Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions

Kalyanmoy Deb deb@iitk.ac.in
Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, Kanpur, PIN 208016, INDIA

Manikanth Mohan manikanthm@gmail.com
Manikanth Mohan, Palappallil House, Nalkalickal P.O., (via) Aranmula, Pathnamthitta (Dist), Kerala, PIN 689533, INDIA

Shikhar Mishra smishr01@arch.cs.umsl.edu
Department of Mathematics and Computer Science, University of Missouri, St. Louis, MO 63121, USA

Abstract

Since the suggestion of a computing procedure of multiple Pareto-optimal solutions in multi-objective optimization problems in the early Nineties, researchers have been on the look out for a procedure which is computationally fast and simultaneously capable of finding a well-converged and well-distributed set of solutions. Most multi-objective evolutionary algorithms (MOEAs) developed in the past decade are either good for achieving a well-distributed solutions at the expense of a large computational effort or computationally fast at the expense of achieving a not-so-good distribution of solutions. For example, although the Strength Pareto Evolutionary Algorithm or SPEA (Zitzler and Thiele, 1999) produces a much better distribution compared to the elitist non-dominated sorting GA or NSGA-II (Deb et al., 2002a), the computational time needed to run SPEA is much greater. In this paper, we evaluate a recently-proposed steady-state MOEA (Deb et al., 2003) which was developed based on the ϵ-dominance concept introduced earlier(Laumanns et al., 2002) and using efficient parent and archive update strategies for achieving a well-distributed and well-converged set of solutions quickly. Based on an extensive comparative study with four other state-of-the-art MOEAs on a number of two, three, and four objective test problems, it is observed that the steady-state MOEA is a good compromise in terms of convergence near to the Pareto-optimal front, diversity of solutions, and computational time. Moreover, the ϵ-MOEA is a step closer towards making MOEAs pragmatic, particularly allowing a decision-maker to control the achievable accuracy in the obtained Pareto-optimal solutions.

Keywords

Multi-objective optimization, evolutionary algorithms, genetic algorithms, Pareto-optimal solutions, ϵ-dominance, computational effort, convergence measure, sparsity measure, hyper-volume metric

1 Introduction

By definition, a search and optimization problem with multiple conflicting objectives resorts to a set of optimal solutions known as the Pareto-optimal solutions. During the recent past, multi-objective evolutionary algorithms (MOEAs) have been gaining increasing attention among researchers and practitioners mainly because of the fact that they can be suitably applied to find multiple Pareto-optimal solutions in one single simulation run. This fact alone enables a user to have a less-subjective search in the first phase of finding a set of well-distributed solutions. Instead of choosing a weight vector emphasizing one objective over the other in a subjective manner, the goal in an MOEA is to first find a set of well-distributed solutions close to the true Pareto-optimal front. Once these solutions are found, in the next phase, some higher-level problem information can be used to select one solution. This two-step procedure of multi-objective problem solving has been discussed in detail in (Deb, 2001). By applying MOEAs to different search and optimization problems, researchers have demonstrated that such a procedure is more pragmatic and efficient compared to the preference-based classical approaches. This procedure is practical because the user gets an opportunity to investigate a number of other trade-off solutions before choosing one particular optimal solution (see (Deb, 2001; Coello et al., 2002; Zitzler et al., 2001a)). The MOEA search procedure is also algorithmically efficient because the discovery of one solution close to the Pareto-optimal front will *pull* a number of other population members towards the Pareto-optimal front, thereby making a parallel and simultaneous discovery of multiple trade-off solutions. This feature has attracted numerous researchers to develop different MOEAs (NSGA-II (Deb et al., 2002a), SPEA (Zitzler and Thiele, 1999), SPEA2 (Zitzler et al., 2001b), PAES (Knowles and Corne, 2000), PESA (Corne et al., 2000), and others). Computer codes (or pseudo-codes) of many of these MOEAs are also available on the Internet (`http://www.iitk.ac.in/pub.htm` for NSGA-II and `http://www.tik.ee.ethz.ch/pisa` for SPEA2 and others.)

It is clear from the existing studies that there are two distinct goals in the development of an MOEA: (i) convergence to the true Pareto-optimal front and (ii) maintenance of a well-distributed set of non-dominated solutions. Although a third goal of achieving both the above tasks in a computationally fast manner is also an important matter, this goal has often been ignored in most studies in the past. It is now well-established that the computation of a well-diversified set of Pareto-optimal solutions is usually time-consuming (Deb et al., 2002b; Laumanns et al., 2002). MOEAs can be clearly classified into two distinct groups. Some MOEAs use a quick-and-dirty diversity preservation operator, thereby finding a reasonably good distribution quickly, whereas some other MOEAs use a more computationally expensive diversity preservation operator in order to obtain a better distribution of solutions. For example, the modified non-dominated sorting genetic algorithm (NSGA-II) (Deb et al., 2002a) uses a crowding approach for diversity preservation that requires a computational complexity of $O(N \log N)$, where N is the population size. On the other hand, the strength Pareto evolutionary algorithm (SPEA) (Zitzler and Thiele, 1999) uses a clustering approach involving Euclidean distance computations that requires a computational complexity of $O(N^3)$. Although in two-objective optimization problems, the diversity obtained by these two MOEAs was reported to be almost similar (Deb et al., 2002a), a remarkably large difference was evident in solving three or more objective problems (Deb et al., 2002b; Khare et al., 2003) . The SPEA produced a much better distribution at the expense of a large computational effort.

Recently, the authors proposed a steady-state MOEA (Deb et al., 2003) based on the ϵ-dominance concept (Laumanns et al., 2002) and demonstrated its working principle with a few proof-of-principle results. In this paper, we address the tripartite task of convergence, maintenance of diversity, and computational efficiency desired in an

MOEA further and extensively evaluate the ϵ-MOEA procedure on standard test problems (having up to four objectives) and compared with four commonly-used MOEAs. The ϵ-dominance does not allow two solutions with a difference less than ϵ_i in the i-th objective to be non-dominated to each other, thereby allowing a good diversity to be maintained in a population. Besides, the method is quite pragmatic because it allows the user to choose a suitable ϵ_i depending on the desired resolution in the i-th objective. In the ϵ-MOEA, two populations (EA and archive) are evolved simultaneously and independently. Using one solution each from both populations, two offspring solutions are created. Each offspring is then used to update both parent and archive populations. The archive population is updated based on the ϵ-dominance concept, whereas a usual domination concept is used to update the parent population. Since the ϵ-dominance concept reduces the cardinality of the Pareto-optimal set and since a steady-state EA is suggested, the maintenance of a diverse set of solutions is possible with a small computational time.

In the remainder of the paper, we briefly discuss a clustered version of NSGA-II and then present the ϵ-MOEA approach in detail. Thereafter, these two MOEAs are compared along with the original NSGA-II and a couple of other state-of-the-art MOEAs – SPEA2 (Zitzler et al., 2001b) and the Pareto envelope based selection algorithm or PESA (Corne et al., 2000) – on a number of two, three and four objective test problems. Finally, we detail a number of important conclusions about the performance of each MOEA which result from the study.

2 A Good Distribution Versus a Quick Computation Time

Besides the convergence to the Pareto-optimal front, one of the equally important goals of multi-objective optimization is to find and maintain a widely distributed set of solutions. Since the Pareto-optimal front can be a convex, non-convex, disconnected, or piece-wise continuous hyper-surface, there are differences of opinion about defining a diversity measure denoting the true spread of a finite set of solutions on or close to the Pareto-optimal front. Although the task is easier for a two objective space, the difficulty arises in the case of higher-dimensional objective spaces. This is the reason why researchers have developed different diversity measures, such as the hyper-volume measure (Zitzler and Thiele, 1999), the spread measure (Schott, 1995), the chi-square deviation measure (Srinivas and Deb, 1994), the R-measures (Hansen and Jaskiewicz, 1998), and others. In maintaining diversity among population (or archive) members, several researchers have used different diversity-preserving operators, such as clustering (Zitzler and Thiele, 1999), crowding (Deb et al., 2002a), pre-specified archiving (Knowles and Corne, 2000), and others. Interestingly, these diversity-preserving operators produce a trade-off between the achievable diversity and the computational time.

The clustering approach of SPEA forms N clusters (where N is the archive size) from $N'(> N)$ population members by initially assuming each of N' members to be a separate cluster. Thereafter, all $\binom{N'}{2}$ Euclidean distances in the objective space are computed. Then, the two clusters with the smallest distance are merged together to form one bigger cluster. This process reduces the number of clusters to $N' - 1$. The inter-cluster distances are computed again[1] and another merging is done. This process is repeated until the number of clusters is reduced to N. With multiple population members occupying two clusters, the average distance of all pair-wise distances between solutions of the two clusters is used. Figure 1 illustrates this procedure. For the two

[1]A special book-keeping procedure can be used to eliminate further computation of pair-wise Euclidean

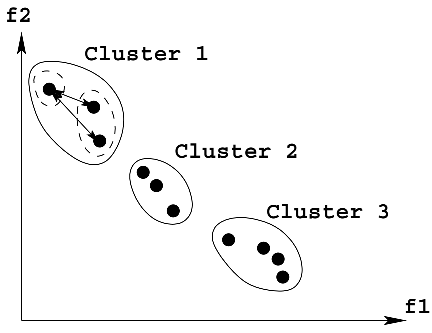

Figure 1: The clustering approach used in SPEA.

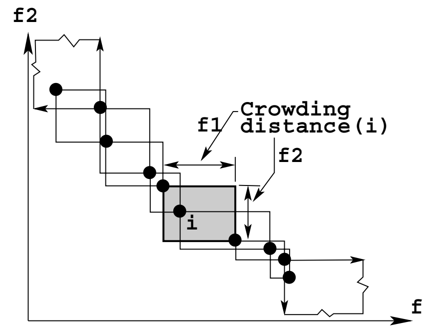

Figure 2: The crowding approach used in NSGA-II.

clusters shown in dashed lines, the average Euclidean distance among the solutions of two clusters is computed as shown in the figure. The average distance is computed for all pairs of clusters and the two clusters with the smallest average distance are merged together, as shown in the figure. If $(N' - N)$ is of the order of N (the archive size), then the procedure requires $O(N^3)$ computations in each iteration. Since this procedure is repeated in every iteration of SPEA, the computational overhead as well as the storage requirements for implementing the clustering concept are large. However, since the clustering is implemented based on the Euclidean distance among solutions, the resulting distribution of clustered solutions is usually good.

On the other hand, NSGA-II uses a crowding operator, in which N' (as large as $2N$, where N is the population size) solutions are processed objective-wise. In each objective direction, the solutions are first sorted in the ascending order of objective value. Thereafter, for each solution an objective-wise *crowding distance* is assigned equal to the difference between normalized objective value of the neighboring solutions. Figure 2 shows the hyper-boxes used to calculate the crowding distance of each solution on a non-dominated front. The overall crowding distance is equal to the sum of the crowding distances from all objectives (computed as the half of the perimeter of the enclosing hyper-box). Once all distance computations are achieved, the solutions are sorted in the descending order of crowding distance and the first N solutions are chosen. This procedure requires $O(N \log N)$ computations. Although the objective-wise distance computation in NSGA-II makes the algorithm computationally faster, the diversity in solutions achievable by NSGA-II is not expected to be as good as that achievable with SPEA.

3 Two Approaches for a Better Spread

Here, we discuss in detail a simple modification to the NSGA-II procedure and an ϵ-dominance based MOEA introduced by the authors recently (Deb et al., 2003) for the main purpose of achieving a better distribution of Pareto-optimal solutions.

3.1 Clustered NSGA-II (C-NSGA-II)

The first approach is a straightforward replacement of NSGA-II's crowding method by the clustering one as used in SPEA. After the parent and offspring population are

distances.

combined into a bigger population of size $2N$ and this combined population is sorted into different non-domination levels, only N good solutions are required to be chosen based on their non-domination levels and *nearness* to each other (Deb et al., 2002a). This procedure is illustrated in Figure 3. For the scenario depicted in the figure, the

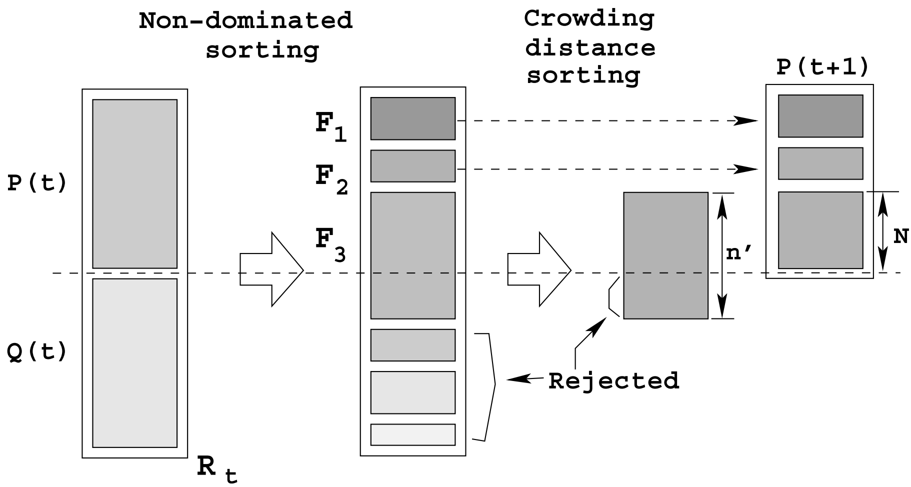

Figure 3: NSGA-II procedure.

first two non-dominated fronts are directly copied to the new population $P(t+1)$, while all members of the third non-dominated front cannot be copied, due to lack of available population slots. The required number of solutions (N') from the third front are chosen in order to have the maximum diversity (in terms of the crowding distance) among the chosen solutions. This is where the original NSGA-II uses a computationally effective crowding procedure, described in the previous section (Figure 2). In the clustered NSGA-II approach, we replace the crowding procedure with the clustering approach described in Figure 1. In this procedure, the solutions in the last permissible non-dominated level (the third front for the scenario in Figure 3) are considered for the clustering procedure. Let us suppose that the number of population slots remaining to be filled is N' and the solutions in the last permissible non-domination level from the combined population is n'. By definition, $n' \geq N'$. To choose N' solutions from n', we form N' clusters from n' solutions and choose one representative solution from each cluster. Ideally the solutions marked with a box in Figure 3 can be chosen as a representative solution of each box. In this procedure, the extreme solution of each extreme cluster can be chosen and the solution close to the center of a cluster can be chosen from intermediate clusters. The clustering algorithm used in this study is similar to that used in SPEA (Zitzler and Thiele, 1999). Although this requires a larger computational time, the clustered NSGA-II is expected to find a better distributed set of Pareto-optimal solutions than the original NSGA-II, but at the expense of a larger computational overhead.

3.2 A Steady-State ϵ-MOEA

The above algorithm is a generational evolutionary algorithm in which all N population members (offspring) are created before comparing them with parent solutions. In the context of single-objective EAs, it has been adequately shown that computational speed can be achieved by using a steady-state EA in which every offspring is compared with the parent population as soon as it is created (Goldberg and Deb, 1991). This way, the parent population gets updated in a steady-state manner, thereby providing better chances of creating good offspring solutions. Unfortunately, there are not many steady-state MOEAs in the literature and authors have proposed an ϵ-MOEA recently (Deb et al., 2003). Here, we elaborate this method and later evaluate its efficiency by comparing it with a number of state-of-the-art MOEAs. The ϵ-MOEA procedure is presented below:

Step 1 `Randomly initialize a population` $P(0)$. `The non-dominated solutions of` $P(0)$ `are copied to an archive population` $E(0)$. `Set the iteration counter` $t=0$.

Step 2 `One solution` p `is chosen from the population` $P(t)$ `using the pop_selection procedure.`

Step 3 `One solution` e `is chosen from the archive population` $A(t)$ `using an archive_selection procedure.`

Step 4 `One offspring solutions` c `is created using` p `and` e.

Step 5 `Solution` c `is included in` $P(t)$ `using a pop_acceptance procedure.`

Step 6 `Solution` c `is included in` $A(t)$ `using a archive_acceptance procedure.`

Step 7 `If termination criterion is not satisfied, set` $t=t+1$ `and go to Step 2, else report` $A(t)$.

It is clear from the above algorithm that the ϵ-MOEA uses two co-evolving populations: an EA population $P(t)$ and an archive population $A(t)$ (where t is the iteration counter). Figure 4 illustrates the algorithm. The MOEA begins with an initial population $P(0)$. The archive population $E(0)$ is assigned with the ϵ-non-dominated solutions of $P(0)$. Thereafter, two solutions, one each from $P(t)$ and $A(t)$ are chosen for mating and an offspring solution c is created. Thereafter, the solution c can enter each of the two populations with different strategies. Here, we discuss the selection and acceptance procedures which can be used in the ϵ-MOEA:

Pop_selection procedure To choose a solution from $P(t)$, two population members from $P(t)$ are picked up at random and a domination check (in the 'usual' sense, shown on left in Figure 4 for minimization of objectives) is made. If one solution dominates the other, the former is chosen. Otherwise, the event indicates that these two solutions are non-dominated to each other and in such a case we simply

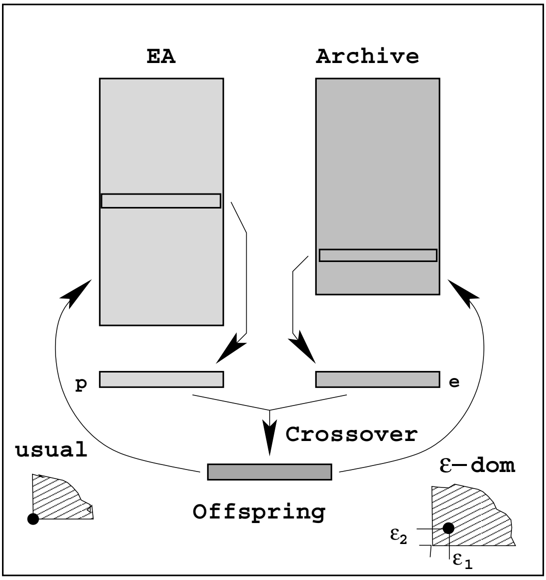

Figure 4: ϵ-MOEA procedure. In showing the dominated regions, the minimization of an objective function is assumed.

choose one of them at random.

Archive_selection procedure To choose a solution e from $A(t)$, several strategies involving a certain relationship with the chosen p can be made. Here, we randomly pick a solution from $A(t)$.

Pop_acceptance procedure The decision whether the offspring c will replace any population member can be made using different strategies. Here, we compare the offspring with all population members. If the offspring dominates one or more population members, then the offspring replaces one of them (chosen at random). On the other hand, if any population member dominates the offspring, it is not accepted. When both the above tests fail (that is, the offspring is non-dominated to the population members), the offspring replaces a randomly chosen population member, thereby ensuring that the EA population size remains unchanged.

Archive_acceptance procedure For the offspring c to be included in the archive population, the offspring is compared with each member of the archive using ϵ-dominance criterion (Laumanns et al., 2002). We describe the procedure in the following paragraph.

Every solution in the archive is assigned an identification array ($\mathbf{B} = (B_1, B_2, \ldots, B_M)^T$, where M is the total number of objectives) as follows:

$$B_j(\mathbf{f}) = \begin{cases} \lfloor (f_j - f_j^{\min})/\epsilon_j \rfloor, & \text{for minimizing } f_j, \\ \lceil (f_j - f_j^{\min})/\epsilon_j \rceil, & \text{for maximizing } f_j. \end{cases} \tag{1}$$

where $f_j^{\min}$ is the minimum possible value of the j-th objective and ϵ_j is the allowable tolerance in the j-th objective below which two values are insignificant to the user.

This ϵ_j value is the same as the ϵ used in the ϵ-dominance definition (Laumanns et al., 2002). The identification array divides the whole objective space into hyper-boxes, each having ϵ_j size in the j-th objective. Figure 5 illustrates that the solution P ϵ-dominates the entire region ABCDA (in the minimization sense), whereas the original dominance definition allows P to dominate only the region PECFP. For brevity, the rest of the discussion is confined to minimization cases alone. However, a similar analysis can be followed for maximization or mixed cases as well. The identification array of P is the coordinates of point A in the objective space. With the identification arrays calculated for the offspring c_i and each archive member a, we use the following procedure (see Figure 6 for an illustration of each of the four cases which may appear): If the identification array $\mathbf{B}_a$ of any archive member a dominates that of the offspring c_i, then it means that the offspring is ϵ-dominated by this archive member and so the offspring is not accepted. This case is illustrated in Figure 6(a). On the other hand, if $\mathbf{B}_{c_i}$ of the offspring dominates the $\mathbf{B}_a$ of any archive member a, the archive member is deleted and the offspring is accepted (Figure 6(b)). If neither of the above two cases occur, then it means that the offspring is ϵ-non-dominated with the archive members. We separate this case into two. If the offspring shares the same $\mathbf{B}$ vector with an archive member (meaning that they belong to the same hyper-box), then they are first checked for the usual non-domination. If the offspring dominates the archive member or the offspring is non-dominated to the archive member but is closer to the $\mathbf{B}$ vector (in terms of the Euclidean distance) than the archive member, then the offspring is retained. These two cases are illustrated in Figure 6(c). Solutions 1 and 2 in Figure 5 also illustrate the latter case. These two solutions occupy the same hyper-box (or have the same $\mathbf{B}$ vector) and they are non-dominated according to the usual definition. Since solution 1 has a smaller distance to the $\mathbf{B}$ vector, it is retained and solution 2 is deleted. In the event of an offspring not sharing the same $\mathbf{B}$ vector with any archive member, the offspring is accepted. This is illustrated in Figure 6(d). It is interesting to note that the former condition ensures that only one solution with a distinct $\mathbf{B}$ vector would exist in each hyper-box. This means that each hyper-box on the Pareto-optimal front can be occupied by only one solution, thereby providing two properties: (i) well-distributed solutions can be maintained and (ii) the final archive size amounting to the total number of Pareto-optimal solutions will be bounded. For this reason, no specific upper limit on the archive size needs to be pre-fixed. The archive will get bounded according to the

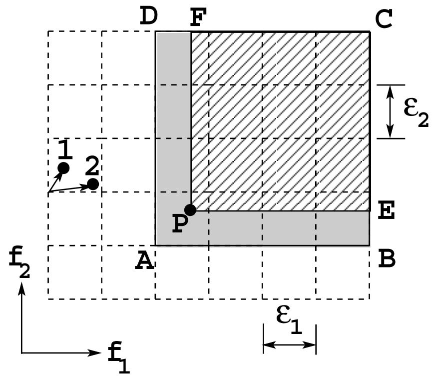

Figure 5: The ϵ-dominance concept (for minimizing f_1 and f_2).

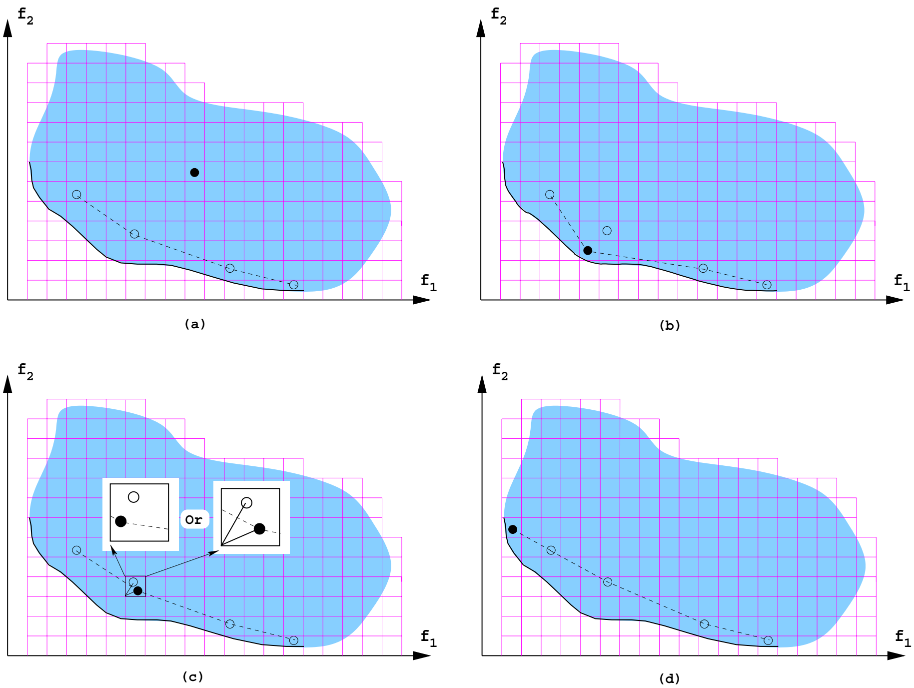

Figure 6: Four cases of accepting a child in the archive are illustrated.

chosen ϵ-vector.

The above procedure is continued for a specified number of iterations and the final archive members are reported as the obtained solutions. A careful observation will reveal the following properties of the ϵ-MOEA procedure:

1. It is a steady-state MOEA.
2. It emphasizes non-dominated solutions.
3. It maintains the diversity in the archive by allowing only one solution to be present in each pre-assigned hyper-box on the Pareto-optimal front.
4. It is an elitist approach.

A C code implementing the ϵ-MOEA procedure is available at `http://www.iitk.ac.in/kangal/soft.htm`. The objective space is divided into a number of grids (or hyper-boxes) and the diversity is maintained by ensuring that a grid or hyper-box can only be occupied by one solution. Although, PAES and its variants (Knowles and Corne, 2000) are developed with the similar idea, we discuss in the following section that ϵ-MOEA is a more general concept.

3.3 Differences with Other Grid-Like MOEAs

Although the above steady-state ϵ-MOEA may look similar to the multi-parent PAES (Knowles and Corne, 2000), there are some differences. The PAES also divides the entire objective space into a number of hyper-boxes. In a steady-state approach, each offspring is compared with a continuously updated archive population for its inclusion. In the event of the offspring being non-dominated with the archive population, it is compared with the hyper-box having the maximum number of solutions in it. If the offspring resides in a less crowded hyper-box, it is accepted and a member from the maximally-crowded hyper-box is deleted at random. The ϵ-dominance concept implemented here with the $\mathbf{B}$-vector domination-check does not allow two non-dominated solutions with a difference less than ϵ_i in the i-th objective to be both present in the final archive. On the other hand, PAES allows more than one member to be present in each hyper-box. Figure 5 can be used to show that fewer non-dominated solutions will be obtained with the ϵ-MOEA approach than PAES. The procedure will not only allow a reduction in the size of the final Pareto-optimal set, but it also has a practical significance. Since a user is not interested in obtaining solutions with a difference less than ϵ_i in the i-th objective, the above procedure allows the user to find solutions according to his/her desire. Even though the actual number of solutions to be obtained by the ϵ-MOEA procedure is unknown, it is bounded. Because of this reason, the overall computational time is expected to be smaller. For the same reason, for a fixed population size, ϵ-MOEA is also likely to find a better spread of solutions than PAES.

The archive update strategy is also similar to that in another study (Laumanns et al., 2002), except in the case when two solutions have the same $\mathbf{B}$ vector. Here, a solution non-dominated to an existing archive member but sharing a common hyper-box can still be chosen if it is closer to its $\mathbf{B}$ vector. The earlier study only accepted an offspring if it dominated the existing member. Here we have presented a steady-state MOEA procedure with an EA population update strategy, an archive update strategy, and a sound recombination plan.

4 Simulation Results

The original study (Deb et al., 2003) performed a limited simulation study comparing ϵ-MOEA with only NSGA-II and the clustered NSGA-II and performing only one simulation run from a particular initial population and using only a few two and three-objective test problems. In this section, we investigate the performance of ϵ-MOEA in more elaborate fashion:

1. The ϵ-MOEA is compared with SPEA2, PESA, NSGA-II, and clustered NSGA-II. These algorithms are commonly-used and have demonstrated their abilities in converging and maintaining diversity of Pareto-optimal solutions.

2. The ϵ-MOEA is applied on 12 test problems (five two-objective ZDT problems, six three-objective DTLZ problems, and one four-objective DTLZ problem).

3. Three performance metrics are used to measure the progress in convergence, maintenance of diversity and computational time.

4. In each case, mean and standard deviation of each of the three performance metrics are compared among all five MOEAs.

SPEA2 (Zitzler et al., 2001b) is an advanced and modified version of SPEA (Zitzler and Thiele, 1999). The SPEA2 uses an improved fitness assignment scheme and a somewhat faster k-th nearest neighbor approach for maintaining diverse solutions, instead of the clustering procedure used in SPEA. In the SPEA2 used here, we keep the sizes for the EA and archive populations the same. We have assigned this size identical to the NSGA-II population size (N). As per the suggestion of the developers, for the nearest neighbor approach, we have set $k = \sqrt{2N}$.

The PESA is a population-version of PAES, in which emphasis is given to solutions residing in a less-crowded hyper-box in both the selection and the offspring-acceptance operators. As suggested in (Corne et al., 2000), we have used an EA population size of 10 and an archive population size of 100 in all the problems discussed in this study. The PESA requires the user to set a hyper-box size parameter. Based on some trail-and-error experiments, we have set 32×32 hyper-boxes for two-objective problems, $6 \times 6 \times 6$ for three-objective problems, and $12 \times 12 \times 12 \times 12$ for the four-objective problem.

The details of the test problems can be found elsewhere (Deb, 1999; Deb et al., 2002b). DTLZ8 is a constrained optimization problem which is also included in the set of test problems. The problems involve as large as 30 decision variables. Since all test problems involve real-valued decision variables, we have used the SBX recombination operator (Deb and Agrawal, 1995) and the polynomial mutation operator (Deb and Goyal, 1996) to create an offspring solution. Recent studies (Ziztler et al., 2003; Knowles and Corne, 2002) suggest the use of binary performance indicators for an appropriate evaluation of two sets of non-dominated data. Although we acknowledge the importance of using binary performance indicators, due to computational complexities involved in them, in this study we use two unary performance indicators (distance from the true Pareto-optimal set and a sparsity measure), each independently indicating the quality of solutions for two different goals of multi-objective optimization: (i) extent of convergence and (ii) extent of sparsity. We also use the hyper-volume measure (Zitzler and Thiele, 1999) for finding a combined convergence and diversity estimate for some test problems. Since all the problems considered in this paper are test problems, the exact knowledge of the Pareto-optimal front is available beforehand. For the convergence metric, we calculate H uniformly distributed (on the f_1-f_2-$\cdots$-f_{M-1}-plane) solutions (the set P^*) on the Pareto-optimal front. For each such point in the $(M-1)$-dimensional plane, f_M is calculated from the known Pareto-optimal front description. Then, the Euclidean distance of each obtained solution from the nearest solution in P^* is computed. The average of the distance value of all obtained solutions is defined as the convergence measure here. The sparsity metric calculation is a somewhat involved and is described in detail while discussing the three-objective optimization results. For a fair comparison, each algorithm is run for a fixed number of solution evaluations. Importantly, we also present the computational time needed to run each MOEA on the same computer (a 1.7 GHz Pentium IV processor).

4.1 Two-Objective Test Problems

First, we consider five two-objective ZDT test problems (Deb, 1999; Zitzler et al., 2000), each providing a different kind of difficulty for MOEAs.

4.1.1 ZDT1 Test Problem

The 30-variable ($n = 30$) ZDT1 problem has a convex Pareto-optimal front. We use a population size of $N = 100$ and the real-parameter SBX recombination operator with $p_c = 1$ and $\eta_c = 15$, and a polynomial mutation operator with $p_m = 1/n$ and $\eta_m = 20$ (Deb, 2001). In order to investigate the effect of ϵ ($\epsilon_1 = \epsilon_2 = \epsilon$ is assumed here) used in the ϵ-MOEA, we use different ϵ values and count the number of solutions found in the archive after 20,000 solution evaluations in each case. Figure 7 shows that as ϵ increases, the number of obtained solutions varies almost proportional to $1/\epsilon$. For an equally-sloped Pareto-optimal straight line in the range $f_1 \in [0,1]$, we would expect $1/\epsilon$

solutions to appear in the archive. But in the case of a non-linear Pareto-optimal front, one would expect a smaller number of solutions in the archive. This will be clear from Figure 8, which shows the distribution of solutions obtained with $\epsilon = 0.05$. The boxes within which a solution lies are also shown in the figure. It is interesting to note that all solutions are ϵ-non-dominated with respect to each other and in each of the expected boxes only one solution is obtained. In the boxes with $f_1 \in [0, 0.05]$, one Pareto-optimal

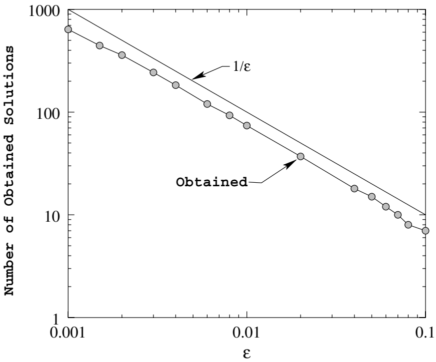

Figure 7: The number of solutions versus ϵ on ZDT1.

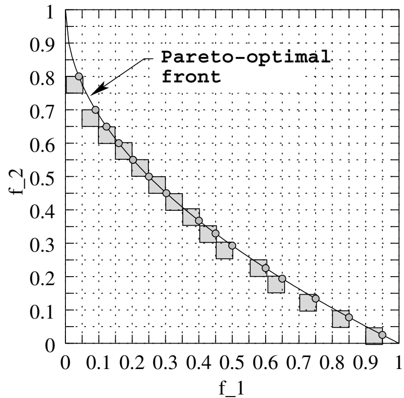

Figure 8: ϵ-MOEA distribution on ZDT1 with $\epsilon = 0.05$.

solution in the minimum f_2 box ($f_2 \in [0.75, 0.80]$) is obtained. The four boxes on top of this box (with larger f_2) are ϵ-dominated and hence not retained in the final archive.

To compare the five MOEAs, we use the convergence metric discussed above, the hyper-volume metric (Zitzler and Thiele, 1999), and a sparsity metric. We use $\epsilon_i = 0.0075$ in order to get roughly 100 solutions in the archive after 20,000 solution evaluations. The convergence metric is computed with $H = 1,000$ equi-spaced solutions on the Pareto-optimal front. Table 1 shows that the convergence of solutions is best achieved with ϵ-MOEA. For the hyper-volume measure, we use the reference point at $(1.1, 1.1)^T$.

Although the convergence of the ϵ-MOEA is better than that of others in this problem, the hyper-volume measure of ϵ-MOEA is not found to be so good. This is mainly due to the absence of extreme solutions on the Pareto-optimal front (refer Figure 8). Fixing a ϵ-vector fixes a unique combination of hyper-boxes in which a Pareto-optimal solution is expected. Figure 8 marks that particular unique combination of boxes for the ZDT1 problem with $\epsilon_1 = \epsilon_2 = 0.05$. Because of the ϵ-dominance concept, the extreme solutions usually get dominated by ones that are both within ϵ_i distance in some objective and comparatively better in some other. The presence or absence of the extreme solutions makes a significant difference in the hyper-volume measure, a matter we discuss later in more detail. Thus, the hyper-volume measure may not be an ideal metric for measuring diversity and convergence together for a set of solutions. We suggest and use a different diversity measure, called the sparsity measure (which we have described in Section 4.3 in detail for any number of objectives). In short, the sparsity measure first projects the obtained solutions on a suitable hyper-plane (a normal unit vector $\vec{\eta} = (1/\sqrt{2}, 1/\sqrt{2})^T$ for ZDT1 is used here) and then computes the non-overlapping area occupied by the solutions on that plane. The higher the metric value,

Table 1: Performance comparison of the five MOEAs for ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 with a limit of 20,000 function evaluations. Best metric values are shown bold. Average of 10 runs is indicated by 'Avg.' and standard deviation is indicated by 'SD'.

	Convergence measure		Sparsity		Hyper-volume		Time (sec)	
MOEA	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
ZDT1								
NSGA-II	0.00054898	6.62e-05	0.858	0.0202	0.8701	3.85e-04	18.29	0.51
C-NSGA-II	0.00061173	7.86e-05	0.994	0.0043	**0.8713**	2.25e-04	1911.18	98.10
PESA	0.00053481	12.62e-05	0.754	0.0331	0.8680	6.76e-04	11.40	0.54
SPEA2	0.00100589	12.06e-05	**0.999**	0.0014	0.8708	1.86e-04	595.94	25.02
ϵ-MOEA	**0.00039545**	1.22e-05	0.991	0.0050	0.8702	8.25e-05	**1.11**	0.03
ZDT2								
NSGA-II	**0.00037851**	1.88e-05	0.855	0.0250	0.5372	3.01e-04	18.63	0.51
C-NSGA-II	0.00040011	1.91e-05	0.994	0.0015	0.5374	4.42e-04	1953.30	123.67
PESA	0.00037942	2.95e-05	0.759	0.0202	0.5329	11.25e-04	11.35	0.62
SPEA2	0.00082852	11.38e-05	**0.999**	0.0015	0.5374	2.61e-04	539.51	17.84
ϵ-MOEA	0.00046448	2.47e-05	0.994	0.0037	**0.5383**	6.39e-05	**1.52**	0.02
ZDT3								
NSGA-II	0.00232321	13.95e-05	0.887	0.0675	1.3285	1.27e-04	20.86	0.89
C-NSGA-II	0.00239445	12.30e-05	0.991	0.0083	1.3277	9.82e-04	1421.89	89.20
PESA	0.00211373	15.38e-05	0.882	0.0492	1.2901	7.49e-03	18.27	1.39
SPEA2	0.00260542	15.46e-05	**0.996**	0.0023	1.3276	2.54e-04	438.92	14.77
ϵ-MOEA	**0.00175135**	7.45e-05	0.986	0.0055	**1.3287**	1.31e-04	**1.09**	0.02
ZDT4								
NSGA-II	0.00639002	0.0043	0.958	0.0328	**0.8613**	0.00640	11.21	1.09
C-NSGA-II	0.00618386	0.0744	**0.998**	0.0029	0.8558	0.00301	124.60	45.19
PESA	0.00730242	0.0047	0.798	0.0352	0.8566	0.00710	6.56	0.57
SPEA2	0.00769278	0.0043	0.989	0.0132	0.8609	0.00536	111.96	35.62
ϵ-MOEA	**0.00259063**	0.0006	0.987	0.0076	0.8509	0.01537	**0.59**	0.04
ZDT6								
NSGA-II	0.07896111	0.0067	0.815	0.0157	0.3959	0.00894	10.19	0.31
C-NSGA-II	0.07940667	0.0110	0.995	0.0029	0.3990	0.01154	2916.23	382.76
PESA	0.06415652	0.0073	0.748	0.0345	0.4145	0.00990	10.57	0.28
SPEA2	**0.00573584**	0.0009	**0.998**	0.0029	**0.4968**	0.00117	319.67	29.89
ϵ-MOEA	0.06792800	0.0118	0.996	0.0023	0.4112	0.01573	**0.82**	0.01

the better is the distribution. This measure is also normalized so that the maximum possible non-overlapping area is 1.000, indicating a complete non-overlap among projected solutions. The measure also involves a size parameter, which is adjusted in such a way that one of the competing MOEAs achieve a near 100% non-overlapping area. For ZDT1, SPEA2 achieves the best average sparsity measure of 0.999 (over five runs), while those of C-NSGA-II (0.994) and ϵ-MOEA (0.991) are close behind.

The table also presents the average of actual computational time taken by each MOEA over five different runs. It is interesting to note that the ϵ-MOEA obtains the best convergence and a good distribution of solutions in a computational time at least an order of magnitude smaller than its nearest competitor. With respect to the clustered MOEAs (SPEA2 and C-NSGA-II) the computational time needed in the ϵ-MOEA is three to four orders of magnitudes less. Although the PESA obtains the next-best computational time, the diversity obtained in the solutions is the poorest.

Another interesting aspect to note is that the standard deviation obtained for all three performance metrics among all 10 independent simulation runs is small compared to the average metric values for all algorithms. This indicates that the chosen parameter setting for each algorithm is adequate for the algorithms to demonstrate their *robustness* to different initial populations.

4.1.2 ZDT2 Test Problem

This problem tests an MOEA's ability to find non-convex Pareto-optimal solutions (Deb, 1999). All five MOEAs are applied to this problem having 30 decision variables. An $\epsilon_i = 0.0076$ is used in the ϵ-MOEA to find roughly 100 solutions in the final archive. The rest of the parameters are the same as those used in ZDT1. Table 1 shows the metric values representing the performance of the competing MOEAs. The NSGA-II performs the best in terms of convergence. Although the SPEA2 is the best in terms of the sparsity measure, it experiences difficulty in convergence. However, like in ZDT1, the ϵ-MOEA demonstrates good convergence and sparsity measures in a very short computational time. Again, the algorithms demonstrate adequate robustness to variation to the initial population.

4.1.3 ZDT3 Test Problem

This problem provides difficulties by introducing discontinuities in the Pareto-optimal front (Deb, 1999). Here too, 30 decision variables are used. To obtain roughly 100 Pareto-optimal solutions, $\epsilon_i = 0.00261$ is used. Other parameters are kept the same as before. The table shows that the ϵ-MOEA achieves the best convergence and takes the least computational time. In terms of preserving diversity, the SPEA2 performs the best, with C-NSGA-II and ϵ-MOEA providing comparable values.

4.1.4 ZDT4 Test Problem

Table 1 also shows the performance measures on the 10-variable ZDT4 problem (Deb, 1999). This problem has a number of local Pareto-optimal fronts, thereby providing hurdles for an MOEA to converge to the global Pareto-optimal front. All parameters used here are identical to that used in ZDT1, except that $\epsilon_i = 0.0058$ is used to get around 100 solutions in the final archive (after 20,000 solution evaluations). The table shows that the ϵ-MOEA is better than the other four MOEAs in terms of convergence and computational time, and is also good in terms of diversity among obtained solutions.

We ignore solving the ZDT5 test problem, as this problem is defined for binary strings. We restrict ourselves to solving real-parameter problems in this paper and highlight that the ϵ-MOEA can also be applied to binary-coded problems by simply changing the recombination and mutation operators.

4.1.5 ZDT6 Test Problem

The 10-variable ZDT6 problem has a non-uniform density of solutions across the Pareto-optimal front (Deb, 1999). Here, we use $\epsilon_i = 0.0067$ to get about 100 solutions in the archive after 20,000 solution evaluations. SPEA2 performs the best in both sparsity and convergence, but at the expense of a large computational time. The ϵ-MOEA is superior to all other MOEAs in terms of the computational time and also has a reasonably good convergence and diversity measures.

Thus, from the two-objective problems studied above, we can conclude that the ϵ-MOEA produces a good convergence and diversity with a smaller (by at least an order of magnitude) computational time than the other four state-of-the-art MOEAs. SPEA2 and C-NSGA-II both give very good diversity, closely followed by ϵ-MOEA.

The disadvantage with SPEA2 and C-NSGA-II lies in the amount of computation time (three or four orders of magnitude more than that needed by the ϵ-MOEA) they demand. The PESA performs poorly in achieving the adequate diversity in all problems. However, the ϵ-MOEA provides a very good compromise between convergence, diversity, and computational time. Table 1 also indicates that ϵ-MOEA produces a small standard deviation value for all metrics, thereby indicating that ϵ-MOEA is a reliable and robust algorithm for multi-objective optimization.

4.2 Effect of Limiting Evaluations

In the above experiments, we have presented the solutions obtained after 20,000 function evaluations. Here, we rerun some of the experiments for 10,000 function evaluations to investigate if any algorithm had a better convergence property. We keep the remaining parameters identical and show three performance measures for three competing algorithms for 10,000 function evaluations in Table 2. It is clear from the table that in all problems ϵ-MOEA and NSGA-II converge much faster near to the true Pareto-optimal front than SPEA2. Moreover, both ϵ-MOEA and NSGA-II converges with a better diversity. However, in all cases ϵ-MOEA is able to come to close to the final front in about 10,000 evaluations only. By comparing Tables 1 and 2, it can be seen that both SPEA2 and NSGA-II are far away from the true front in the case of ZDT4 and in ZDT6, SPEA2 is away from the true front. In general, SPEA2 makes a slower convergence compared to ϵ-MOEA and NSGA-II. Thus, in addition to maintaining a good spread, ϵ-MOEA is also better in terms the rate of convergence to the true Pareto-optimal front.

Table 2: Performance comparison of the three MOEAs for ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 run up to 10,000 function evaluations.

	Convergence measure		Sparsity		Hyper-volume	
MOEA	Avg.	SD	Avg.	SD	Avg.	SD
ZDT1						
NSGA-II	0.00064963	6.37e-05	0.959	0.0103	**0.8705**	2.13e-04
SPEA2	0.00967400	1.75e-03	1.000	0.0000	0.8568	2.64e-03
ϵ-MOEA	**0.00040322**	1.79e-05	**1.000**	0.0000	0.8703	6.41e-05
ZDT2						
NSGA-II	**0.00041808**	2.25e-04	0.899	0.2459	0.4518	0.1801
SPEA2	0.01192230	5.48e-03	0.802	0.4167	0.4305	0.1704
ϵ-MOEA	0.00047377	2.87e-05	**1.000**	0.0000	**0.5379**	7.59e-05
ZDT3						
NSGA-II	0.00235714	1.73e-04	0.954	0.0378	1.3120	0.0351
SPEA2	0.00531642	6.29e-04	0.997	0.0019	1.3064	0.0042
ϵ-MOEA	**0.00191443**	7.43e-05	**1.000**	0.0000	**1.3142**	0.0262
ZDT4						
NSGA-II	0.37601900	0.1605	0.949	0.0375	0.3207	0.1992
SPEA2	0.30709100	0.2101	0.585	0.3544	0.2454	0.3475
ϵ-MOEA	**0.01536600**	0.0389	**1.000**	0.0000	**0.8176**	0.0629
ZDT6						
NSGA-II	0.08236520	0.0036	0.902	0.0328	0.3940	0.0043
SPEA2	0.21286500	0.0377	0.788	0.0400	0.2357	0.0463
ϵ-MOEA	**0.07294320**	0.0073	**1.000**	0.0000	**0.4040**	0.0098

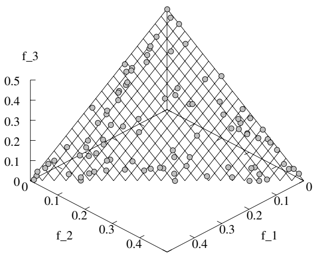

Figure 9: NSGA-II distribution on DTLZ1.

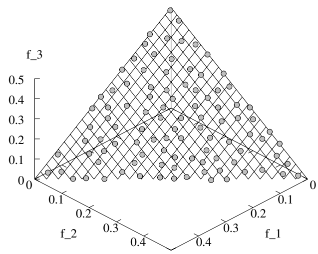

Figure 10: C-NSGA-II distribution on DTLZ1.

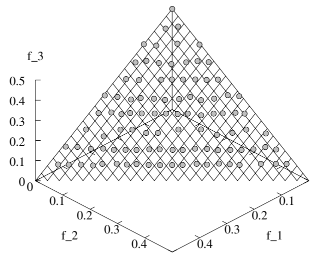

Figure 11: ϵ-MOEA distribution on DTLZ1.

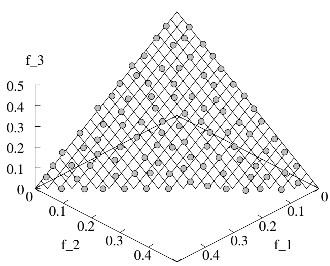

Figure 12: SPEA2 distribution on DTLZ1.

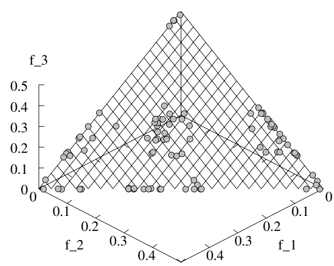

Figure 13: PESA distribution on DTLZ1.

4.3 Three-Objective Test problems

Now, we consider six, three-objective test problems developed elsewhere (Deb et al., 2002b).

4.3.1 DTLZ1 Test Problem

First, we consider the three-objective DTLZ1 test problem with $n = 7$ variables. The Pareto-optimal solutions lie on a three-dimensional plane satisfying: $f_1+f_2+f_3 = 0.5$ in the range $f_1, f_2, f_3 \in [0, 0.5]$. For each algorithm, we use a population size of 100, real-parameter representation with the SBX recombination (with $\eta_c = 15$ and $p_c = 1$), the polynomial mutation operator (with $\eta_m = 20$ and $p_m = 1/n$) (Deb, 2001), and a maximum of 30,000 function evaluations. For ϵ-MOEA, we have chosen $\epsilon = [\frac{5}{24}, \frac{5}{24}, \frac{5}{10}]^T$. The solutions obtained in each case are shown on the objective space in Figures 9 to 13 for different MOEAs.

It is clear from the figures that the distribution of solutions with the original NSGA-II and PESA is poor compared to the other three MOEAs. Using $H = 5,000$ solutions, we tabulate the average convergence measure for five independent runs in column 2 of Table 3. It is observed that the convergence of the ϵ-MOEA is relatively better than that of the other four MOEAs with identical function evaluations.

Column 4 of the table shows the hyper-volume measure (averaged over five runs) calculated with respect to the reference solution at $f_1 = f_2 = f_3 = 0.7$. Since a large hyper-volume is better for minimization problems, the table indicates that the SPEA2 is the best, followed by the NSGA-II. But Figure 9 and Table 3 show that the NSGA-II is neither better in sparsity nor in convergence than ϵ-MOEA. The hyper-volume

Table 3: Comparison of five MOEAs in terms of their convergence and diversity measures on DTLZ1.

MOEA	Convg. measure		Sparsity		Hyper-volume		Time (sec)	
	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
NSGA-II	0.00351	0.00067	0.84111	0.03570	0.31322	5.893e-04	41.07	10.640
C-NSGA-II	0.00346	0.01583	0.99289	0.00777	0.31308	14.119e-04	1098.84	275.077
PESA	0.00360	0.00290	0.74156	0.03338	0.28248	33.242e-04	12.22	0.837
SPEA2	0.00333	0.03541	**0.99978**	0.00050	**0.31598**	6.977e-04	549.88	37.008
ϵ-MOEA	**0.00329**	0.00092	0.981326	0.01815	0.30088	15.020e-04	**1.55**	0.133

metric fails to capture this aspect, despite a better convergence and sparsity of ϵ-MOEA solutions. This is due to the fact that the extreme solutions contribute a lot to the hyper-volume metric. Since the ϵ-dominance does not allow two solutions with a difference of ϵ_i in the i-th objective to be mutually non-dominated to each other, it will not be usually possible to obtain the extreme corners of the Pareto-optimal front. However, the diversity of solutions elsewhere on the Pareto-optimal front is ensured by the archive update procedure of ϵ-MOEA. It is then easy to see that without the extreme solutions the hyper-volume measure fails to add the large portion of hyper-volume metric which is contributed by the extreme solutions. The advantage in having a well-distributed set of solutions on the interior of the Pareto-optimal front was not enough to account for the loss in diversity due to the absence of a few boundary solutions. We argue that in this sense the hyper-volume metric is biased towards the boundary solutions. Thus, we do not use this measure for the rest of the test problems used in this paper.

Instead, we define a *sparsity* measure, which is similar to the entropy measure (Farhang-Mehr and Azarm, 2002) or the grid diversity measure (Deb and Jain, 2002) introduced elsewhere. The Pareto-optimal solutions are first projected on a suitable hyper-plane (with a unit normal vector $\vec{\eta}$). Figure 14 illustrates the calculation procedure for this measure. A hyper-box of a certain size d is centered around each projected solution. The total hyper-volume covered by these hyper-boxes is used as the measure of sparsity of solutions. If a solution set has many clustered points, then their hyper-boxes will overlap with each other and the obtained sparsity measure will be small. On the other hand, if the solutions are well distributed, the hyper-boxes will not overlap and a large overall measure will be obtained. To normalize the measure, we divide the total hyper-volume by the total expected hyper-volume calculated with a same-sized solution set having no overlap between the hyper-boxes. Thus, the maximum sparsity achievable is 1.000 and the larger the sparsity measure, the better is the distribution. However, the choice of the parameter d is important here. A too small value of d will make any distribution have the maximum sparsity measure of 1.000, whereas a very large value of d will make every distribution have a small sparsity measure. We solve this difficulty in choosing a suitable d value by finding the smallest possible value that will make one of the competing distributions achieve the maximum sparsity value of 1.000. This value of d is used for computing the sparsity for other distributions. In all case studies discussed here, points are projected on a plane equi-slopped to the coordinate axes ($\vec{\eta} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$). The third column of Table 3 shows that the SPEA2 attains the best distribution, followed by C-NSGA-II, and then ϵ-MOEA. The distribution of solutions, as can be seen in Figures 9 to 11, also support these calculations.

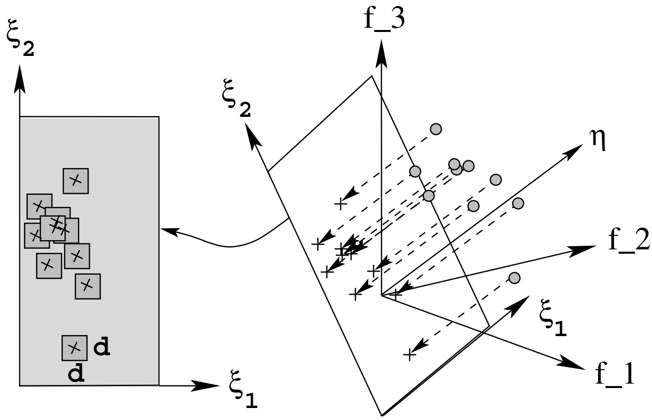

Figure 14: The sparsity measure is illustrated.

Although the C-NSGA-II and SPEA2 achieve better distributions, they are also computationally the slowest among the five MOEAs. Since the clustering in C-NSGA-II and SPEA2 algorithms require comparison of each population member with every other for computing adequate number of clusters in every generation of C-NSGA-II, the time taken is at least three orders of magnitude more than that of ϵ-MOEA. Similarly, the extensive calculations needed for the truncation operator of SPEA2 also causes it to slow down. A visual comparison between Figures 11 and 12 indicates that the distribution of solutions obtained with ϵ-MOEA is not bad and comparable to that obtained with SPEA2. Based on the convergence measure, the diversity measure, and the elapsed computational time, we can conclude that the ϵ-MOEA emerges as a good compromised algorithm.

4.3.2 DTLZ2 Test Problem

Next, we consider the 12-variable DTLZ2 test problem with a spherical Pareto-optimal front satisfying $f_1^2 + f_2^2 + f_3^2 = 1$ in the range $f_1, f_2, f_3 \in [0, 1]$. Identical parameters to those used in DTLZ1 are used here. A total of $H = 8,000$ Pareto-optimal solutions are considered as P^* for the convergence metric computation. For the ϵ-MOEA, we have used $\epsilon = [0.06, 0.06, 0.066]^T$. This produces about 100 solutions on the Pareto-optimal front. Table 4 shows the comparison of performance measures of the five MOEAs after 30,000 function evaluations. Figures 15 to 18 show the distribution of solutions

Table 4: Performance comparison of the five MOEAs for DTLZ2.

	Convergence measure		Sparsity		Time (sec)	
MOEA	Avg.	SD	Avg.	SD	Avg.	SD
NSGA-II	0.0137186	0.0020145	0.931111	0.0124474	17.16	0.196
C-NSGA-II	0.0107455	0.0008424	**0.999778**	0.0004968	7837.42	81.254
PESA	**0.0106292**	0.0025483	0.945778	0.0309657	88.01	12.901
SPEA2	0.0126622	0.0009540	0.998889	0.0007855	2164.42	19.858
ϵ-MOEA	0.0108443	0.0002823	0.999104	0.0009316	**2.01**	0.032

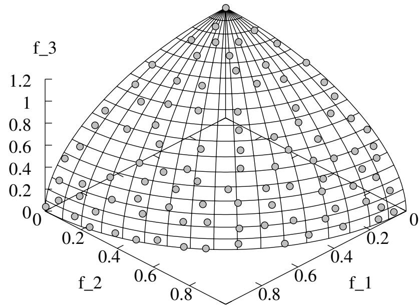

Figure 15: C-NSGA-II distribution on DTLZ2.

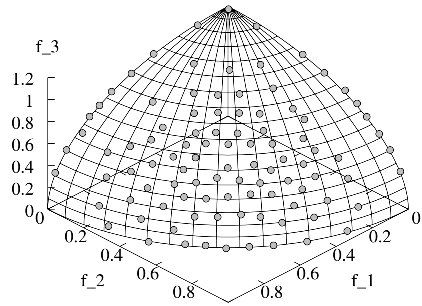

Figure 16: ϵ-MOEA distribution on DTLZ2.

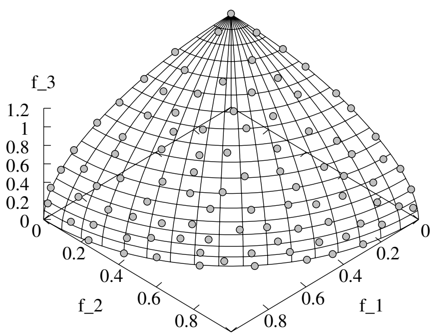

Figure 17: SPEA2 distribution on DTLZ2.

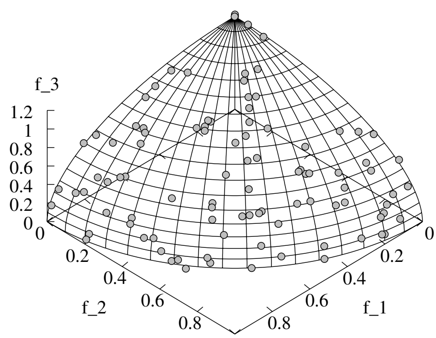

Figure 18: PESA distribution on DTLZ2.

obtained by all the other MOEAs except NSGA-II. Although, C-NSGA-II, SPEA2 and ϵ-MOEA produced a very similar sparsity measure (Table 4), they were each produced in different ways. Since, the Euclidean distance is used in the clustering approach of C-NSGA-II and SPEA2 for maintaining diversity, a uniform spread of solutions on the front is observed. In the case of ϵ-MOEA, there seems to be a considerable gap between the boundary solutions and their nearest neighbors. This happens because of the fact that there is a gentle slope near the boundary solutions on a spherical surface and the ϵ-dominance consideration does not allow any solution to be non-dominated within an ϵ_i in the i-th objective. But, wherever there is a considerable change of slope, more crowded solutions are found. It was argued above that such a set of solutions with a minimum pre-specified difference in objective values has a practical significance and hence we advocate the use of ϵ-MOEA in this paper. It is clear from Table 4 that the ϵ-MOEA achieves a good convergence and diversity measure with a much less computational time than C-NSGA-II and SPEA2.

4.3.3 DTLZ3 Test Problem

Here, we concentrate on the 12-variable DTLZ3 test problem, which has a large number of local Pareto-optimal fronts and one global front (Deb et al., 2002b). The global front satisfies the equation $f_1^2 + f_2^2 + f_3^2 = 1$. Because of the presence of a large number of local fronts, parallel to the global front, a population size of $N = 200$ is used in all the MOEAs. In the case of the ϵ-MOEA, we use $\epsilon = [0.042, 0.0425, 0.04]^T$ in order to obtain approximately 200 solutions on the Pareto-front. The total number of function evaluations used is 100,000. Table 5 shows the performance metrics of the five MOEAs.

Table 5: Performance comparison of the five MOEAs for DTLZ3.

MOEA	Convergence measure		Sparsity		Time (sec)	
	Avg.	SD	Avg.	SD	Avg.	SD
NSGA-II	0.0149156	0.01028	0.839228	0.02961	136.45	31.080
C-NSGA-II	0.0202315	0.00898	0.995521	0.00613	24046.03	4690.032
PESA	0.0130633	0.00449	0.722296	0.02785	89.49	12.527
SPEA2	0.0122429	0.00194	**0.999771**	0.00031	9080.81	963.723
ϵ-MOEA	**0.0122190**	0.00223	0.993207	0.00974	**9.42**	2.180

Once again, we find that C-NSGA-II, SPEA2 and ϵ-MOEA achieve very similar values of the sparsity measure. As before, the ϵ-MOEA turns out to be the best compromise among all the MOEAs considered in this study.

4.3.4 DTLZ4 Test Problem

The 12-variable DTLZ4 test problem introduces a non-uniform density of solution on the three-objective Pareto-optimal front. In this problem, a uniform distribution of Pareto-optimal solutions is difficult to obtain. The global optimal front for this problem is the curve represented by: $f_1^2 + f_2^2 + f_3^2 = 1$ with $f_1, f_2, f_3 \in [0, 1]$. Since this problem has a greater density of solutions near the f_3-f_1 and f_1-f_2 planes, certain runs of all the MOEAs produced solutions only on these planes. So, for this particular problem, ten different runs were taken instead of five. For the evaluation of the sparsity measure, only those runs were considered that gave solutions throughout the Pareto-front and not just on any of the above mentioned planes. Column 2 of Table 6 shows the number of runs (the more, the better) which produced a distribution of solutions on the entire Pareto-optimal front. The performance metric values of the five MOEAs after 30,000 function evaluations are shown in Table 6. For ϵ-MOEA, we have used

Table 6: Performance comparison of the five MOEAs for DTLZ4.

MOEA	# success (of 10)	Convergence measure		Sparsity		Time (sec)	
		Avg.	SD	Avg.	SD	Avg.	SD
NSGA-II	5	0.0125015	0.00132	0.928666	0.00947	16.99	0.180
C-NSGA-II	8	0.0113708	0.00087	**0.998611**	0.00114	8753.12	866.452
PESA	3	0.0100756	0.00040	0.857037	0.04895	31.18	4.342
SPEA2	5	0.0163104	0.01186	0.997942	0.00151	881.57	23.477
ϵ-MOEA	6	**0.00977548**	0.00020	0.997759	0.00301	**2.35**	0.016

$\epsilon = [0.07, 0.07, 0.03]^T$ to obtain 100 solutions on the Pareto-optimal front. Although the C-NSGA-II has the maximum success in this problem in terms of maintaining a good spread of solutions, from all the performance comparisons, the ϵ-MOEA emerges as a good compromise.

4.3.5 DTLZ5 Test Problem

The DTLZ5 is a three-objective, 12-variable problem with a Pareto-optimal curve: $f_1^2 + f_2^2 + f_3^2 = 1$ with $f_1, f_2, f_3 \in [0, 1]$. This problem tests an MOEA's ability to find a lower-dimensional Pareto-optimal front, while working with a higher-dimensional objective space (Deb et al., 2002b). Table 7 shows the performance measures. Here, we use

Table 7: Performance comparison of the five MOEAs for DTLZ5.

	Convergence measure		Sparsity		Time (sec)	
MOEA	Avg.	SD	Avg.	SD	Avg.	SD
NSGA-II	0.00208342	11.976e-05	0.953778	0.00992	11.49	0.036
C-NSGA-II	0.00256138	30.905e-05	0.996667	0.00314	1689.16	81.365
PESA	0.00094626	11.427e-05	0.772110	0.02269	53.27	11.836
SPEA2	0.00197846	16.437e-05	**1.000000**	0.00000	633.60	14.082
ϵ-MOEA	**0.000953623**	4.892e-05	0.980867	0.01279	**1.45**	0.051

$\epsilon_i = 0.005$ for the ϵ-MOEA. These results are obtained after 20,000 function evaluations. It is also clear from the table that the ϵ-MOEA is the quickest and the best in terms of achieving convergence. SPEA2 performs best in maintaining diversity, closely followed by ϵ-MOEA and C-NSGA-II.

For brevity, we do not show the simulation results for DTLZ6, DTLZ7, and DTLZ9 here. Instead, we attempt the DTLZ8 test problem, which is presumably difficult to solve for finding a good spread of solutions using classical optimization methods.

4.3.6 DTLZ8 Test Problem

Here, we consider the three-objective, 30-variable DTLZ8 test problem (Deb et al., 2002b). The overall Pareto-optimal region is a combination of two fronts: (i) a line and (ii) a plane. This problem involves three inequality constraints, which are handled using the constraint-domination principle suggested elsewhere (Deb, 2001) for all the algorithms. Figure 19 shows the distribution of points obtained using the SPEA2 after 100,000 evaluations. In this test problem, as discussed elsewhere (Deb et al., 2002b), the domination-based MOEAs suffer from what is known as the 'redundancy problem'. For two distinct solutions on the line portion of the Pareto-optimal front, many other non-Pareto-optimal solutions appear as non-dominated. In Figure 19, the redundant solutions are those that are on the adjoining sides (shown shaded) of the Pareto-optimal line. Further, note that the SPEA2 is also unable to get rid of these solutions, as these are non-dominated to some Pareto-optimal solutions. But with ϵ-MOEA, many of these redundant solutions get ϵ-dominated by the Pareto-optimal solutions. Figure 20 shows the solutions obtained with ϵ-MOEA having $\epsilon = [0.02, 0.02, 0.04]^T$. With a 30-variable decision space, the density of solutions near the Pareto-optimal line and close to the $f_3 = 0$ plane on the Pareto-optimal front are very small. Thus, it may be, in general, difficult to find solutions on these portions of the Pareto-optimal front. For the ϵ-MOEA, we have used $\eta_c = 2$ and $\eta_m = 5$. However, we used $\eta_c = 15$ and $\eta_m = 20$ for SPEA2 as they produced better results. It is clear from the plot that the ϵ-MOEA is able to find a reasonable distribution of solutions on the line and the plane. Although ϵ-MOEA is able to eliminate most of the redundant solutions, some of them still remain. However, the number of such solutions is much smaller than that obtained in a procedure that uses the original dominance criterion. We do not show the distribution of solutions obtained using other MOEAs, as they produce distributions worse than SPEA2.

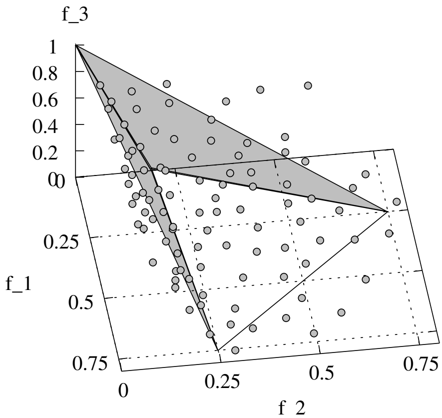

Figure 19: SPEA2 distribution on DTLZ8.

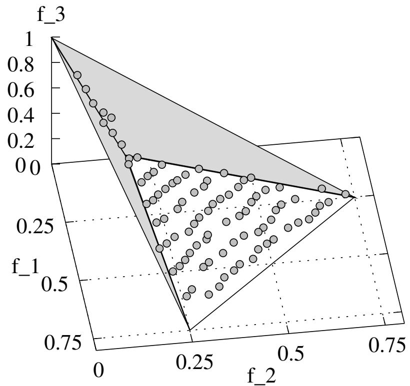

Figure 20: ϵ-MOEA distribution on DTLZ8.

4.4 A Four-Objective Test Problem

To have a look at the performance of the five MOEAs on a four-objective test problem, we apply all the MOEAs to the 13-variable DTLZ2 test problem. This problem has a 'spherical' Pareto-front in four dimensions given by the equation : $f_1^2 + f_2^2 + f_3^2 + f_4^2 = 1$ with $f_i \in [0, 1]$ for $i = 1$ to 4. The performance of the five MOEAs after 30,000 function evaluations are shown in Table 8. For the ϵ-MOEA, $\epsilon = [\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}]^T$ was used to obtain

Table 8: Performance comparison of the five MOEAs for the four-objective DTLZ2 test problem.

	Convergence measure		Sparsity		Time (sec)	
MOEA	Avg.	SD	Avg.	SD	Avg.	SD
NSGA-II	0.1775980	0.04816	0.808138	0.02779	19.27	1.165
C-NSGA-II	0.0274868	0.00597	0.968640	0.00684	9579.40	816.043
PESA	**0.0252412**	0.00543	0.852032	0.05248	283.83	20.656
SPEA2	0.0654494	0.01303	**0.998816**	0.00119	3003.94	100.727
ϵ-MOEA	0.0396261	0.00201	0.995556	0.00532	**6.98**	0.483

around $N = 100$ solutions in the final archive. All the other parameters in this case were kept the same as those used in the two objective DTLZ2 test problem (discussed in Section 4.3.2). For the sparsity measure, we have used a $\vec{\eta}$ which is equally inclined to all objective axes. Although PESA performs the best in terms of convergence, both ϵ-MOEA and C-NSGA-II produce very similar values. The convergence of SPEA2 and NSGA-II are comparatively worse. The diversity of solutions obtained using SPEA2 is the best, followed by ϵ-MOEA and C-NSGA-II. The NSGA-II performs the worst both in terms of convergence and diversity in this problem. Although NSGA-II is reportedly shown to perform quite well for two or three-objective problems, this study demonstrates that its quick-and-dirty crowding operator is not adequate in maintaining a good distribution of solutions in a problem having many objectives. On the other hand, the C-NSGA-II and SPEA2 finds a good distribution of solutions but at the expense of very large computation times, Even in this problem, it is seen that the ϵ-MOEA emerges out to be a balanced algorithm, producing very good convergence and diversity with a very small computational effort.

5 Conclusions

It has always been a dream of a multi-objective EA researcher and practitioner to be able to find a well-distributed set of solutions as close to the true Pareto-optimal front as possible with as small a computational time as possible. Although past studies have either demonstrated a good distribution with a large computational overhead or a not-so-good distribution quickly, a recent effort by the authors showed a way to arrive at a good compromised procedure quickly. In this paper, we elaborate that procedure and evaluate it extensively by considering more test problems and by comparing it with two other state-of-the-art methodologies to realize that dream into practicality. The tripartite task of achieving convergence, maintaining diversity and requiring a quick computational time have been all achieved by using careful strategies in choosing mating partners from two co-evolving populations and in accepting the created offspring to each population. The use of ϵ-dominance criterion has been found to have two advantages: (i) it has helped in reducing the cardinality of Pareto-optimal region and (ii) it has ensured that no two obtained solutions are within an ϵ_i from each other in the i-th objective. The first aspect is useful in using the ϵ-MOEA to higher-objective problems and to somewhat lessen the 'redundancy' problem (Deb et al., 2002b) inherent to dominance based algorithms applied to many-objective problems. The second aspect also makes the approach highly pragmatic, particularly in making the MOEA approach interactive with a decision-maker.

On all 12 test problems, the ϵ-MOEA has been successful in finding well-converged and well-distributed solutions with a much smaller computational effort than a number of state-of-the-art MOEAs including NSGA-II, SPEA2, and PESA. The consistency in achieving convergence and diversity of solutions over multiple simulation runs and the requirement of only a fraction of computational effort needed compared to other MOEAs suggest the use of the ϵ-MOEA to more complex and real-world problems.

The ϵ-MOEA procedure can be extended by using different other procedures for selecting mating solutions and accepting an offspring solution. To obtain a better spread, the suggestions of other ϵ-dominance criteria outlined elsewhere (Laumanns et al., 2002) can be implemented to the ϵ-MOEA. To obtain different range of diversity in each objective, a different ϵ for each objective can be chosen. In the event of finding a biased distribution of solutions around a particular region in the Pareto-optimal front, the ϵ_i can be chosen as a function of f_i. Although some such extensions are logical future research steps, this extensive study with ϵ-MOEA has amply shown its superiority over the currently used MOEAs and should be immediately applied to real-world multi-objective optimization problems.

References

Coello, C. A. C., VanVeldhuizen, D. A., and Lamont, G. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer Academic Publishers.

Corne, D., Knowles, J., and Oates, M. (2000). The Pareto envelope-based selection algorithm for multiobjective optimization. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature VI (PPSN-VI)*, pages 839–848.

Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal*, 7(3):205–230.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley.

Deb, K. and Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2002a). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K. and Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26(4):30–45.

Deb, K. and Jain, S. (2002). Running performance metrics for evolutionary multi-objective optimization. In *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL-02)*, pages 13–20.

Deb, K., Mohan, M., and Mishra, S. (2003). Towards a quick computation of well-spread pareto-optimal solutions. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632)*, pages 222–236.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002b). Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pages 825–830.

Farhang-Mehr, A. and Azarm, S. (2002). Diversity assessment of pareto-optimal solution sets: An entropy approach. In *Proceedings of the World Congress on Computational Intelligence*, pages 723–728.

Goldberg, D. E. and Deb, K. (1991). A comparison of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms 1 (FOGA-1)*, pages 69–93.

Hansen, M. P. and Jaskiewicz, A. (1998). Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Lyngby: Institute of Mathematical Modelling, Technical University of Denmark.

Khare, V., Yao, X., and Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632)*, pages 376–390.

Knowles, J. D. and Corne, D. (2002). One metrics for comparing nondominated sets. In *Congress on Evolutionary Computation (CEC-2002)*, pages 711–716. Piscataway, NJ: IEEE Press.

Knowles, J. D. and Corne, D. W. (2000). Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation Journal*, 8(2):149–172.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282.

Schott, J. R. (1995). Fault tolerant design using single and multi-criteria genetic algorithms. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Srinivas, N. and Deb, K. (1994). Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal*, 2(3):221–248.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2):125–148.

Zitzler, E., Deb, K., Thiele, L., Coello, C. A. C., and Corne, D. (2001a). *Proceedings of the First Evolutionary Multi-Criterion Optimization (EMO-01) Conference (Lecture Notes in Computer Science (LNCS) 1993)*. Heidelberg: Springer.

Zitzler, E., Laumanns, M., and Thiele, L. (2001b). SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In Giannakoglou, K. C., Tsahalis, D. T., Périaux, J., Papailiou, K. D., and Fogarty, T., editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece. International Center for Numerical Methods in Engineering (Cmine).

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Ziztler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.