IMPROVED MAXIMIN SELECTION FOR WELL SPREAD PARETO FRONTS

S. I. Valdez Peña, S. Botello Rionda and A. Hernández Aguirre

Comunicación Técnica No I-07-02/16-02-2007
(CC/CIMAT)

Improved MaxiMin selection for well spread Pareto fronts

S. I. Valdez Peña, S. Botello Rionda, and A. Hernández Aguirre
Center for Research in Mathematics (CIMAT),
Department of Computer Science,
A.P. 402, Guanajuato, Gto. 36000, México
ivvan, botello, artha@cimat.mx

February 15, 2007

Abstract

An algorithm to achieve maximal spread and almost perfectly distributed Pareto fronts is presented. The MaxiMin algorithm add points to the archive of selected individuals one by one, each point which is added maximizes the distance from the current selected points. This method is independent of the evolutionary operators used to perform the search. This work explains how to combine the MaxiMin selection with state of the art multi-objective algorithms, such as NSGA-II, SPEA2, and DEMO. Experiments were ran with and without MaxiMin for comparison purposes. ε-MOEA is used as reference. Performance metrics and graphical results are shown for comparison.

1 Introduction

Multi-objective optimization algorithms (MOEAs) find optimum solutions in a space shared by several conflicting functions. The optimality measure used by most approaches is the Pareto optimality criterion and the set of vectors in agreement with it is the Pareto set, or non-dominated set. The hyper-surface shaped by this set, when projected to function space, is called the Pareto front. Usually, the Pareto front has an infinite number of points. Computationally, it is impossible to work with such a large number of points. Therefore, MOEAs usually work with a bounded archive to preserve the best solutions found. In order to accomplish the desired features, two goals have been established for MOEAs [1]:

1. To find a set of solutions as close as possible to the Pareto-optimal front.
2. To find a set of solutions as diverse as possible.

Several mechanisms have been proposed in order to accomplish the second goal. The main idea of these mechanisms involves complementing the optimality criteria with a crowding measure. Some proposals are: the niche count used by NPGA [2], the hypercube perimeter used by NSGA-II [3], the $k-th$ nearest neighbor based distance in SPEA2 [4], the n-dimensional grid used by PAES in function space [5], the nearest neighbor average proposed in PDE [6], and the ε-dominance used by ε-MOEA. The DEMO algorithm [7] uses the same diversity preservation mechanism used by NSGA-II, but for some problems it reports better diversity metric values. In this context, observe that most of the mechanism implemented by MOEAs to preserve diversity are independent of the evolutionary search engine. A brief reminder of the most common and well performed mechanisms is presented in Section 2.

Section 3 presents an improved selection strategy called MaxiMin to select well distributed points on the Pareto front with no detriment of the convergence capacity. This mechanism was studied before by Solteiro et al. in [8], as a sorting strategy to preserve diverse solutions. By using spreading metrics, preliminary results about the MaxiMin capacity were shown in [8]. Discussion about the metrics used by Solteiro et al., and the performance metrics used in this paper are presented in Section 5.

Since the MaxiMin does not depend on the search engine, several proposals are presented to insert the MaxiMin mechanism in some of the most useful and well performed MOEAs, such as NSGA-II, SPEA2 and DEMO. Finally it is compared to the original implementation of these algorithms by using diversity and convergence performance metrics. Comparisons with the ε-MOEA are also performed. This algorithm was especially designed for achieving well distributed solutions; thus, this comparison is particularly interesting. The MaxiMin algorithm, however, shows some dependency to population size. The analysis is presented in Section **??**, and conclusions are given in Section 7.

2 Overview of diversity mechanism approaches

As noted, diversity is one of the most important issues in multi-objective evolutionary optimization. Since MOEAs work with a finite population size, solutions which represent the whole optimum space are required. Also, in most cases, maintaining diverse solutions is a good strategy to improve performance. Diversity helps individuals to escape from local minima, avoiding premature convergence to a fake optimum set. In addition, the optimum set found by a MOEA can be used for a non-automated decision maker, who would choose a unique solution. In order to generate and to preserve diverse solutions, several mechanisms have been proposed. The discussion below briefly reviews the diversity strategies of the three algorithms that will be enhanced with MaxiMin selection and used later for the experiments. The ε-MOEA algorithm rests on the concept of ε-dominance to achieve diversity. It is described for the sake of completeness but it is only used as a reference. However, it is important to mention that diversity control strategies are part of several algorithms, for instance the Niched Pareto Genetic Algorithm (NPGA), proposed by Horn et al. [2], the Pareto Archive Evolution Strategy, proposed by Knowles and Corne [5], and the Pareto-frontier Differential Evolution, proposed by Abbas et al. [6].

2.1 The Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)

The NSGA-II [3] uses a Pareto ranking as a partial fitness assignment. Each solution is ranked by the Pareto front it belongs to. Then, to preserve diverse solutions, a hyper-cube perimeter measure is used to calculate a crowding distance (this step completes the fitness assignment). There are two steps where the crowding distance is used: first in the crowded tournament selection, and for truncation of the best solutions when the archive is overfilled. Once the function evaluation is computed, each solution is ranked with the front it belongs to. Solutions from the same front are used to assign the crowding distance. First, the solutions with the same rank are sorted by their objective function values, for each objective function. Then, a crowding distance is computed as follows:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{max} - f_m^{min}} \tag{1}$$

where $m = 1, 2...M$ is the objective function index. I^m are sorted indexes by the m objective function. A large distance is assigned to the boundary solutions, $d_{I_1^m} = d_{I_l^m} = \infty$. For all other solutions $j = 2\ to\ (l-1)$, the crowding distance is calculated as in Equation 1, l is the number of solutions in the current rank. In tournament selection, the crowding distance is used when candidates do not dominate each other. For the truncation method, the best ranked individuals are preserved, crowding distance is used only to discriminate individuals with the same rank.

2.2 The Improved Strength Pareto Evolutionary Algorithm (SPEA2)

In this method a fitness assignment is performed by calculating a strength value, which is the sum of how many individuals each individual dominates [4]. The sum of the strength values of individuals which dominates an individual x is the fitness value of x. To preserve diverse solutions, a nearest neighbor distance function is used as crowding distance, it is calculated as follows:

$$d_i = \frac{1}{2 + \sigma_j^k} \tag{2}$$

where σ_j^k is the distance to the $k-th$ nearest neighbor. This distance is used in tournament selection to determine a winner when candidates have the same fitness. In the truncation procedure to maintain a fixed size archive, dominated solutions with the worst crowding values are removed first, if the number of non-dominated individuals is great than the fixed archive size, individuals with the worst crowding distance are removed too.

2.3 Differential Evolution for Multi-objective Optimization (DEMO)

The algorithm proposed by Robic and Filipic [7], uses the differential evolution crossover operator [9]. It does not perform any tournament for parent selection, the parents are picked randomly. When a child dominates his father, the last one is deleted from the archive, otherwise the father is preserved. All children are added to the archive, then a truncation procedure is applied when there are more non-dominated solutions than those required. Using the crowding distance (and ranking) from the NSGA-II [3], the DEMO algorithm truncates solutions. Even though the same fitness and crowding measure are used in NSGA-II, the DEMO algorithm reports a better diversity [7].

2.4 ε-MOEA

This algorithm was proposed by Deb et al. [10]. It uses the ε-*dominance* criterion presented by Laumans et al. in [11]. Basically, the objective space is split into a grid determined by an ε vector given (the number of hyper-boxes is given by ε). Two evolving populations are used: P that is a usual population randomly initialized and E that is an archive population with the best solutions found. One offspring is generated from two solutions mated, one from P and one from E. The solution from P is chosen by a tournament selection using usual dominance to determine the winner, if both competitors are non-dominated, one of them is chosen randomly. The solution from E is simply randomly picked. Then, solutions in the archive and the new candidate solution (offspring) are assigned an identification array $B = (B_1, B_2, ...B_M)^T$, where M is the number of objective functions. This vector is computed as follows:

$$B_j(f) = \begin{cases} \lfloor (f_j - f_{min})/\varepsilon_j \rfloor, & \text{for minimizing } f_j, \\ \lceil (f_j - f_{min})/\varepsilon_j \rceil, & \text{for maximizing } f_j, \end{cases} \tag{3}$$

Using B, the candidate solution is compared with each solution in the archive E. If the candidate dominates any solution, the last is deleted and the candidate is added to the archive. Otherwise, if any solution has the same B values as the candidate solution, it is compared using usual dominance, and the winner is added to the archive. If the candidate solution and a solution from E have the same B values, and are non-dominated in the usual sense, the solution with the smallest distance to B is the winner. In the case that no solution in the archive has the same values as the candidate, and the candidate is non-dominated, it is accepted to the archive.

3 The MaxiMin Selection

The algorithm presented in this section is based on a distance measure over selected points. A significant difference among this approach and some presented before (see [3], [4], [6], [2]), is that MaxiMin works with distances over the selected individuals, in order to get a good spreading. Other approaches usually calculate a crowding measure among the archive and the evolving population. These measures consider all the individuals, without considering if they will be selected or not. These kinds of measures do not ensure a good spread of solutions, because eliminating clustered or crowded points from the merged population is not to choose the best uniformly distributed points in the selected individuals. The presented approach adds points one by one, checking that each point which is added keeps a good spreading in the selected set. As we work with selected points, calculation of distances among all the points is not necessary, we only need to calculate the distance among selected points and remaining points in the merged population.

According to pseudo-code in Figure 2, in line 1 the values of the objective functions are normalized between 0 and 1, using the maximum and minimum known values of every function. $F(S)$ are the function values of individuals S to

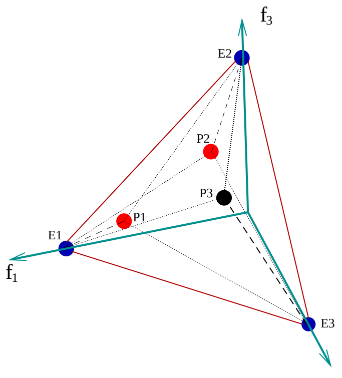

Figure 1: Graphical explanation for the MaxiMin algorithm.

which the MaxiMin will be applied.

In line 2, extremes of the Pareto front are selected using reference points: $[1,0,0,\ldots,0]$, $[0,1,0,\ldots,0]$, ..., $[0,0,0,\ldots,1]$. The extreme points are the nearest to each reference point. So, extreme individuals are selected from S to be inserted in D. Solteiro et al. proposed in [8] using the maximum or minimum in every objective function. We propose using the extreme points as are defined before, considering that in some problems there can be many (infinite) minima/maxima and all of them can be non-dominated individuals. An example of this type of Pareto front is the DTLZ1 problem proposed by Deb et al. in [12], there is a set of points which are minima for every function. Actually, the minima for f_1 is the line created where the front cuts the plane $f_2 - f_3$. We have a similar case for f_2 and f_3, minimum is not unique. Using reference points we can approach a better idea of the front limits and shape, and most of the time extreme points are unique. There are as many extreme points as objective functions.

In lines 3 to 9, a distance measure attached to each remaining individuals in the set $S \setminus D$ is initialized, calculating the (minimum) distance from the set $S \setminus D$ to D. This measure is stored in d.

Function in line 10 finds the index with the maximum distance measure, this point is added to D in line 11.

Lines 12 to 24 are used to calculate distances from set $S \setminus D$ to set D, and adds the point with the maximum distance measure to D. A point is added each loop, and the distance measures are updated if necessary.The remaining points are added one by one until the whole fixed size archive is filled.

A graphical explanation of the algorithm for three objectives is shown in Figure 1. The extreme points are E1, E2, and E3. The lines trace distances among the selected points (initially the extreme points) and the remaining points. The dashed lines are distances which are taken as a measure for the remaining points (minD(P_i)). Then, the point with the maximum measurement is $P3$ which will be the next selected point.

4 Enhancing MOEAs with the MaxiMin Selection

The MaxiMin algorithm can be used with any fitness assignment, and any search engine. One must keep in mind that the goal of MaxiMin is to select uniformly spaced vectors to populate the next generation. In the following we describe three examples in which NSGA-II, SPEA2, and DEMO are modified to work with the MaxiMin selection.

normalize($F(S)$);
Select_extreme($F(S)$,D)
For $i \in S \setminus D$
    $d_i = M$
    For $l \in D$
        if $\left(d_i > \sqrt{\sum_k^M (f_{k,i} - f_{k,l})^2} \right)$
            $d_i = \sqrt{\sum_k^M (f_{k,i} - f_{k,l})^2}$
    EndFor
EndFor
m=***find_max(d)***
$D = D \cup x_m$
While $|D| <$ *archive size*
    $d_{max} = 0$
    For $i \in S \setminus D$
        if $\left(d_i > \sqrt{\sum_k^M (f_{k,i} - f_{k,m})^2} \right)$
            $d_i = \sqrt{\sum_k^M (f_{k,i} - f_{k,m})^2}$
        if $(d_i \geq d_m ax)$
            $d_{max} = d_i$
            $m_{temp} = i$
        EndIf
    EndFor
    $m = m_{temp}$
    $D = D \cup x_m$
End While

Figure 2: Pseudo-code for the MaxiMin algorithm.

4.1 Enhancing the NSGA-II with the MaxiMin selection.

Suppose we have μ parents, λ children, and α individuals in the external archive with the best individuals [3]. Parents are selected from the external archive by using a binary tournament. Children are generated using SBX crossover and polynomial mutation. Children are evaluated and merged with the parents in a mixed population of size $\mu + \lambda$. In order to select the best α individuals, the MaxiMin algorithm is applied for this purpose. The NSGA-II ranks the population into fronts, for every front it calculates a crowding distance. Thus, we applied the MaxiMin algorithm to each front. Since the MaxiMin algorithm selects individuals one by one, we can attach a MaxiMin-rank to each individual, according to the order in which each individual is added to the selected set. Then, we assign $1/rank_{MaxiMin}$ as the crowding distance used to compare solutions in the same front. The MaxiMin is used again to give a new crowding distance to the selected set in the external archive. This crowding distance will be used for the binary tournament, when Pareto dominance is not enough to determine a winner.

4.2 Enhancing the SPEA2 with the MaxiMin selection.

In our implementation the SPEA2 algorithm uses the same crossover and mutation as the NSGA-II. Fitness assignment is described in 2 and for details in [4]. In order to apply the MaxiMin, we proceed as follows: after evaluation, individuals are sorted by fitness. We choose the first α individuals (α is the archive size), and individuals with the same fitness value as the individual in the α position. Among individuals with the same fitness as α, the MaxiMin algorithm is applied to select the needed individuals to fill an external archive of size α. MaxiMin is used again (as in NSGA-II) to give a new crowding distance to selected individuals (possible parents).

4.3 Enhancing DEMO with the MaxiMin selection.

The differential evolution for multi-objective optimization (DEMO) uses a special crossover to generate children [7]. A first selection step is performed by comparing offspring only with their own father. If a child dominates his parent the last is removed from the external archive. A second selection is performed at the end of the children's generation, when parents and children are merged in one pool. Members of the pool are ranked and selection proceeds as in NSGA-II with Maximin.

5 Performance metrics

Solteiro et al. in [8] use the spacing index (SP), the distance-based distribution index ($\Delta\prime$), and the minimal distance graph (MDG), in order to show the performance of the MaxiMin selection. These indices have some disadvantages. The SP index does not measure the coverage of solutions on the Pareto front, it only measures uniformity in solution space. Therefore, solutions having a very good spacing but clustered at one end of the front get very high marks. The $\Delta\prime$ index has the same problem, it does not weigh in any form the coverage of solutions. In addition to the mentioned disadvantages, these two measures (SP and $\Delta\prime$) are defined only for two dimensional problems. The MDG has the same absence of a coverage factor. In Solteiro's work [8] also no convergence metric is presented. Even though we are improving diversity of solutions, a convergence metric is necessary in order to analyze how diversity improvement impacts on convergence. Metrics must show that convergence and coverage-spacing behavior can be successfully achieved.

- **Convergence Metric: Generational distance**

  This metric finds the average distance of solutions Q from the reference front $P*$, as follows (Veldhuizen, 1999, see [1]):

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|} \tag{4}$$

  The metric was calculated for $p = 2$. The parameter d_i is the Euclidean distance (in the objective space) between the solution $i \in Q$ and the nearest member in P^*:

$$d_i = \min_{k=1}^{|P^*|} \left(\sqrt{\sum_{m=1}^{M} (f_m^{(i)} - f_m^{*(k)})^2} \right), \tag{5}$$

  where $f_m^{*(k)}$ are the objective function values, of the $k-th$ member in P^*. In order to have a less subjective measure, objective function values were normalized using the extreme values in the reference fronts.

- **Diversity Metric: Spread**

  The diversity metric suggested by Deb et al. (2000.a, see [1]) is in Equation 6. Where d_i can be any distance measure, in this case the Euclidean distance in the normalized objective space, between the $i-th$ solution and its consecutive neighbor for two-objective problems. For three objectives d_i is the distance to the nearest neighbor. The extreme points are selected as is explained in Section 3. Thus, $|Q|-1$ distances can be computed for two-objective problems, and $|Q|$ for three objective problems.

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|-1} |d_i - \bar{d}|}{\sum_{m=1}^{M} d_m^e + (|Q|-1)\bar{d}} \tag{6}$$

- **Normalized Diversity**

  In order to improve the comparison of the results, the column "ND" (for normalized diversity) is computed for all tables (each problem is reported in one table). Therefore, every algorithm reported in a table gets a ND value which indicates its performance rank. The algorithm with the best diversity value gets a 0; the worst one gets a 1. Normalized diversity is computed through Equation 7.

$$ND = \frac{\Delta - \Delta_{min}}{\Delta_{max} - \Delta_{min}} \tag{7}$$

  Where Δ is the mean of the diversity metric value, computed according Equation 6. Δ_{max} and Δ_{min} are the maximum and minimum means.

- **Coverage Metric: Maximum Spread**

  The metric presented in this section is based in the maximum spread metric defined by Zitzler [1]. This metric measures the length of the diagonal of a hyperbox formed by the extreme function values observed in the non-dominated set. The normalized version of the maximum spread metric is defined by Equation 8.

$$D = \sqrt{\sum_{m=1}^{M} \left(\frac{max_{i=1}^{|Q|} f_m^i - min_{i=1}^{|Q|} f_m^i}{F_m^{max} - F_m^{min}} \right)^2} \tag{8}$$

  A disadvantage of the maximum spread metric is that it does not show when the front is covered in all the functions, some function values can be larger than the reference values, and some other can be smaller, and in average the resulting metric could not represent the real coverage. Thus, a metric which measures the real coverage of the front is presented in Equation 9, the best metric value is 0 when the reference front and the front we are measuring have the same limits, otherwise it is greater than 0.

$$D = \sqrt{\sum_{m=1}^{M} \left(1 - \frac{max_{i=1}^{|Q|} f_m^i - min_{i=1}^{|Q|} f_m^i}{F_m^{max} - F_m^{min}} \right)^2} \tag{9}$$

6 Experiments

We present our comparative study of different evolutionary algorithms. The ZDT and DTLZ test have well known Pareto fronts; also, performance metrics are well known and widely used. A True Front of 500 uniformly distributed points was used to compute both performance metrics of every algorithm on the ZDT test suite (available at http://dis.ijs.si/tea/demo.htm). For DTLZ test suite, we use a set of about 10,000 points as Reference Front. Versions of NSGA-II, SPEA2, and DEMO, enhanced with MaxiMin and standard versions were used to solve the test suite problems. For all cases the number of function evaluations were 30,000.

6.1 ZDT test problems

The two-objective set of problems ZDT is given in Table 1. ZDT5 is discrete so it was not solved. Diversity and convergence metrics are calculated for the four algorithms with and without MaxiMin selection. Populations sizes of 66, 100, and 130 were used for comparison of this set of problems. The different parameters used by the algorithms are described next:

NSGA-II. Mutation probability was $1/n$, where n is the number of variables, and crossover probability was 0.9 for all cases. The η parameters used for the SBX crossover and polynomial mutation were 15 and 20 respectively. For the MaxiMin algorithm when it is inserted in NSGA-II, mutation probability was set in $1/(2n)$ for all populations, and probability crossover of 0.9 for all cases.

ZDT1	
Decision space	$x \in [0,1]^{30}$
Objective functions	$f_1(x) = x_1$
	$f_2(x) = g(x)(1 - \sqrt{x_1/g(x)})$
	$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
Optimal solutions	$0 \leq x_i^* \leq 1$ *and* $x_i^* = 0$ *for* $i = 2, \ldots, 30$
ZDT2	
Decision space	$x \in [0,1]^{30}$
Objective functions	$f_1(x) = x_1$
	$f_2(x) = g(x)(1 - (x_1/g(x))^2)$
	$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
Optimal solutions	$0 \leq x_i^* \leq 1$ *and* $x_i^* = 0$ *for* $i = 2, \ldots, 30$
ZDT3	
Decision space	$x \in [0,1]^{30}$
Objective functions	$f_1(x) = x_1$
	$f_2(x) = g(x)(1 - x_1/g(x) - \frac{x_1}{g(x)}\ sin(10\pi x_1)$
	$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
Optimal solutions	$0 \leq x_i^* \leq 1$ *and* $x_i^* = 0$ *for* $i = 2, \ldots, 30$
ZDT4	
Decision space	$x \in [0,1] \times [-5,5]^9$
Objective functions	$f_1(x) = x_1$
	$f_2(x) = g(x)(1 - (x_1/g(x))^2)$
	$g(x) = 1 + 10(n-1)\sum_{i=2}^{n}(x_i^2 - 10\ cos(4\pi x_i))$
Optimal solutions	$0 \leq x_i^* \leq 1$ *and* $x_i^* = 0$ *for* $i = 2, \ldots, 30$
ZDT6	
Decision space	$x \in [0,1]^{10}$
Objective functions	$f_1(x) = 1 - e^{-4x_1} sin(6\pi x_1)^6$
	$f_2(x) = g(x)(1 - (f_1(x)/g(x))^2)$
	$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
Optimal solutions	$0 \leq x_i^* \leq 1$ *and* $x_i^* = 0$ *for* $i = 2, \ldots, 30$

Table 1: The ZDT test problems

SPEA2. With and without MaxiMin, mutation probability was set in $1/(2n)$ for all cases. Individual crossover probability was 1.0, and variable crossover probability was 0.9

DEMO. For all cases when using DEMO, probability crossover was 0.3

ε-MOEA. Mutation probability was $1/n$ for all cases and crossover probability of 1.0. The ε values from ZDT1 to ZDT6 respectively for both objective functions.are

- for population size of 66:
  $[0.01137, 0.01145, 0.004005, 0.01137, 0.009]$,
- when using a population size of 100:
  $[0.0075, 0.0076, 0.00261, 0.0074, 0.00595]$,
- and for a population size of 130:
  $[0.0057, 0.0058, 0.002, 0.0057, 0.0046]$

The ε values were chosen such that sizes of the final archive were approximately 66, 100, and 130 for comparison purposes.

The parameters were chosen empirically, trying to obtain the best performance for each algorithm.

- Comments to ZDT1. All algorithms have convergence value close to zero (reached the True Front), but the diversity metric makes the difference. As we can see in Table 2, the MaxiMin algorithm notably improves the diversity of solutions, with no reduction of convergence properties of the algorithms. Pareto fronts are shown in Figure 3 for comparison purposes.
- Comments on ZDT2. Once again the MaxiMin version of each algorithm improves diversity of solutions. Results are shown in Table 3, and a sample view of the Pareto fronts are shown in Figure 4.

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	66	3.53E-01 (3.16E-02)	1.00	4.08E-04 (1.16E-04)	0.002 (6.67E-03)
NSGA-II (MaxiMin)	66	3.92E-02 (2.06E-02)	0.01	3.85E-04 (9.68E-05)	0.002 (6.90E-03)
SPEA2	66	3.29E-01 (2.91E-02)	0.92	3.70E-04 (7.08E-05)	0.016 (8.45E-03)
SPEA2 (MaxiMin)	66	3.76E-02 (7.33E-03)	0.00	3.67E-04 (1.61E-05)	0.000 (2.15E-04)
DEMO	66	3.17E-01 (2.66E-02)	0.89	2.61E-04 (5.66E-05)	0.000 (1.85E-09)
DEMO (MaxiMin)	66	3.86E-02 (3.86E-03)	0.00	3.37E-04 (3.75E-05)	0.000 (1.00E-09)
ε-MOEA	66	3.41E-01 (7.84E-03)	0.96	1.45E-04 (1.63E-05)	0.079 (1.15E-03)
NSGA-II	100	3.46E-01 (2.79E-02)	0.98	2.76E-04 (3.49E-05)	0.000 (2.43E-04)
NSGA-II (MaxiMin)	100	3.10E-01 (4.43E-03)	0.86	2.82E-04 (1.23E-05)	0.001 (5.13E-04)
SPEA2	100	3.21E-01 (2.10E-02)	0.90	2.86E-04 (3.29E-05)	0.010 (4.65E-03)
SPEA2 (MaxiMin)	100	3.08E-01 (5.26E-03)	0.86	2.82E-04 (1.39E-05)	0.001 (1.18E-03)
DEMO	100	3.18E-01 (2.59E-02)	0.89	2.29E-04 (2.68E-05)	0.000 (4.37E-07)
DEMO (MaxiMin)	100	3.09E-01 (4.05E-03)	0.86	2.41E-04 (3.55E-05)	0.000 (1.28E-07)
ε-MOEA	100	3.50E-01 (9.47E-03)	0.99	1.15E-04 (8.20E-06)	0.061 (1.08E-03)
NSGA-II	130	3.47E-01 (2.64E-02)	0.98	2.40E-04 (2.52E-05)	0.000 (2.36E-04)
NSGA-II (MaxiMin)	130	6.19E-02 (1.75E-02)	0.08	2.40E-04 (5.69E-06)	0.000 (3.94E-04)
SPEA2	130	3.11E-01 (2.30E-02)	0.87	2.58E-04 (2.52E-05)	0.006 (3.49E-03)
SPEA2 (MaxiMin)	130	5.95E-02 (1.39E-02)	0.07	2.40E-04 (8.90E-06)	0.000 (4.64E-04)
DEMO	130	3.32E-01 (2.67E-02)	0.93	2.01E-04 (2.38E-05)	0.000 (2.98E-06)
DEMO (MaxiMin)	130	6.25E-02 (9.24E-03)	0.08	2.19E-04 (1.84E-05)	0.000 (1.16E-06)
ε-MOEA	130	3.51E-01 (8.51E-03)	0.99	1.18E-04 (6.31E-06)	0.054 (2.24E-03)

Table 2: Diversity and convergence metrics for problem ZDT1

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	66	3.40E-01 (3.03E-02)	0.98	2.85E-04 (8.95E-05)	0.000 (5.23E-05)
NSGA-II (MaxiMin)	66	4.11E-02 (1.41E-02)	0.02	1.36E-04 (2.76E-05)	0.000 (4.50E-04)
SPEA2	66	3.38E-01 (2.93E-02)	0.97	2.24E-04 (9.37E-05)	0.011 (5.84E-03)
SPEA2 (MaxiMin)	66	3.61E-02 (3.24E-03)	0.00	1.39E-04 (2.58E-05)	0.000 (4.20E-04)
DEMO	66	3.24E-01 (3.20E-02)	0.93	1.16E-04 (7.66E-06)	0.000 (6.29E-09)
DEMO (MaxiMin)	66	3.75E-02 (2.42E-03)	0.00	1.16E-04 (3.88E-06)	0.000 (1.00E-09)
ε-MOEA	66	2.54E-01 (1.15E-02)	0.70	1.45E-04 (1.50E-05)	0.006 (5.78E-04)
NSGA-II	100	3.47E-01 (2.57E-02)	1.00	1.59E-04 (4.21E-05)	0.001 (3.70E-03)
NSGA-II (MaxiMin)	100	3.08E-01 (3.91E-03)	0.87	1.11E-04 (1.82E-05)	0.000 (1.67E-04)
SPEA2	100	3.09E-01 (2.47E-02)	0.88	1.45E-04 (3.45E-05)	0.008 (4.58E-03)
SPEA2 (MaxiMin)	100	3.09E-01 (3.47E-03)	0.88	1.04E-04 (1.49E-05)	0.000 (1.71E-04)
DEMO	100	3.24E-01 (2.75E-02)	0.93	9.36E-05 (3.95E-06)	0.000 (5.87E-07)
DEMO (MaxiMin)	100	3.04E-01 (3.63E-03)	0.86	9.04E-05 (3.79E-06)	0.000 (1.66E-07)
ε-MOEA	100	2.59E-01 (8.73E-03)	0.72	1.29E-04 (8.46E-06)	0.001 (3.85E-04)
NSGA-II	130	3.44E-01 (3.01E-02)	0.99	1.30E-04 (1.33E-05)	0.001 (4.78E-04)
NSGA-II (MaxiMin)	130	7.60E-02 (1.37E-02)	0.13	1.09E-04 (8.47E-06)	0.001 (2.39E-04)
SPEA2	130	3.25E-01 (1.84E-02)	0.93	1.75E-04 (8.13E-05)	0.007 (8.00E-03)
SPEA2 (MaxiMin)	130	7.98E-02 (1.63E-02)	0.14	1.12E-04 (8.43E-06)	0.001 (3.63E-04)
DEMO	130	3.22E-01 (1.97E-02)	0.92	8.27E-05 (3.70E-06)	0.000 (4.14E-06)
DEMO (MaxiMin)	130	5.15E-02 (6.08E-03)	0.05	8.30E-05 (2.76E-06)	0.000 (2.04E-06)
ε-MOEA	130	2.66E-01 (9.20E-03)	0.74	1.34E-04 (1.15E-05)	0.002 (1.35E-03)

Table 3: Diversity and convergence metrics for problem ZDT2

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	66	7.09E-01 (1.91E-02)	0.28	1.67E-04 (2.61E-05)	0.014 (5.27E-02)
NSGA-II (MaxiMin)	66	6.62E-01 (8.21E-03)	0.01	1.14E-04 (1.65E-05)	0.008 (3.79E-02)
SPEA2	66	7.35E-01 (2.06E-02)	0.43	2.03E-04 (9.00E-05)	0.018 (3.93E-02)
SPEA2 (MaxiMin)	66	6.60E-01 (7.66E-04)	0.00	1.13E-04 (7.02E-06)	0.000 (1.39E-04)
DEMO	66	6.93E-01 (1.38E-02)	0.19	1.16E-04 (1.25E-05)	0.000 (1.16E-04)
DEMO (MaxiMin)	66	6.62E-01 (3.27E-03)	0.01	1.16E-04 (7.52E-06)	0.000 (9.66E-05)
ε-MOEA	66	8.09E-01 (3.33E-03)	0.85	8.85E-05 (4.76E-06)	0.032 (3.66E-02)
NSGA-II	100	7.41E-01 (1.30E-02)	0.46	1.19E-04 (1.26E-05)	0.000 (4.55E-04)
NSGA-II (MaxiMin)	100	7.03E-01 (2.61E-03)	0.25	9.52E-05 (4.56E-06)	0.007 (3.79E-02)
SPEA2	100	7.60E-01 (1.70E-02)	0.57	1.31E-04 (2.71E-05)	0.007 (5.14E-03)
SPEA2 (MaxiMin)	100	7.05E-01 (5.03E-03)	0.26	9.55E-05 (4.68E-06)	0.000 (2.41E-04)
DEMO	100	7.25E-01 (7.61E-03)	0.37	9.34E-05 (6.36E-06)	0.000 (1.47E-04)
DEMO (MaxiMin)	100	7.07E-01 (4.79E-03)	0.27	9.43E-05 (5.95E-06)	0.000 (1.73E-04)
ε-MOEA	100	8.31E-01 (7.20E-03)	0.98	7.79E-05 (4.35E-06)	0.020 (6.42E-04)
NSGA-II	130	7.58E-01 (8.80E-03)	0.56	1.05E-04 (8.76E-06)	0.000 (2.11E-04)
NSGA-II (MaxiMin)	130	7.23E-01 (2.87E-03)	0.36	9.47E-05 (6.16E-06)	0.000 (2.27E-04)
SPEA2	130	7.70E-01 (1.27E-02)	0.63	1.14E-04 (8.96E-06)	0.005 (2.82E-03)
SPEA2 (MaxiMin)	130	7.23E-01 (3.96E-03)	0.36	9.35E-05 (6.89E-06)	0.000 (2.18E-04)
DEMO	130	7.45E-01 (7.08E-03)	0.49	8.42E-05 (6.36E-06)	0.000 (1.86E-04)
DEMO (MaxiMin)	130	7.25E-01 (4.29E-03)	0.37	8.45E-05 (7.94E-06)	0.000 (1.82E-04)
ε-MOEA	130	8.35E-01 (4.42E-03)	1.00	7.30E-05 (3.73E-06)	0.024 (3.71E-02)

Table 4: Diversity and convergence metrics for problem ZDT3

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	66	3.49E-01 (4.19E-02)	0.69	4.43E-04 (1.83E-04)	0.002 (1.25E-03)
NSGA-II (MaxiMin)	66	5.44E-02 (2.92E-02)	0.01	4.24E-04 (1.03E-04)	0.001 (1.32E-03)
SPEA2	66	4.61E-01 (3.41E-01)	0.94	1.98E-02 (6.85E-02)	0.923 (3.19E+00)
SPEA2 (MaxiMin)	66	5.07E-02 (1.82E-02)	0.00	4.22E-04 (7.64E-05)	0.001 (1.01E-03)
DEMO	66	4.85E-01 (1.18E-01)	1.00	2.33E-02 (2.79E-02)	0.107 (1.09E-01)
DEMO (MaxiMin)	66	1.76E-01 (1.39E-01)	0.29	1.28E-02 (1.57E-02)	0.062 (7.12E-02)
ε-MOEA	66	3.56E-01 (1.21E-02)	0.70	3.44E-04 (1.44E-04)	0.074 (2.70E-03)
NSGA-II	100	3.41E-01 (2.93E-02)	0.67	3.53E-04 (1.23E-04)	0.002 (1.44E-03)
NSGA-II (MaxiMin)	100	3.05E-01 (8.41E-03)	0.59	3.34E-04 (8.34E-05)	0.001 (9.92E-04)
SPEA2	100	3.89E-01 (2.50E-01)	0.78	6.13E-03 (2.70E-02)	0.417 (1.91E+00)
SPEA2 (MaxiMin)	100	3.10E-01 (2.26E-02)	0.60	3.68E-04 (2.51E-04)	0.005 (2.05E-02)
DEMO	100	3.68E-01 (5.20E-02)	0.73	1.46E-03 (3.76E-03)	0.008 (2.54E-02)
DEMO (MaxiMin)	100	3.47E-01 (6.38E-02)	0.68	3.93E-03 (5.75E-03)	0.025 (3.89E-02)
ε-MOEA	100	3.52E-01 (1.38E-02)	0.69	2.59E-04 (1.36E-04)	0.061 (8.38E-03)
NSGA-II	130	3.34E-01 (2.26E-02)	0.65	3.41E-04 (1.15E-04)	0.002 (1.95E-03)
NSGA-II (MaxiMin)	130	1.12E-01 (4.35E-02)	0.14	3.02E-04 (1.59E-04)	0.001 (1.69E-03)
SPEA2	130	3.91E-01 (2.53E-01)	0.78	6.03E-03 (2.74E-02)	0.534 (2.52E+00)
SPEA2 (MaxiMin)	130	1.08E-01 (5.59E-02)	0.13	2.71E-04 (4.29E-05)	0.002 (6.08E-03)
DEMO	130	3.55E-01 (3.15E-02)	0.70	5.39E-04 (1.97E-03)	0.003 (1.52E-02)
DEMO (MaxiMin)	130	1.22E-01 (3.70E-02)	0.16	5.27E-04 (1.94E-03)	0.003 (1.52E-02)
ε-MOEA	130	3.52E-01 (1.10E-02)	0.69	2.54E-04 (8.79E-05)	0.053 (7.57E-03)

Table 5: Diversity and convergence metrics for problem ZDT4

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	66	6.28E-01 (5.56E-02)	0.74	2.25E-04 (2.42E-05)	0.119 (1.16E-04)
NSGA-II (MaxiMin)	66	1.59E-01 (4.73E-03)	0.02	2.88E-04 (1.18E-05)	0.119 (1.52E-04)
SPEA2	66	4.99E-01 (2.45E-01)	0.54	5.27E-03 (1.30E-02)	0.313 (5.31E-01)
SPEA2 (MaxiMin)	66	1.58E-01 (3.65E-03)	0.01	2.86E-04 (1.44E-05)	0.119 (1.68E-04)
DEMO	66	7.93E-01 (4.24E-02)	1.00	1.08E-04 (8.46E-06)	0.119 (6.39E-08)
DEMO (MaxiMin)	66	1.49E-01 (2.95E-03)	0.00	9.92E-05 (4.75E-06)	0.119 (3.58E-07)
ε-MOEA	66	2.75E-01 (9.33E-03)	0.20	3.02E-04 (1.23E-05)	0.127 (4.21E-03)
NSGA-II	100	4.18E-01 (2.69E-02)	0.42	4.91E-04 (6.92E-05)	0.120 (3.73E-04)
NSGA-II (MaxiMin)	100	3.81E-01 (9.08E-03)	0.36	5.53E-04 (7.80E-05)	0.120 (3.57E-04)
SPEA2	100	5.80E-01 (2.84E-01)	0.67	5.29E-03 (8.91E-03)	0.397 (5.58E-01)
SPEA2 (MaxiMin)	100	3.80E-01 (7.18E-03)	0.36	5.46E-04 (8.26E-05)	0.120 (3.31E-04)
DEMO	100	7.92E-01 (4.61E-02)	1.00	8.76E-05 (5.01E-06)	0.119 (1.72E-07)
DEMO (MaxiMin)	100	3.93E-01 (3.26E-03)	0.38	8.18E-05 (3.69E-06)	0.119 (9.63E-08)
ε-MOEA	100	2.90E-01 (1.19E-02)	0.22	3.25E-04 (3.04E-05)	0.123 (7.42E-04)
NSGA-II	130	3.99E-01 (2.27E-02)	0.39	9.36E-04 (1.00E-04)	0.121 (9.35E-04)
NSGA-II (MaxiMin)	130	3.12E-01 (1.83E-02)	0.25	1.12E-03 (1.27E-04)	0.122 (7.92E-04)
SPEA2	130	5.83E-01 (3.18E-01)	0.67	5.89E-03 (8.93E-03)	0.488 (7.41E-01)
SPEA2 (MaxiMin)	130	3.05E-01 (1.90E-02)	0.24	1.06E-03 (1.08E-04)	0.122 (7.50E-04)
DEMO	130	7.88E-01 (3.75E-02)	0.99	7.63E-05 (4.18E-06)	0.119 (8.55E-08)
DEMO (MaxiMin)	130	1.60E-01 (9.62E-03)	0.02	7.36E-05 (2.30E-06)	0.119 (4.93E-07)
ε-MOEA	130	3.00E-01 (1.30E-02)	0.23	3.67E-04 (3.07E-05)	0.121 (1.64E-03)

Table 6: Diversity and convergence metrics for problem ZDT6

- Comments on ZDT3. The ZDT3 problem has a disconnected Pareto front, it is especially interesting for this characteristic. As we can observe, there is no need of any tuning to use the MaxiMin, diversity is improved for all cases in comparison to the original selection of each algorithm. Results shown in Table 4, and a sample view of the Pareto fronts are shown in Figure 5.
- Comments on ZDT4. We can observe that behavior with or without MaxiMin is similar in convergence. As in the other cases MaxiMin improves diversity. Results shown in Table 5, and and a sample view of the Pareto fronts are shown in Figure 6.
- Comments to ZDT6. In the last problem of this set, MaxiMin improves diversity maintaining convergence. Results shown in Table 6, and a sample view of the Pareto fronts are shown in Figure 7.

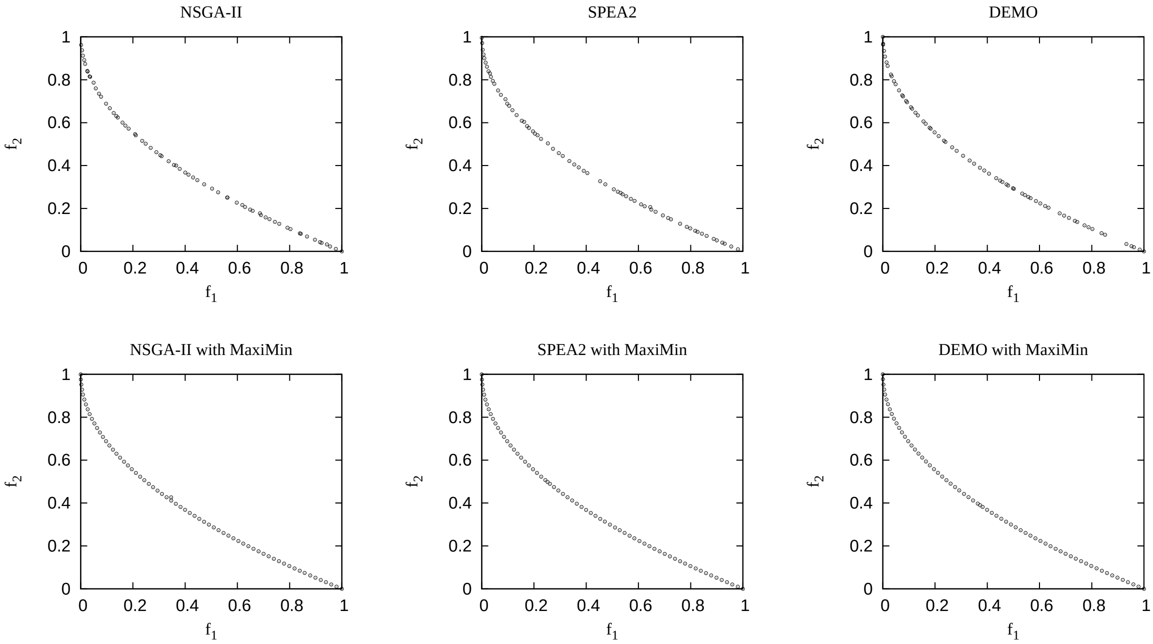

Figure 3: Graphical comparison for problem ZDT1

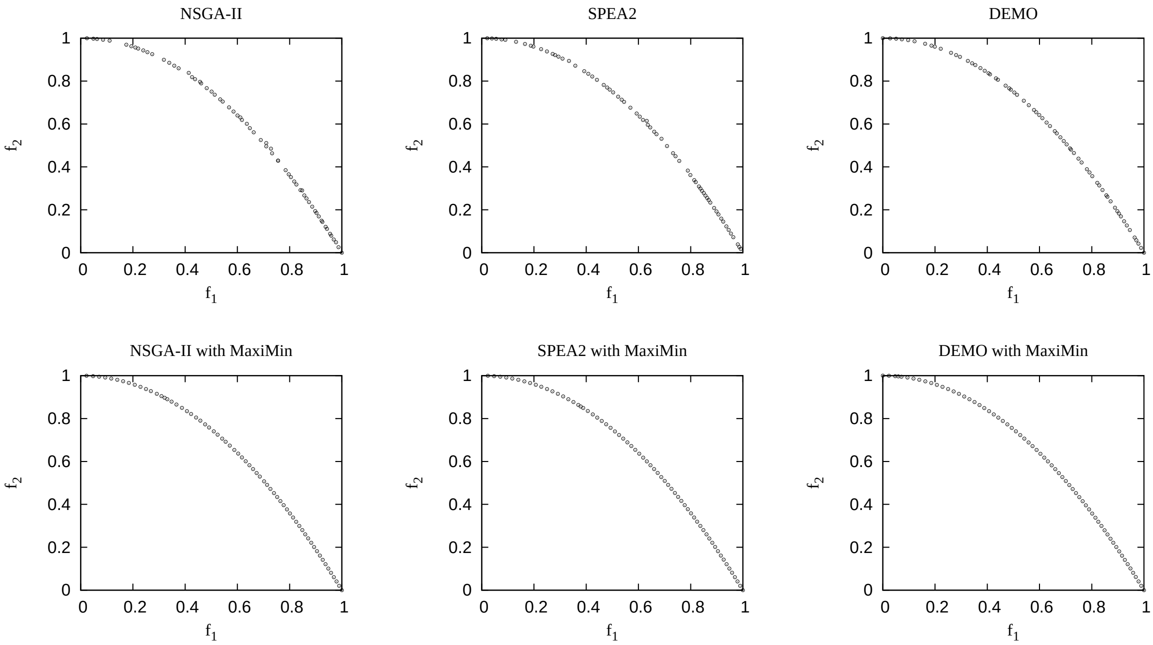

Figure 4: Graphical comparison for problem ZDT2

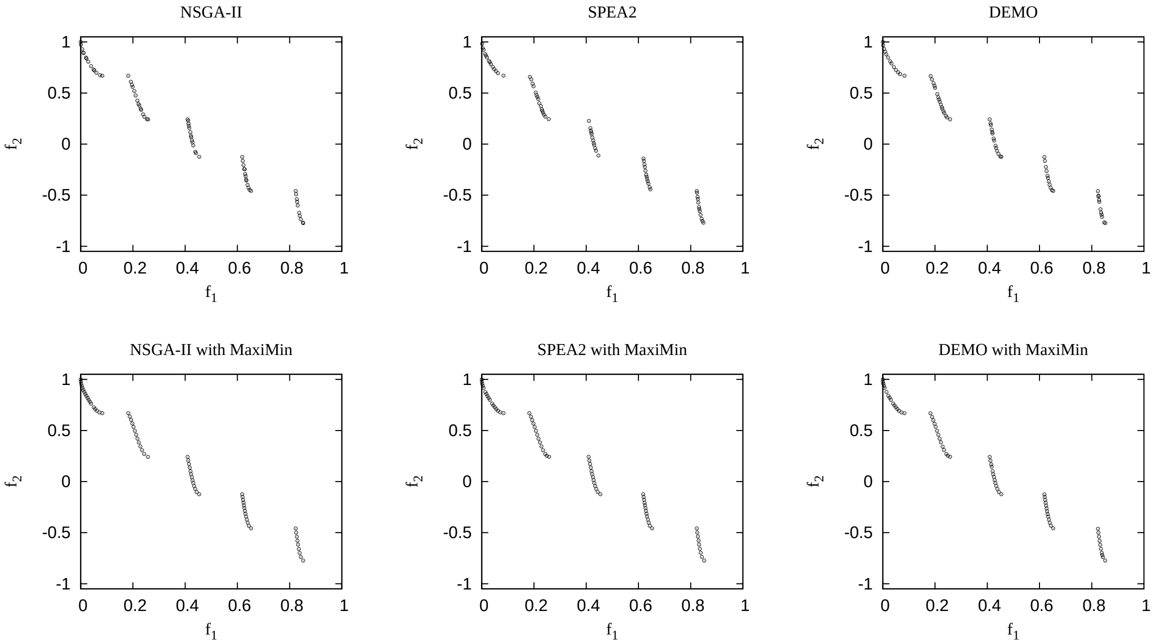

Figure 5: Graphical comparison for problem ZDT3

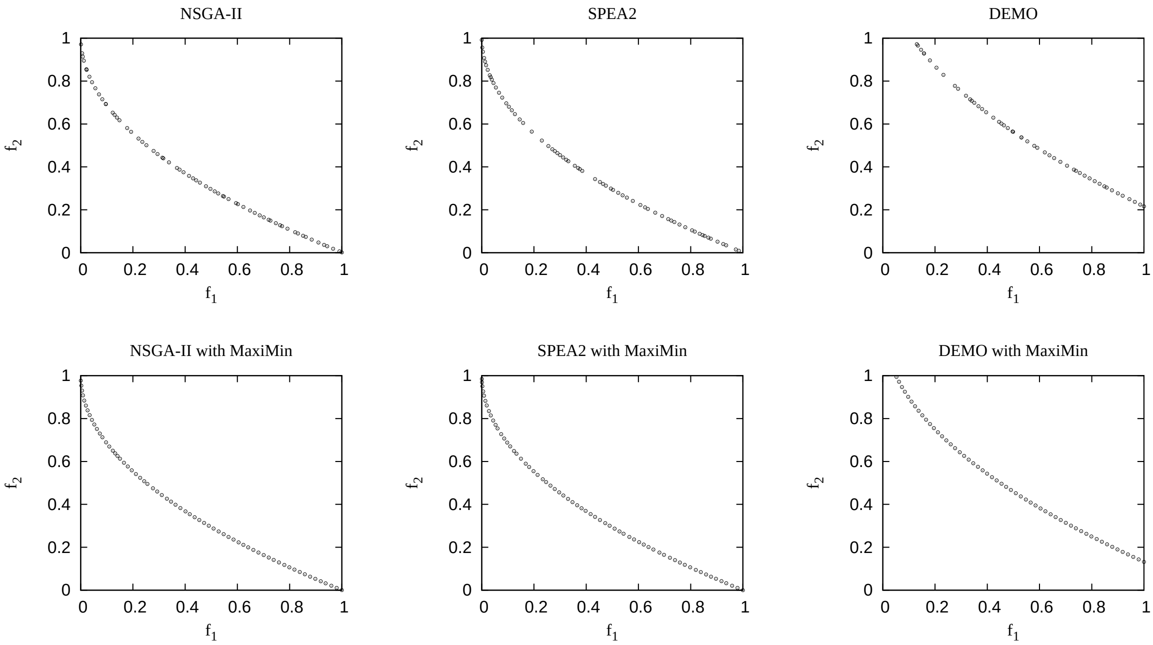

Figure 6: Graphical comparison for problem ZDT4

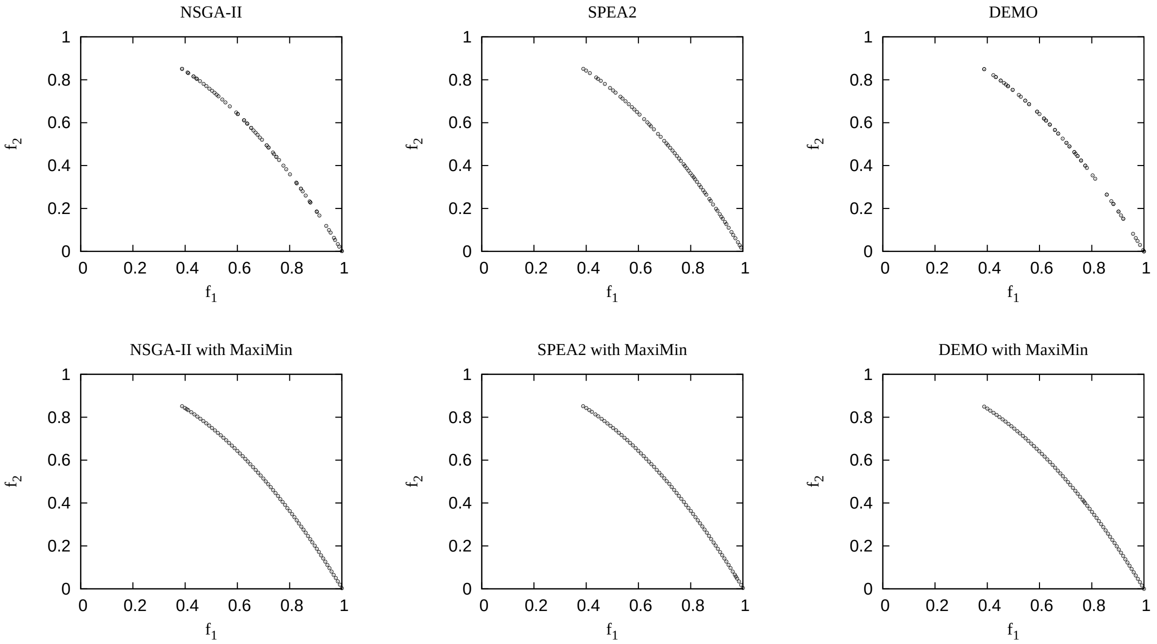

Figure 7: Graphical comparison for problem ZDT6

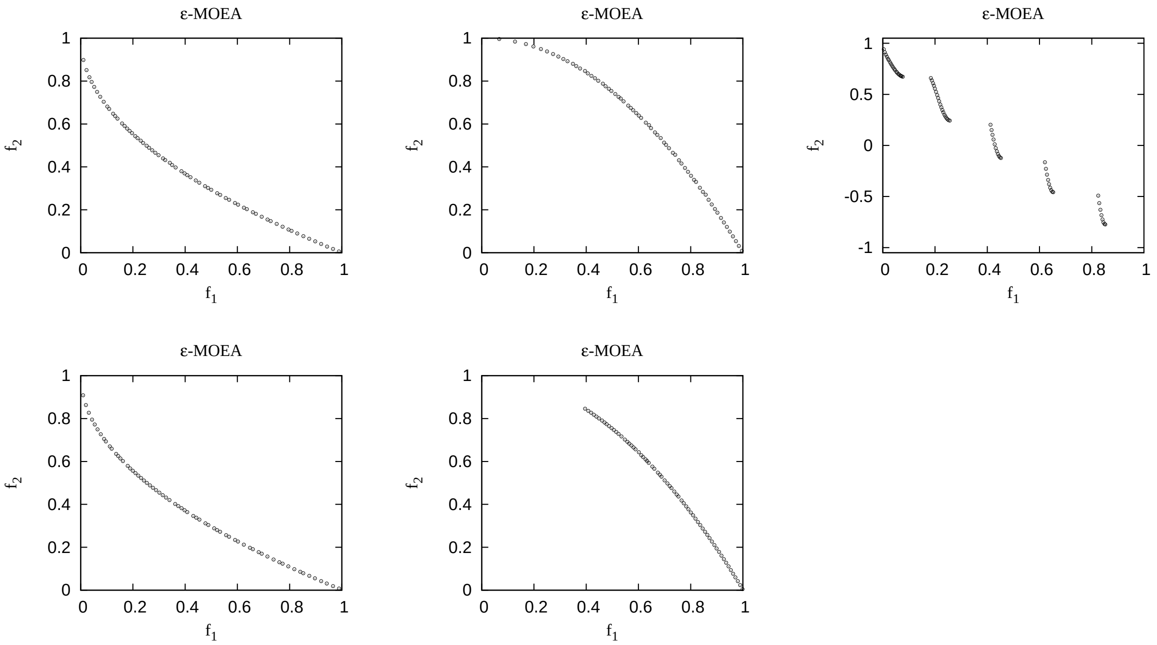

Figure 8: Pareto fronts from ε-MOEA runs, for the ZDT test problems.

DTLZ1	
Decision space	$x \in [0,1]^7$
Objective functions	$f_1(x) = 0.5(1+g(x_M))x_1x_2$
	$f_2(x) = 0.5(1+g(x_M))(1-x_2)x_1$
	$f_3(x) = 0.5(1+g(x_M))(1-x_1)$
	$g(x_M) = 100+(5+\sum_{i=3}^{7}(x_i-0.5)^2)-cos(20\pi(x_i-0.5)))$
Optimal solutions	$0 \leq x_1, x_2^* \leq 1$ *and* $x_i^* = 0.5$ *for* $i = 3,...,7$
DTLZ2	
Decision space	$x \in [0,1]^{12}$
Objective functions	$f_1(x) = (1+g(x_M))\ cos(x_1\pi/2)\ cos(x_2\pi/2)$
	$f_2(x) = (1+g(x_M))\ cos(x_1\pi/2)\ sin(x_2\pi/2)$
	$f_3(x) = (1+g(x_M))\ sin(x_2\pi/2)$
	$g(x_M) = \sum_{i=3}^{12}(x_i-0.5)^2$
Optimal solutions	$0 \leq x_1, x_2^* \leq 1$ *and* $x_i^* = 0.5$ *for* $i = 3,...,7$
DTLZ4	
Decision space	$y \in [0,1]^{12}$
Mapping	$x_i = y_i^1 00$
Objective functions	$f_1(x) = (1+g(x_M))\ cos(x_1\pi/2)\ cos(x_2\pi/2)$
	$f_2(x) = (1+g(x_M))\ cos(x_1\pi/2)\ sin(x_2\pi/2)$
	$f_3(x) = (1+g(x_M))\ sin(x_2\pi/2)$
	$g(x_M) = \sum_{i=3}^{12}(x_i-0.5)^2$
DTLZ5	
Decision space	$x \in [0,1]^{12}$
Objective functions	$f_1(x) = (1+g(x_M))\ cos(\theta_1)\ cos(\theta_2)$
	$f_2(x) = (1+g(x_M))\ cos(\theta_1)\ sin(\theta_2)$
	$f_3(x) = (1+g(x_M))\ sin(\theta_2)$
	$g(x_M) = \sum_{i=3}^{12}(x_i-0.5)^2$
	$\theta_j = \frac{\pi}{4(1+g(x_M))}(1+2g(x_M)x_j)$ *for* $j = 1,2$

Table 7: The DTLZ test functions

6.2 DTLZ test problems

The DTLZ problems are a scalable set proposed by Deb et al. [12]. These problems can be built for any number of variables and objective functions. Nonetheless, a three-objective version was solved by NSGA-II and SPEA2 algorithms, with and without Maximin selection. The DTLZ test functions are listed in Table 7, and diversity and convergence metrics are shown in Table 8 to 11. The MaxiMin algorithm can be applied for any number of objective functions without any change, and a similar behavior is expected, regardless of the number of objectives. For some algorithms as NSGA-II and DEMO, it can be observed that its diversity mechanism has a poor performance, that becomes worse when dimensionality of the objective space is increased. Due to convergence problems, DEMO results are only shown for the first problem. Most of the algorithms have problems to converge in DTLZ3 problem; thus, in this case comparison is not convenient.

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	100	6.92E-01 (3.40E-01)	1.00	4.87E-02 (1.18E-01)	2.250 (6.64E+00)
NSGA-II (MaxiMin)	100	1.69E-01 (1.72E-01)	0.18	9.45E-03 (2.09E-02)	0.196 (3.76E-01)
SPEA2	100	2.13E-01 (6.35E-02)	0.25	7.57E-03 (2.06E-02)	0.245 (5.27E-01)
SPEA2 (MaxiMin)	100	1.10E-01 (6.15E-02)	0.09	1.74E-03 (4.59E-03)	0.062 (2.14E-01)
DEMO	100	4.42E-01 (4.54E-02)	0.61	4.96E-03 (1.76E-02)	0.067 (2.54E-01)
DEMO (MaxiMin)	100	7.33E-02 (3.66E-02)	0.03	1.35E-02 (2.68E-02)	0.200 (4.07E-01)
ε-MOEA	100	1.89E-01 (3.55E-02)	0.21	8.43E-04 (4.52E-04)	0.088 (1.26E-02)
NSGA-II	136	6.30E-01 (2.30E-01)	0.90	3.97E-02 (7.04E-02)	1.710 (3.41E+00)
NSGA-II (MaxiMin)	136	1.83E-01 (1.26E-01)	0.20	7.93E-03 (1.37E-02)	0.386 (6.34E-01)
SPEA2	136	3.22E-01 (1.62E-01)	0.42	1.43E-02 (1.72E-02)	0.726 (7.87E-01)
SPEA2 (MaxiMin)	136	2.65E-01 (3.96E-01)	0.33	5.03E-02 (1.52E-01)	3.060 (1.18E+01)
DEMO	136	4.48E-01 (3.36E-02)	0.62	2.90E-04 (7.39E-06)	0.000 (5.75E-06)
DEMO (MaxiMin)	136	5.18E-02 (7.91E-03)	0.00	3.01E-04 (6.94E-05)	0.000 (2.07E-03)
ε-MOEA	136	1.81E-01 (2.22E-02)	0.20	6.77E-04 (2.44E-04)	0.087 (1.38E-02)

Table 8: Diversity and convergence metrics for problem DTLZ1

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	100	5.26E-01 (4.99E-02)	1.00	2.49E-03 (4.49E-03)	0.050 (1.07E-01)
NSGA-II (MaxiMin)	100	1.42E-01 (1.43E-02)	0.04	8.17E-04 (1.48E-04)	0.006 (7.46E-03)
SPEA2	100	1.70E-01 (1.43E-02)	0.11	1.13E-03 (1.69E-04)	0.011 (9.70E-03)
SPEA2 (MaxiMin)	100	1.46E-01 (1.72E-02)	0.05	7.89E-04 (1.43E-04)	0.006 (6.96E-03)
ε-MOEA	100	2.57E-01 (1.23E-02)	0.33	9.76E-04 (5.44E-05)	0.014 (5.60E-03)
NSGA-II	136	5.07E-01 (4.12E-02)	0.95	1.07E-03 (1.35E-04)	0.021 (1.11E-02)
NSGA-II (MaxiMin)	136	1.25E-01 (1.39E-02)	0.00	6.43E-04 (1.06E-04)	0.006 (8.48E-03)
SPEA2	136	1.63E-01 (1.47E-02)	0.09	9.36E-04 (8.50E-05)	0.009 (8.24E-03)
SPEA2 (MaxiMin)	136	1.27E-01 (1.21E-02)	0.00	6.81E-04 (1.12E-04)	0.006 (7.17E-03)
ε-MOEA	136	2.51E-01 (9.83E-03)	0.31	7.41E-04 (5.14E-05)	0.010 (7.27E-03)

Table 9: Diversity and convergence metrics for problem DTLZ2

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	100	5.12E-01 (7.28E-02)	1.00	1.25E-03 (3.01E-04)	0.054 (1.43E-01)
NSGA-II (MaxiMin)	100	1.60E-01 (8.18E-02)	0.04	8.52E-04 (1.93E-04)	0.029 (1.04E-01)
SPEA2	100	3.10E-01 (2.24E-01)	0.45	1.47E-03 (6.78E-04)	0.188 (2.60E-01)
SPEA2 (MaxiMin)	100	1.62E-01 (8.61E-02)	0.05	8.19E-04 (1.62E-04)	0.029 (1.04E-01)
ε-MOEA	100	4.57E-01 (1.94E-01)	0.85	1.57E-03 (1.64E-03)	0.335 (2.82E-01)
NSGA-II	136	5.02E-01 (5.83E-02)	0.97	9.90E-04 (1.71E-04)	0.033 (1.03E-01)
NSGA-II (MaxiMin)	136	1.45E-01 (9.10E-02)	0.00	7.25E-04 (1.29E-04)	0.029 (1.04E-01)
SPEA2	136	2.07E-01 (1.47E-01)	0.17	1.38E-03 (3.90E-04)	0.083 (1.69E-01)
SPEA2 (MaxiMin)	136	1.72E-01 (1.44E-01)	0.07	6.79E-04 (1.90E-04)	0.068 (1.73E-01)
ε-MOEA	136	3.36E-01 (1.87E-01)	0.52	6.93E-04 (2.98E-04)	0.106 (2.15E-01)

Table 10: Diversity and convergence metrics for problem DTLZ4

Algorithm	Pop Size	Diversity Mean (StdDev)	ND	Convergence Mean (StdDev)	Max. Spread Mean (StdDev)
NSGA-II	100	4.39E-01 (5.71E-02)	0.79	9.96E-04 (2.83E-03)	0.024 (7.87E-02)
NSGA-II (MaxiMin)	100	2.33E-01 (2.74E-02)	0.24	1.77E-04 (7.57E-05)	0.001 (1.08E-03)
SPEA2	100	2.77E-01 (2.88E-02)	0.35	2.34E-04 (7.06E-05)	0.007 (3.97E-03)
SPEA2 (MaxiMin)	100	2.37E-01 (2.35E-02)	0.25	1.38E-04 (7.26E-05)	0.000 (1.13E-03)
ε-MOEA	100	4.78E-01 (2.28E-02)	0.90	7.46E-05 (1.08E-05)	0.001 (7.63E-04)
NSGA-II	136	4.27E-01 (4.96E-02)	0.76	2.02E-04 (9.83E-05)	0.004 (9.12E-03)
NSGA-II (MaxiMin)	136	1.45E-01 (3.54E-02)	0.00	1.20E-04 (4.04E-05)	0.000 (8.77E-04)
SPEA2	136	2.64E-01 (2.82E-02)	0.32	1.54E-04 (1.18E-04)	0.005 (1.00E-02)
SPEA2 (MaxiMin)	136	1.51E-01 (2.57E-02)	0.02	1.24E-04 (4.44E-05)	0.001 (1.40E-03)
ε-MOEA	136	5.17E-01 (2.44E-02)	1.00	5.69E-05 (4.91E-06)	0.001 (5.52E-04)

Table 11: Diversity and convergence metrics for problem DTLZ5

- Comments to DTLZ1. For the DTLZ1 problem all the algorithms show good convergence and diversity values, but note that the MaxiMin version of each algorithm improves the original. A view of the Pareto fronts is shown in Figure 6.2.
- Comments to DTLZ2. For DTLZ2 problem the metrics give similar bahavior as for DTLZ1, i.e., MaxiMin version is better. A view of the Pareto fronts is shown in Figure 6.2.
- Comments to DTLZ4. Algorithms with MaxiMin can achieve a good convergence as well as diversity. A view of the Pareto fronts is shown in Figure 6.2.
- Comments to DTLZ5. Convergence metric reports the best values among all problems, and diversity is well close to zero. Notice that MaxiMin outperforms ε-MOEA. A view of the Pareto fronts is shown in Figure 6.2.

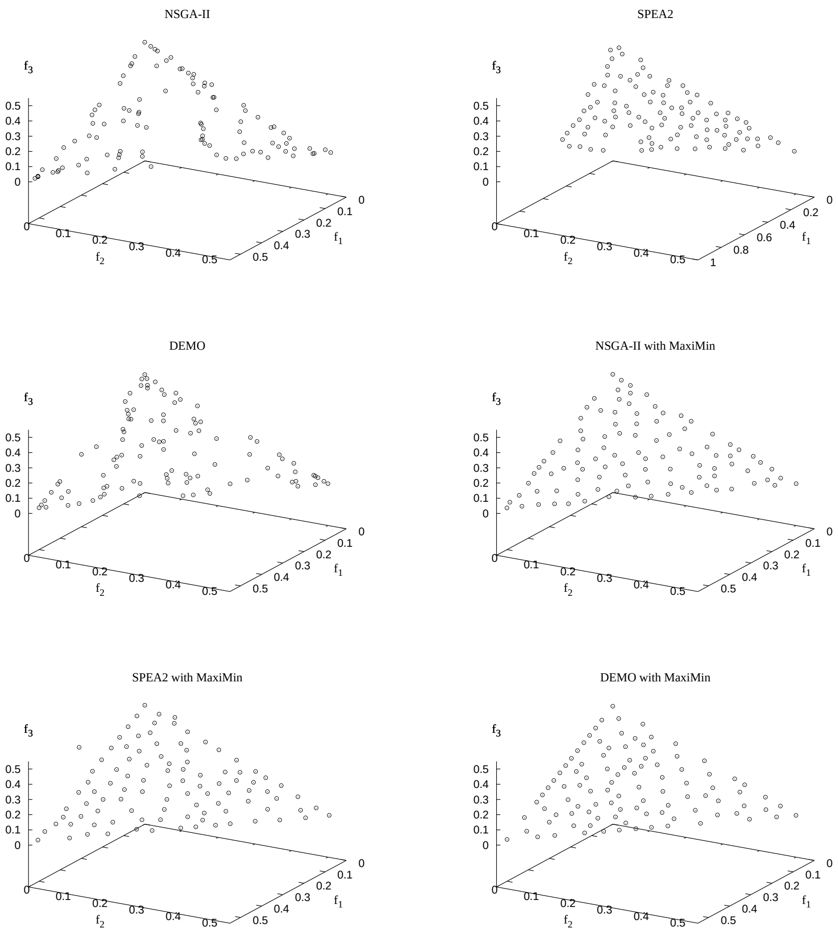

Figure 9: Graphical comparison for problem DTLZ1.

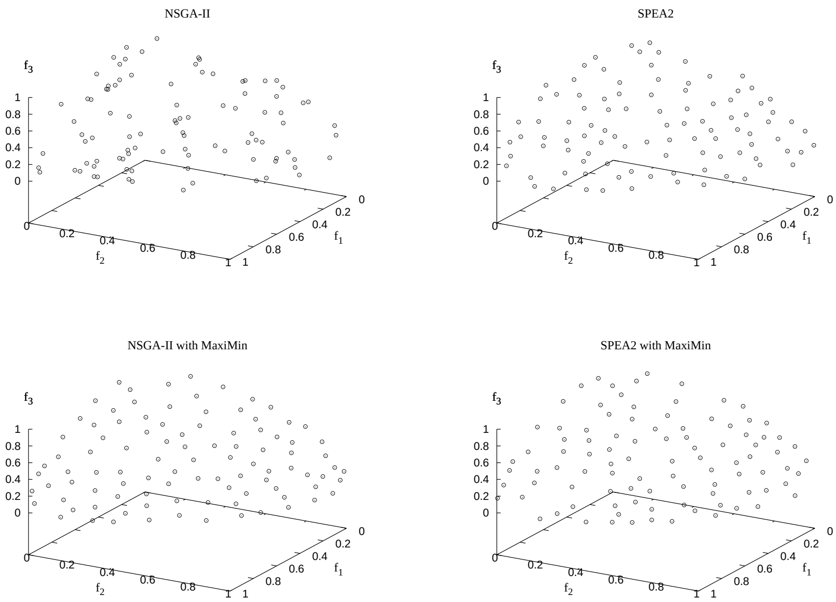

Figure 10: Graphical comparison for problemDTLZ2.

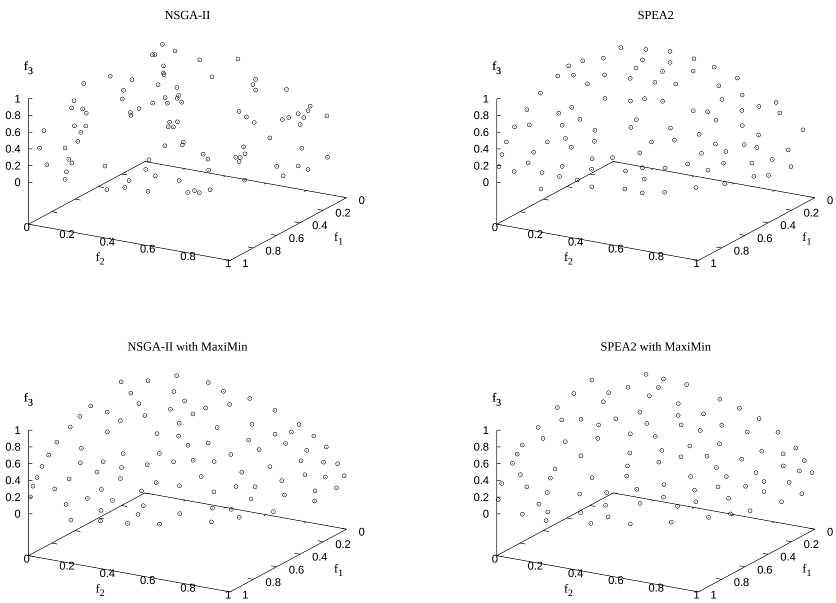

Figure 11: Graphical comparison for problem DTLZ4.

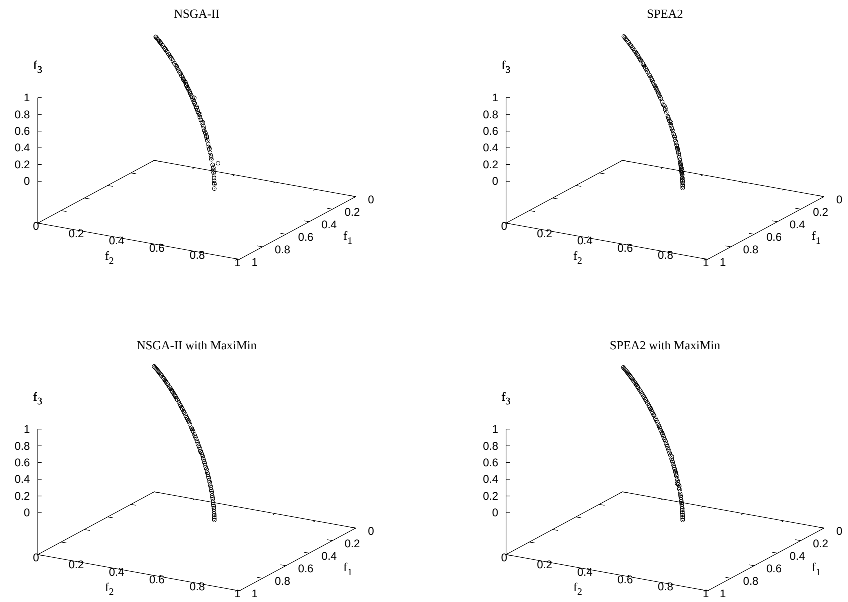

Figure 12: Graphical comparison for problem DTLZ5.

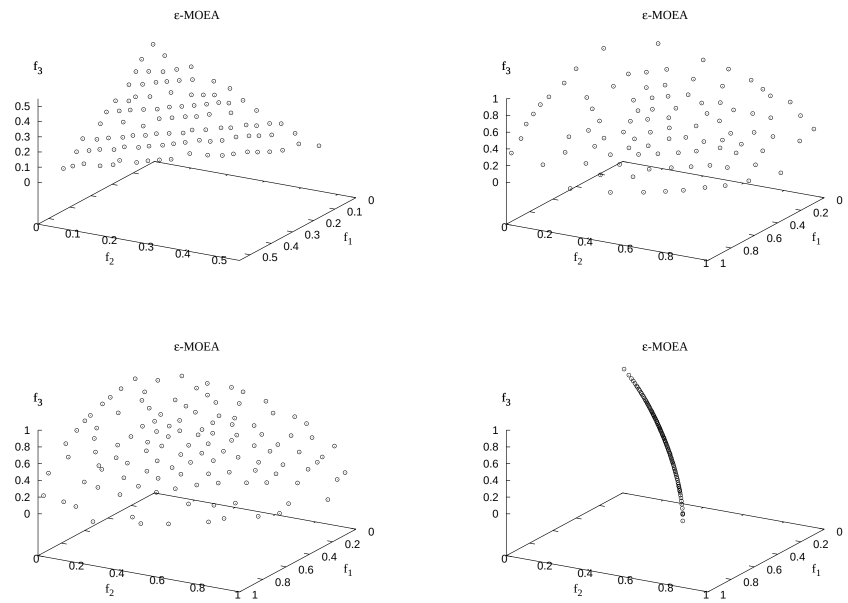

Figure 13: Pareto fronts from ε-MOEA runs, for the DTLZ test problems.

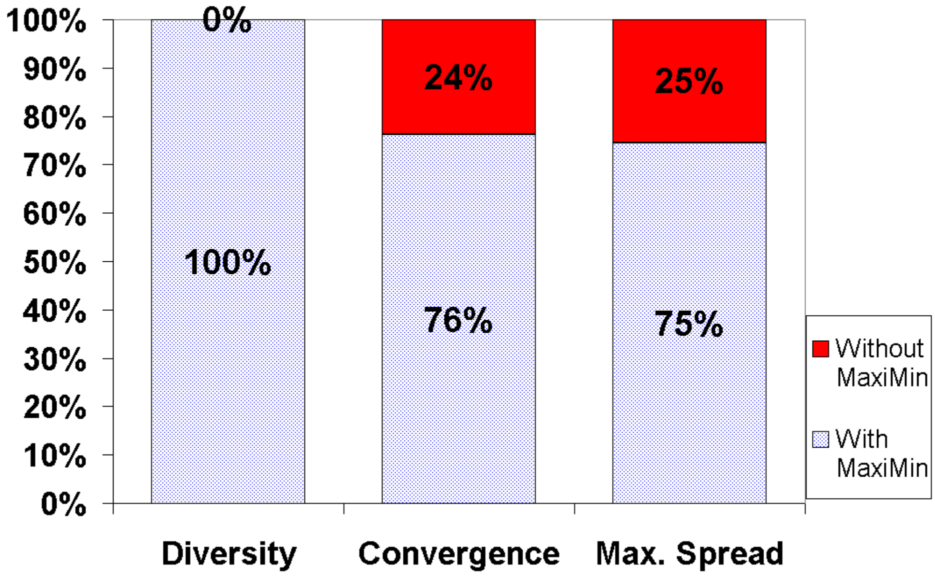

Figure 14: Relative comparison among algorithms with and without MaxiMin

6.3 General MaxiMin Performance

According to experiments developed, the MaxiMin selection substantially improves the search engines in SPEA2, DEMO and NSGA-II. Figure 14 shows a relative comparison among algorithms with and without MaxiMin. To create the graphic, the MaxiMin version of the algorithms are compared with the original versions. We took the means of the performance metrics presented in Tables 2 to 11. Then, we compared the mean of the MaxiMin version against the original, for the same population size. Counting the number of means in which the MaxiMin versions are better than the original over the total number of sets of runs a percentage is calculated. The Figure 14 clearly shows that for all cases the diversity is improved by the MaxiMin versions, and most of the cases convergence and maximum spread are improved too.

Observe that for each problem there is a MaxiMin version which has a better diversity value than ε-MOEA.

7 Conclusions

A novel strategy to discriminate overcrowded points in the Pareto front is presented. The MaxiMin algorithm performs as well or better than others for diversity, maintaining or even improving convergence. This algorithm, immune to the different sizes of the search space on each dimension, has several advantages:

- It does not have scalability problems. This means that performance (spreading of solutions)
- It does not require parameters, as do other approaches such as ε-MOEA, PAES, etc.
- It scales up without complications. For any size of functin space, MaxiMin performance is consistent.
- It has a low complexity. The MaxiMin algorithm does not require sorting, or distance computing among all individuals, etc.
- It is independent of the evolutionary search engine.
- It has a stable behavior. Usually the MaxiMin algorithm performs similarly for the same parameters.

- Knowledge about minimum and maximum function values are not needed.

According to the experiments developed, there are very good approaches such as ε-MOEA to achieve diversity in MOEAs. Although, it was designed to perform well in a variety of problems, for non-connected or non-typical problems (e.g. ZDT3, DTLZ5) it cannot achieve a good performance. In problems such as DTLZ2, spacing of solutions when using ε-MOEA depends on the curvature of the Pareto front, due to the inherent hyper-grid that is built. Thus, ε-MOEA allows us to tackle only certain solutions or points in the objective space. Another disadvantage of this approach is the tuning of the ε vector.

The MaxiMin algorithm also has some disadvantages such as population size dependence, and that it does not actually give information about which individual is better than another in a less crowding sense.

References

[1] Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley and Sons, Chichester, England (2001)

[2] Horn, J., Nafpliotis, N., Goldberg, D.: A Niched pareto Genetic Algorithm for Multiobjective Optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation. Volume 1., IEEE Service Center (1994) 82–87

[3] Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. Technical Report 2000001, Indian Institute of Technology, Kanpur, India (2000)

[4] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)

[5] Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front using the Pareto Archived Evolution Strategy. MIT Press Journals **8** (2000) 149–172

[6] Abbass, H., Sarker, R., Newton, C.: PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems. In: Proceedings of the 2001 Congress on Evolutionary Computation, CEC 2001. Volume 2., IEEE Service Center (2002) 971–978

[7] Robic, T., Filipic, B.: DEMO: Differential Evolution for Multiobjective Optimization. In: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization. EMO 2005, Lecture Notes in Computer Science Vol. 3410 (2005) 707–720

[8] Pires, E.J.S., de Moura Oliveira, P.B., Machado, J.A.T.: Multi-objective Maximin Sorting Scheme. In: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization. EMO 2005, Lecture Notes in Computer Science Vol. 3410 (2005) 165–175

[9] Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA (1995)

[10] Deb, K., Mohan, M., Mishra, S.: A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. Technical Report 2003002, Indian Institute of Technology, Kanpur, India (2003)

[11] Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evolutionary Computation **10** (2002) 263–282

[12] Deb, K., Thiele, L., Laumanns, M., Ziztler, E.: Scalable Multi-objective Optimization Test Problems. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002. Volume 1., IEEE Service Center (2002) 825–830