AE-374, Sector-1, Salt Lake City Kolkata (Calcutta) 700064, India, 2005.

[23] Lakishmi Rajamani Omar Al Jadaan, C.R. Rao. Parametric study to enhance genetic algorithm performance,using ranked based roulette whell selection method. In International Conference on Multidisciplinary Information Sciences& Technologies (InSciT2006), volume 2, pages 274–278, Merida,Spain, October 2006.

[24] Lakshmi Rajamani Omar Al Jadaan, C. R. Rao. Parametric study to enhance genetic algorithm performance,using ranked based roulette whell selection method. In InSciT2006, volume 2, pages 274–278, Merida, Spain, 2006.

[25] J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. PhD thesis, Vanderbilt University, Nashville, Tennessee, 1984.

[26] W. Siedlecki and J. Sklansky. Constrained genetic optimization via dynamic reward-penalty balancing and its use in pattern recognition. In Int. Conf. Genetic Algorithms, pages 141–149, 1989.

[27] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. Evol. Comput., 2(3):221–248, Fall 1995.

[28] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications. PhD thesis, ETH 13398, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1999.

[29] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, 8:173–195, 2000.

[30] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In M. Schoenauer A. E. Eiben, T. B¨ack and H.-P. Schwefel, editors, Parallel Problem Solving From Nature, pages 292–301. Springer-Verlag, 1998.

[31] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, 7:117–132, 2003.

APPENDIX

All benchmark functions are described in [8].

X. CTP1

$$Min\ f_1(x) = x_1$$

$$Min\ f_2(x) = g(x) e^{\frac{-f_1(x)}{g(x)}}$$

Where

$$g(x) = 1.0 + x_2$$

Subject to :

$$g_1(x) = f_2(x) - a_1 e^{-b_1 f_1(x)} \geq 0$$

$$g_2(x) = f_2(x) - a_2 e^{-b_2 f_1(x)} \geq 0$$

Where $a_1 = 0.858$, $b_1 = 0.541$, $a_2 = 0.728$, $b_2 = 0.295$.

XI. CTP2

$$Min\ f_1(x) = x_1$$

$$Min\ f_2(x) = g(x)\left(1 - \frac{-f_1(x)}{g(x)}\right)$$

Where

$$g(x) = 1.0 + x_2$$

Subject to:

$$g(x) = \cos(\theta)\left[f_2(x) - e\right] - \sin(\theta) f_1(x) \geq$$

$$a \left| \sin\left\{ b\pi \left[\sin(\theta)(f_2(x) - e) + \cos(\theta) f_1(x) \right]^c \right\} \right|^d$$

Where $x_1 = [0,1]$, and $\theta = -0.2\pi$, $a = 0.2$, $b = 10$, $c = 1$, $d = 6$, $e = 1$.

XII. CTP3

$$Min\ f_i(x) = x_1$$

$$Min\ f_i(x) = g(x)\left(1 - \frac{-f_1(x)}{g(x)}\right)$$

Where

$$g(x) = 1.0 + x_2$$

Subject to:

$$g_i(x) = \cos(\theta)\left[f_2(x) - e\right] - \sin(\theta) f_1(x)$$

$$\geq a \left| \sin\left\{ b\pi \left[\sin(\theta)\left(f_2(x) - e\right) + \cos(\theta) f_1(x) \right]^c \right\} \right|^d$$

Where $x_1 = [0,1]$, $\theta = -0.2\pi, a = 0.1$, $b = 10, c = 1, d = 0.5, e = 1$.

XIII. CTP4

$$Min\ f_i(x) = x_1$$

$$Min\ f_i(x) = g(x)\left(1 - \frac{-f_1(x)}{g(x)}\right)$$

Where

$$g(x) = 1.0 + x_2$$

Subject to:

$$g_i(x) = \cos(\theta)\left[f_2(x) - e\right] - \sin(\theta) f_1(x)$$

$$\geq a \left| \sin\left\{ b\pi \left[\sin(\theta)\left(f_2(x) - e\right) + \cos(\theta) f_1(x) \right]^c \right\} \right|^d$$

Where $x_1 = [0,1]$, $\theta = -0.2\pi, a = 0.75$, $b = 10, c = 1, d = 0.5, e = 1$

XIV. CTP5

$$Min\ f_i(x) = x_1$$

$$Min\ f_i(x) = g(x)\left(1 - \frac{-f_1(x)}{g(x)}\right)$$

Where

$$g(x) = 1.0 + x_2$$

Subject to:

$$g_i(x) = \cos(\theta)\left[f_2(x) - e\right] - \sin(\theta) f_1(x)$$

$$\geq a \left| \sin\left\{ b\pi \left[\sin(\theta)\left(f_2(x) - e\right) + \cos(\theta) f_1(x) \right]^c \right\} \right|^d$$

Where $x_1 = [0,1]$, $\theta = -0.1\pi, a = 40$, $b = 0.5, c = 1, d = 2, e = -2s$.