An Efficient Multi-objective Evolutionary Algorithm with Steady-State Replacement Model

Dipti Srinivasan[1] Lily Rachmawati[2]

Department of Electrical and Computer Engineering

National University of Singapore
4 Engineering Drive 3
Singapore 117576
(65) 65166544

[2]dipti@nus.edu.sg; [1]g0402564@nus.edu.sg

ABSTRACT

The generic Multi-objective Evolutionary Algorithm (MOEA) aims to produce Pareto-front approximations with good convergence and diversity property. To achieve convergence, most multi-objective evolutionary algorithms today employ Pareto-ranking as the main criteria for fitness calculation. The computation of Pareto-rank in a population is time consuming, and arguably the most computationally expensive component in an iteration of the said algorithms. This paper proposes a Multi-objective Evolutionary Algorithm which avoids Pareto-ranking altogether by employing the transitivity of the domination relation. The proposed algorithm is an elitist algorithm with explicit diversity preservation procedure. It applies a measure reflecting the degree of domination between solutions in a steady-state replacement strategy to determine which individuals survive to the next iteration. Results on nine standard test functions demonstrated that the algorithm performs favorably compared to the popular NSGA-II in terms of convergence as well as diversity of the Pareto-set approximation, and is computationally more efficient.

Categories and Subject Descriptors

I.2.M Computing Methodologies

General Terms

Algorithms

Keywords

Multi-objective optimization, Genetic algorithms

1. INTRODUCTION

The optimization of multiple conflicting objectives is a common denomination for many problems in engineering. In the presence of conflicting objectives, no single solution achieves optimality in all objectives. Pareto-optimality, proposed by eminent economists Pareto and Edgeworth, captures the inevitability of trade-off between non-commensurable objectives in a criterion commonly employed in multi-objective optimizations [1]. A solution is Pareto-optimal if no other solution could be found in the feasible region which performs better in at least one objective and equivalent or better in the rest. According to the criterion a multi-objective optimization has a set of Pareto-optimal solutions. The image of this set of solutions in the objective space, the Pareto-front, displays the objective trade-off characteristics for the problem.

Such MOEAs employ fitness functions based on the Pareto domination relation to achieve selection pressure towards the Pareto-optimal front. Domination is defined as follows [1]. For a set of objectives $\mathbf{F} = \{f_1, f_2, \ldots f_m, \ldots, f_M\}$ and solutions $\mathbf{X_i}$ and $\mathbf{X_j}$, solution $\mathbf{X_j}$ dominates $\mathbf{X_i}$ if and only if:

1. $f_m^j \le f_m^i$ for all $m \in [1, M], m \ne n$
2. $f_n^j < f_n^i$ for $1 \le n \le M$

Otherwise, the two solutions are said to be non-dominated. The Pareto-optimal set consists of solutions which are non-dominated with respect to all other feasible solutions.

The common practice in MOEAs is to construct a Pareto-rank, or domination-rank, which expresses the candidate solutions' extent of domination in the population. This population may be taken from the current population and/or an archive population. Among solutions with identical Pareto-rank, selection is conducted based on a crowding measure. While Pareto-rank promotes convergence to the Pareto-front, crowding measure promotes diversity on the front. NPGA [2], PESA-II [3], PAES [4], NSGA-II [5], SPEA2 [6], HPMOEA [7], and OMOEA-II [8] are some of the more recent representatives of high-performing MOEAs designed on these principles.

Pareto-ranking promotes convergence of the algorithm, but is widely recognized as an expensive procedure. In the algorithms mentioned previously, Pareto-rank assigns candidate solutions a score from the number of candidate solutions which dominate and/or are dominated by it. The ranking process is arguably the most computationally-expensive building block of an MOEA, even in highly-efficient NSGA-II [5], forming the lower bound on the computational complexity of the MOEA.

Improving the efficiency of MOEAs is a concern of several recent papers. Jensen [9] proposed modified data structures and search algorithms to increase the efficiency of Pareto-ranking procedures

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-186-4/06/0007…$5.00.

in some of the more established MOEAs. Although the approach shows promise, tangible results are minimal. Researchers have proposed algorithms which altogether avoid expensive Pareto-ranking. The Artificial Life community has proposed algorithms circumventing the need for expensive Pareto-ranking [10-12]. Because of the computational model, explicit Pareto-ranking may be replaced by simple binary-tournament-style interaction of individuals. SEAMO [13], proposed in 2004, implements a steady-state evolutionary algorithm with simple replacement strategy based on pair-wise comparison of generated child against a subset of the population. Although the performance is comparable to NSGA-II on some test functions, SEAMO does not fare well in the challenging test problem ZDT6 [14].

In this paper the authors propose a MOEA which relies on a steady-state replacement strategy based on a measure of domination degree instead of domination extent. The algorithm is not a steady-state evolutionary algorithm. The term steady-state refers to the serial manner in which each of the children produced in the current iteration is considered for propagation to the next iteration. The steady-state replacement strategy exploits the transitivity of the domination criteria to achieve selection pressure towards the Pareto-front with lower computational cost.

The paper is organized as follows. In section 2 a study on the role of Pareto-ranking including a review of representative MOEAs with a highlight on the fitness assignment scheme and selection procedure is given. In section 3 the proposed algorithm is described, followed by the simulation results on nine well-known test problems in section 4. The results show that the proposed algorithm provides a simple and inexpensive alternative to Pareto-ranking, and performs extremely well on the test problems.

2. PARETO-RANKING IN MOEA

Selection is a key factor in an Evolutionary Algorithm. Since Goldberg's suggestion [15], MOEAs have been designed with a selection strategy dependent on Pareto-ranking along with niching mechanism. In this section we present a review of a number of well-known baseline MOEAs with a focus on the selection criteria. A baseline MOEA here denotes an MOEA comprising the underlying framework of population initiation, selection, off-spring generation and population replacement.

In the early nineties, Horn et al. (NPGA [2]) proposed an algorithm with restricted selection strategy. Population replacement is accomplished by binary tournaments with reference to a randomly selected subset of the current population. If either of the randomly-picked candidates is dominated by any member of the reference set, the other is selected to proceed to the next generation. If both or neither are dominated by the reference set, a fitness sharing score based on diversity breaks the tie. The effect of this approximation is increased computational efficiency but decreased Pareto-approximation quality compared to algorithms employing a complete population wide Pareto-ranking.

The (1+1) PAES [4] maintains a single-member population evolved through mutation and a non-dominated solution archive. Niching is achieved by dividing the phenotype space adaptively into identical-sized hyper-cubes. Inclusion into the archive is decided by pair-wise comparison between parent and offspring, with domination as the main criteria and grid location as the second criteria.

In $(1+\lambda)$ PAES and $(\mu+\lambda)$ PAES fitness is calculated based on a domination score and grid location. Domination score of -1 is assigned to solutions dominated by any member of the archive, while the score is equal to the number of solutions in the archive dominated by the solution. Solutions with better domination score always have better fitness regardless of its grid location. Binary tournament for parent selection and inclusion into the archive are based on this fitness score.

SPEA2 [6] also maintains an archive of non-dominated solutions. A strength figure, defined as the number of dominating solutions in the combined archive and current population, is computed for each member of the population. A solution is assigned a fitness denoting the sum of the strength of other solutions in the combined population which dominate it, and a density figure which is the inverse of the distance to the k-th nearest neighbor in the combined population. Binary tournament with replacement is applied to the combined population to obtain candidates for recombination and mutation. The archive at the next iteration includes solutions with fitness lower than 1 (non-dominated with respect to the combined population). A constant archive size is maintained, by including solutions with the next best fitness values or by pruning the current members of the archive according to the distance measure, as required.

In [16], the fitness of a solution is determined as the number of solutions in the current population which dominates it. Deb et al. in [5] proposed a highly efficient non-dominated sorting algorithm to compute this non-domination rank. Non-dominated sorting iteratively finds the layers of non-dominated fronts. NSGA-II applies the non-domination rank to the parent selection and generation replacement. A fast crowding-distance-estimation is employed to break ties in parent selections in binary tournaments. The population at the next iteration is obtained by selecting solutions from the combined offspring-parent population layer by layer from the best non-dominated front onwards, with the crowding measure to break ties between solutions on the same front.

Recognizing the complexity of fitness assignment based on population wide Pareto-ranking, Mumford proposed a steady state evolutionary algorithm (SEAMO [13]) with limited comparisons in the generation replacement strategy. The steady state algorithm compares each generated offspring with the parents. A comparison between parents and offspring is conducted and the dominated individual is discarded. In the case of non-domination the algorithm selects randomly from the population an individual which is dominated by the offspring. This individual is then replaced by the offspring. An additional selection rule allowing a non-dominated offspring to replace the parent when the offspring is more desirable could be inserted to enhance the spread of the solutions.

Studies conducted in the performance of the multi-objective algorithms ([5-6], [13-14]) show that NSGA-II and SPEA2 are among the most competitive algorithms for a set of difficult test problems. Population wide comparison aided by elitism seems to be crucial to the convergence property of the algorithm. NPGA performs poorly because it approximates Pareto-rank with comparisons against a fraction of the population.

Even in the absence of Pareto-ranking, population-wide comparison based on the Pareto-domination criteria unsurprisingly seems to improve the performance of the algorithm. In [13] a variety of population replacement strategy was examined. It was observed that Pareto-front approximation obtained was improved when the entire population instead of only the parents was considered in deciding whether or not to include the offspring.

Pareto-rank is a population-dependent measure. The phenomenon above can be understood if we consider the non-transitivity of the non-domination relation. Domination in the Pareto-optimality sense imposes a partial order on the population which is transitive and asymmetric [1]. The induced non-domination relation, is not transitive. Because of this non-transitivity, pair-wise comparison between two solutions is strictly necessary to establish domination or non-domination between them except in a special case. This is explained via Figure 1 which shows three candidate solutions A, B and C in a problem with two objective functions in the objective space.

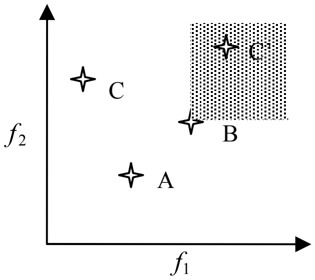

Figure 1 Non-transitivity in Pareto non-domination relation

The domination or non-domination of the points A, B and C with respect to one another can only be obtained by comparing each pair, except when C is located in the shaded region, where the transitivity of the Pareto-domination relation may be used to infer the relation of A and C given the relation between A and B, and B and C.

With more comparisons more information is available on how close a candidate solution is to the Pareto-front in comparison to other solutions in the population. Accordingly, the information gained avails a more accurate selection pressure. Whether Pareto-ranking or some other mechanism is employed in the selection, when Pareto-optimality criterion is used population-wide comparison benefits the convergence property of the acquired Pareto-front approximation. In the next section a multi-objective genetic algorithm without explicit Pareto-ranking is proposed. It employs population-wide comparison, but in such a way that the special case mentioned above is exploited to reduce the computation time. Because of the population-wide comparison, the complexity of the algorithm is as NSGA-II $O(MN^2)$. However the constant term in the complexity is lower than that in NSGA-II. Over many iterations this constant factor will bring about a significant difference in actual computation time, as is demonstrated in section IV.

At this juncture the authors would like to highlight that the focus of this paper is on the baseline MOEA. Many modifications and enhancements to the above-mentioned baseline MOEA have been proposed to improve the efficiency and effectiveness. The baseline algorithms have also been extended to address more specific problems in multi-objective optimization. Adaptive PAES [17] modify the representation scheme of the standard PAES and thereby improved the performance. The framework of NSGA-II has been used with modification in offspring generation mechanism in OMOEA and OMOEA-II [8], combined with a thermodynamic-inspired Gibb's entropy figure to form the rank of a particular solution in HPMOEA[7], employed in conjunction with the S-measure as the secondary selection criteria in SMS-EMOEA[18] to produce highly effective algorithms with very good convergence and distribution characteristics.

Many other variants have been introduced. However, the paper is not concerned with enhanced versions of the algorithms. The design of the proposed algorithm outlined in the next section as well as the simulation results and performance should be viewed in this light. It is to be expected that the modifications and enhancement correctly applied to the proposed algorithm will yield similar improvements on the performance.

3. THE PROPOSED APPROACH

The baseline multi-objective genetic algorithm proposed here employs an elitism strategy similar to that in NSGA-II, but is distinct in that no Pareto-ranking is performed on the population. Parent selection is conducted in a simple binary tournament while generation replacement is achieved through a steady-state procedure based on the notion of the degree of domination.

The selection strategy in general is distinct from that employed in Horn and Nafpliotis' NPGA in two respects. Firstly NPGA in effect conducts Pareto-ranking within a subset of existing candidate solutions at any particular iteration whereas the proposed algorithm does not employ Pareto-ranking. Secondly the steady-state replacement model includes comparisons with all, instead of a randomly selected subset, of existing solutions at any particular iteration.

3.1 The main loop

Initially a population of N candidate solutions is randomly generated, and the objective function values of the chromosomes are computed. Let us denote this as $\mathbf{F}_{\text{Parent}}$, an $N \times M$ array where N is the size of the population and M is the number of objectives to be minimized. Binary tournament with replacement is employed to select parents from the current population based on the Pareto-domination criteria. At this point no ranking of candidate solutions is necessary. The two solutions picked at random to enter the tournament, S_1 and $S_{2,}$ are compared as shown in Figure 2:

Crossover and mutation are applied as usual to the two parents to obtain two children. A child population of size N is generated such that no duplicates of the parent population and the child population are allowed. In the event that a child chromosome is identical to another in either the child or parent population, the chromosome is destroyed and crossover and mutation is repeated to obtain a new offspring. At the completion of the child population creation, the objective functions are evaluated for each offspring and stored as an $N \times M$ array $\mathbf{F}_{\text{Childr}}$. The arrays $\mathbf{F}_{\text{Parent}}$ and $\mathbf{F}_{\text{Childr}}$ are used to determine which child should replace which member of the parent population in the generation replacement described in the following section. The resulting N-member population proceeds to the next generation and the process is repeated until the termination criterion is fulfilled.

```
If $S_1 \succ S_2$
      then parent = $S_1$
else
     if $S_2 \succ S_1$
          then parent = $S_1$
     else
          if (crowding_distance($S_1$) > crowding_distance($S_2$))
                   then parent = $S_1$
          else
               if (crowding_distance($S_2$) > crowding_distance($S_1$))
                        then parent = $S_2$
               else
                        pick randomly n $\in (S_1, S_2)$
                        parent = n
```

Figure 2 Parent selection criteria 1

3.2 Generation replacement

In the replacement strategy, each child is considered in succession. If it is dominated by any member of the parent population, it is discarded. If this individual is non-dominated with respect to all members of the parent population, the crowding criteria is invoked. Otherwise, for the set of members from the parent population which are dominated by this individual, a degree of domination is computed and the member of the parent population corresponding to the largest degree of domination is replaced by this child. The degree of domination is computed as the dynamically-scaled difference between the child's objective function values and those of the dominated individuals in the parent population.

In summary, generation replacement is accomplished as follows. For each child C_j, subtract from the rows of $\mathbf{F}_{\text{Parent}}$ row j of $\mathbf{F}_{\text{Child}}$ and store the result in a $N \times M$ matrix Γ_j. For corresponding members of the parent population which dominate or are dominated by the child C_j (this may be determined by checking the sign of the elements in the respective rows in Γ_j), compute the sum of the corresponding rows in Γ_j and store this in a $N \times 1$ array Δ_j.

Elements of array Δ_j that correspond to members in the parent population which are non-dominated with respect to the child C_j are set to zero. The array Δ_j is sorted and the member of the parent population corresponding to the largest positive number in Δ_j is replaced by the child C_j. If no chromosome in the parent population corresponds to a positive number in Δ_j, the negative numbers are considered. If no negative number exists, i.e. if the entire parent population is non-dominated with respect to the child C_j, then the distance preservation measure is applied to choose the member of the parent population to be replaced by C_j. The crowding distance estimation in NSGA-II is employed for the (parent + C_j) population and the individual with the smallest crowding distance is removed from the population.

The array Δ_j approximates the degree of domination between the child C_j and members of the parent population if domination exists. To avoid one objective dominating another in the minimization of a non-commensurable set of objectives, the values in Γ_j are scaled dynamically (Figure 3). The scale is obtained as the normalized difference between the largest and smallest values of each respective objective function obtained in the current parent population.

```
for each child j
     for each parent k
          sign = 0
          for each objective m
               $\Gamma_j(k,m) = F_{\text{Parent}}(k,m) - F_{\text{Child}}(j,m)$
               sign = sgn($\Gamma_j(k,m)$) + sign

          if (abs(sign) = M)
             $\Delta_j = \sum_m (\Gamma_j(k,m) / scale_m)$
          else
             $\Delta_j = 0$

     sort ( $\Delta_j$)
     if (max($\Delta_j$) > 0)
          replace parent corresponding to max($\Delta_j$) with child
   else
          if (max($\Delta_j$) = 0 and min($\Delta_j$) = 0)
               compute crowding_distance(POP + ChildPOP$_j$)
               discard individual with minimum crowding_distance
```

Figure 3 Generation replacement with adaptive scaling

An alternative strategy to determine the degree of domination is illustrated in Figure 4. At each replacement round involving a member of the child population, one of the *M* objectives, say objective *k*, is randomly selected. Note that only the members of the parent population which are dominated by the child population are considered. The individual scoring the worst in objective *k* is discarded (Figure 4).

```
for each child j
     i = rand(1,M) // i is a random number in [1,M]
     for each parent k
          sign = 0
          for each objective m
                sign = sgn($F_{\text{Parent}}(k,m) - F_{\text{Child}}(j,m)$) + sign


          if (abs(sign) = M)
             $\Delta_j = F_{\text{Parent}}(k,i) - F_{\text{Child}}(j,i)$
          else
             $\Delta_j = 0$

    sort ( $\Delta_j$)
     if (max($\Delta_j$) > 0)
          replace parent corresponding to max($\Delta_j$) with child
    else
          if (max($\Delta_j$) = 0 and min($\Delta_j$) = 0)
               compute crowding_distance(POP + ChildPOP$_j$)
               discard individual with minimum crowding_distance
```

Figure 4 Alternative generation replacement

The generation replacement algorithm presented above was described for the minimization of objectives. Maximization problems may be handled with trivial transformation of the objective functions.

3.3 Selection without Pareto-rank

Parent chromosome is selected in a simple binary tournament comparing two parent candidates based on the domination relationship between them instead of the extent of their domination in the population. When both are non-dominated with respect to each other, the one with the largest crowding distance is selected. The strategy is pursued based on the observation that the domination relationship between two parent candidates is sufficient to approximate the relative quality of the parent candidates in terms of their proximity to the actual Pareto-front. Binary tournament based on Pareto-ranking will yield the same winner for cases where one parent candidate dominates the other. A difference in the selected parent may occur where both candidates are non-dominated. Noting that solutions far away from each other are likely to produce better offspring in crossover, a candidate occupying the sparser region is selected in the proposed algorithm. This allows a greater possibility of selecting parents far away from each other.

The generation replacement algorithm implements elitism by retaining the best members of [parent + j-th child (C_j)] population. The replacement strategy is steady state in that each child is considered in succession. The generation replacement strategy is insensitive to the order in which the child solution is considered. It takes advantage of the transitivity of the domination relation described as the special case in Section II. In the same iteration of the main loop, no member of the current population is ever replaced by an inferior solution in the Pareto-optimality sense. By exploiting this property, a steady-state replacement is able to weed out solutions based on the degree of domination in a simple manner.

The crowding distance criterion is considered only when the (parent + child_j) population are non-dominated. The crowding-distance criterion is transitive and likewise replacements based on the criterion do not deteriorate the spread of the population within a single iteration of the main loop.

4. RESULTS AND DISCUSSION

To examine the performance of the proposed algorithm, we conducted a study with nine bi-objective test functions. These test problems are widely employed in studies of MOEAs, exhibiting features which often make it difficult for multi-objective GAs to converge to the Pareto-front. The test problems examine the performance of multi-objective GA in the presence of a large number of decision variables, non-convex and disconnected regions in the Pareto-front as well as multi-modality. A summary of the test functions is provided in Table 1. For further description of the test functions the reader is referred to [19-22] and [14-15].

The comparison with NSGA-II was undertaken because of its competitive performance in terms of convergence, solution diversity and computational efficiency. Other similar comparative studies evaluating the performance of NSGA-II against other similarly competitive algorithms are available for reference.

The simulations were designed very closely to that in [5]. Initial solutions are generated randomly. The chromosomes were real-coded. Simulated binary crossover operator [23] and single point mutation were employed with crossover probability of 0.9 and mutation probability of 0.08.

Table 1 Test functions

Problem	Objective Functions	Features
SCH	$f_1 = x^2;\ f_2 = (x-2)^2$ $x \in [-10^3, 10^3]$	Convex
KUR	$f_1 = \Sigma_{i=1}^{3}[-10\exp(-0.2\sqrt{x_i{}^2 + x_{i+1}{}^2}]$ $f_2 = \Sigma_{i=1}^{3}[\lvert x_i \rvert^{0.8} + 0.5\sin(x_i^3)]$ $x_i \in [-5,5]; i = 1,2,3$	Nonconvex, disconnected Pareto-front
FON	$f_1 = 1 - \exp\left(-\Sigma_{i=1}^{3}(x_i - \frac{1}{\sqrt{3}})^2\right)$ $f_2 = 1 - \exp\left(-\Sigma_{i=1}^{3}(x_i + \frac{1}{\sqrt{3}})^2\right)$ $x_i \in [-4,4]; i = 1,2,3$	Nonconvex
POL	$f_1 = [1 + (A_1 - B_1)^2 + (A_2 - B_2)^2]$ $f_2 = [(x_1 - 3)^2 + (x_2 - 1)^2]$ $A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\sin 2$ $A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\sin 2$ $A_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\sin x_2$ $B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\sin x_2$ $x_{1,2} \in [-\pi, \pi]$	Nonconvex, disconnected Pareto-front
ZDT1	$f_1(x) = x_1; f_2(x) = g(x)h(f_1, g)$ $g(x) = 1 + \frac{9}{n-1}\Sigma_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{f_1 / g}$ $x_i \in [0,1]; i = 1,2,..n$ $n = 30$	Convex
ZDT2	$f_1(x) = x_1; f_2(x) = g(x)h(f_1, g)$ $g(x) = 1 + \frac{9}{n-1}\Sigma_{i=2}^{n} x_i$ $h(f_1, g) = 1 - (f_1 / g)^2$ $x_i \in [0,1]; i = 1,2,..n$ $n = 30$	Nonconvex
ZDT3	$f_1(x) = x_1; f_2(x) = g(x)h(f_1, g)$ $g(x) = 1 + \frac{9}{n-1}\Sigma_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{f_1 / g} - (f_1 / g)\sin(10\pi f_1)$ $x_i \in [0,1]; i = 1,2,..n$ $n = 30$	Disconnected Pareto-front
ZDT4	$f_1(x) = x_1; f_2(x) = g(x)h(f_1, g)$ $g(x) = 1 + 10(n-1) + \Sigma_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{f_1 / g}$ $x_1 \in [0,1]$ $x_i \in [-5,5]; i = 2,..n\ ; n = 10$	Nonconvex; ±21^9 local Pareto-front
ZDT6	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1); f_2(x) = g(x)h(f_1, g)$ $g(x) = 1 + 9[(\Sigma_{i=2}^{n} x_i)/9]^{0.25}$ $h(f_1, g) = 1 - (f_1 / g)^2$ $x_i \in [0,1]; i = 1,2,..n$ $n = 10$	Nonconvex; non-uniform density of Pareto- front

To investigate the effect of avoiding Pareto-rank in the parent selection criteria, simulations with population of size 100 for 100 iterations are conducted for two parent selection criteria. The criterion in Figure 2 does not employ Pareto-rank while that in Figure 5 considers Pareto-rank computed using the non-dominated sorting in NSGA-II [5]. Figure 6 illustrates the Pareto-front approximations obtained for the test problem KUR. From the figure, it can be seen that no discernable difference exists in the Pareto-front approximations obtained with the two parent selection criteria.

```
If Pareto-rank($S_1$) < Pareto-rank($S_2$)
    then parent = $S_1$
else
    if Pareto-rank($S_1$) > Pareto-rank($S_2$)
        then parent = $S_1$
    else
        if (crowding_distance($S_1$) > crowding_distance($S_2$))
            then parent = $S_1$
        else
            if (crowding_distance($S_2$) > crowding_distance($S_1$))
                then parent = $S_2$
            else
                pick randomly $n \in (S_1, S_2)$
                parent = n
```

Figure 5 Parent selection criteria 2

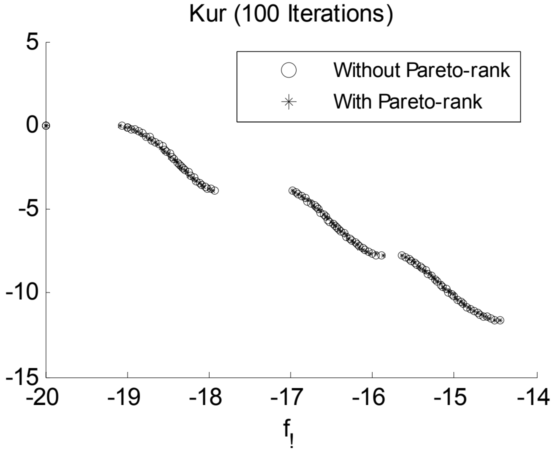

Figure 6 Test problem KUR after 100 iterations: parent selection with and without Pareto-ranking

To evaluate the performance of the overall algorithm, the proposed approach was run five times for each test problem, with a population of size 100 evolved over 250 iterations. The experiments were conducted in MATLAB, on a Pentium IV3.00 GHz CPU.

Repeated simulations suggested equivalent performance of generation replacement strategies described in Figure 3 and Figure 4. For the results quoted below parent selection criteria described in Figure 1 and generation replacement strategy in Figure 3 were employed. Figure 7 shows all non-dominated solutions obtained after 250 generations of the proposed algorithm. The CPU time for the proposed approach averaged at 20 seconds, while that for NSGA-II is 9 minutes and 17 seconds.

From the resulting final populations the average and variance of a convergence-metric and a diversity-metric defined in [5] were evaluated. The convergence-metric (Υ) evaluates the proximity of the obtained solutions to uniformly-spread points in the true Pareto-front while the diversity-metric (Δ) measures the spread and span of the obtained solutions. The results of this study are reproduced in Table 2 and 3. Results for a similar experiment on NSGA-II are quoted for comparison.

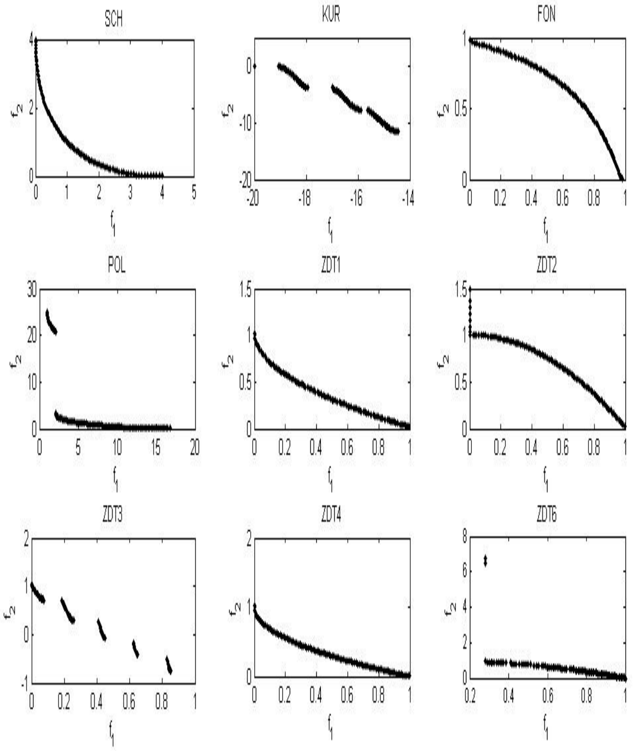

Figure 7 Pareto-front approximation for the 9 test problems

Table 2 Convergence Metric (Υ)

Problem	Real-Coded NSGA II ([2])		Proposed Approach	
	Mean	Variance	Mean	Variance
SCH	0.0033910	0.0000000	0.0032526	0.0000000
KUR	0.0289640	0.0000180	0.0076588	0.0000008
FON	0.0019310	0.0000000	0.0017750	0.0000000
POL	0.0155530	0.0000010	0.0109732	0.0000006
ZDT1	0.0334820	0.0047500	0.0184504	0.0000307
ZDT2	0.0723910	0.0316890	0.0212754	0.0000920
ZDT3	0.1145000	0.0079400	0.0091456	0.0000004
ZDT4	0.5130530	0.1184600	0.0020248	0.0000010
ZDT6	0.2965640	0.0131350	0.0037373	0.0000000

The set of results in Tables 2 and 3 show that in terms of convergence and diversity, the elimination of Pareto-rank in the generation replacement does not have any negative effect on the Pareto-front approximation yielded by the proposed algorithm. In fact, the proposed algorithm performed comparably or better than real coded NSGA-II in the metrics defined for the 9 test problems. Likewise, the convergence is better or comparable to that obtained in NSGA-II considering the crowding distance measure once the (parent + child$_j$) population is non-dominated. This improvement performance in terms of spread and convergence may be explained by the more aggressive generation replacement strategy used in the proposed approach. A child only replaces a dominated member in the parent population with the highest domination degree. When the child is non-dominated by the members of the parent population, the crowding distance measure is immediately considered.

Table 3 Diversity Metric (Δ)

Problem	Real-Coded NSGA II ([2])		Proposed Approach	
	Mean	Variance	Mean	Variance
SCH	0.4778990	0.0034710	0.0000259	0.0000000
KUR	0.4147700	0.0009920	0.0000332	0.0000000
FON	0.3780650	0.0006390	0.0010621	0.0000003
POL	0.4521500	0.0028680	0.0002600	0.0000000
ZDT1	0.3903070	0.0018760	0.0428000	0.0001963
ZDT2	0.4307760	0.0047210	0.0352000	0.0001430
ZDT3	0.7385400	0.0197060	0.0020600	0.0000001
ZDT4	0.7061200	0.0646480	0.0117800	0.0000261
ZDT6	0.6680250	0.0099230	0.0759975	0.0173058

The weakness of the proposed algorithm is that like NSGA-II and many other baseline MOEA [24] the population may deteriorate across generations, i.e. at iteration *j* the population may contain solutions inferior in the Pareto-optimality sense to those at iteration *i,*where *i<j.*

5. CONCLUSION

A baseline multi-objective genetic algorithm was proposed which did not employ Pareto-ranking. The performance of the algorithm is generally not inferior to the well-known NSGA-II on nine difficult test problems. This indicates that convergence to the true Pareto-front may be promoted by a fitness measure based on the estimated degree of domination where such domination exists, and a steady-state population replacement strategy.

Although the population replacement strategy involves population-wide comparison in terms of domination, this was performed in a manner which exploits the transitivity of the Pareto-domination relation and effectually decreases the constant term in the $O(MN^2)$ complexity term. This results in potentially significant time saving over multiple iterations of the algorithm. The lower computational cost may be highly beneficial especially for applications where Pareto-ranking accounts for a considerable portion of the computation time.

6. REFERENCES

[1] K. Deb, Multi-Objective Optimization using Evolutionary Algorithm. Chichester, England: John Wiley & Sons, 2001.

[2] J. Horn, N. Nafpliotis, and D.E. Goldberg, "A niched Pareto Genetic algorithm for multiobjective optimization," in Proceedings of the First IEEE Conference on Evolutionary Computation, Z. Michalewicz, Ed. Piscataway, NJ: IEEE Press, 1994, pp. 82-87.

[3] D. W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, "PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization", in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, Morgan Kaufmann Publishers, pp. 283-290.

[4] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation.* Piscataway, NJ: IEEE Press, 1999, pp. 98-105.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elititst Multiobjective Genetic Algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, April 2002, pp. 182-197

[6] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", *Evolutionary Methods for Design, Optimization and Control,* 2002.

[7] X. Zou, M. Liu, L. Kang, and J. He, "A High Performance Multi-objective Evolutionary Algorithm Based on the Principles of Thermodynamics, *Eighth Int. Conf. Parallel Problem Solving from Nature (PPSN VIII)*, X. Yao et al. Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 922-931.

[8] S. Zeng, S. Yao, L. Kang, and Y. Liu, "An Efficient Multi-objective Evolutionary Algorithm: OMOEA-II", *EMO 2005*, C.A. Coello Coello et. al. Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 108-119.

[9] M.T. Jensen, "Reducing the Run-time Complexity of Multi-Objective EAs: The NSGA-II and Other Algorithms", *IEEE Transactions on Evolutionary Computation,* vol. 7, no. 5, 2003, pp.503-515

[10] K. Socha and M. Kisiel-Dorohinicki, "Agent-Based Evolutionary Multiobjective Optimization", in *Proceedings of the 2002Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 2002, pp. 109-114.

[11] M. Laumanns, G. Rudolph, and H.-P Schwefel, "A Spatial Predator-Prey Approach to Multi-Objective Optimization: a Preliminary Study", in *Fifth Int. Conf. on Parallel Problem Solving from Nature(PPSN V),* A.E. Eiben, T. Back, M. Schonauer, and H.-P. Schwefel, Eds. Berline, Germany: Springer-Verlag, 1998, pp. 241-249

[12] A. Berry and P. Vamplew, "The Combative Accretion Model – Multiobjective Optimization Without Explicit Pareto Ranking", *EMO 2005,* C. A. Coello Coello et al. Eds, Berlin: Springer-Verlag, 2005, pp. 77-91.

[13] C. L. Mumford, "Simple Population Replacement Strategies for a Steady-State Multi-objective Evolutionary Algorithm", *GECCO 2004*, K. Deb et al. Eds. Berlin: Springer-Verlag, 2004, pp. 1389-1400.

[14] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.,* vol. 8, no. 2, pp. 173-195, Summer 2000.

[15] D.E. Goldberg,, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

[16] C.M. Fonseca and P.J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part I: A unified formulation," *IEEE Trans. Syst., Man., Cybern. A*, vol 28, pp. 38-47, Jan. 1998

[17] M. Oltean, C. Grosan, A. Abraham, and M. Koppen, "Multiobjective Optimization Using Adaptive Pareto Archive Evolution Strategy", *5th Int. Conf. on Intelligent*

Sytems Design and Applications (ISDA'05) Piscataway, NJ: IEEE Press, 2005, pp. 558-563.

[18] M. Emmerich, N. Beune, and B. Naujoks, "An EMO Algorithm Using the Hypervolume Measure as Selection Criterion," in *EMO 2005,* C. A. Coello Coello et al. Eds. Berlin: Springer-Verlag, 2005, pp. 62-76

[19] J.D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the First International Conference on Genetic Algorithms,* J.J. Grefensttete, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 93-100.

[20] C.M. Fonseca and P.J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part II: Application example," *IEEE Trans. Syst., Man., Cybern. A*, vol 28, pp. 38-47, Jan. 1998.

[21] C. Poloni, "Hybrid GA for multiobjective aerodynamic shape optimization," in *Genetic Algorithms in Engineering and Computer Science,* G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds. New York: Wiley, 1997, pp. 397-414.

[22] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature,* H.-P Schwefel and R.Manner, Eds Berlin, Germany: Springer-Verlag, 1990, pp.193-197.

[23] K. Deb and S. Agrawal, "Simulated binary crossover for continuous search space," in *Complex Syst.,* Apr. 1995, vol. 9, pp. 115-148.

[24] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization", *Evolutionary Computations,* vol. 10, no. 3, 2002, pp. 263-282.