Carlos A. Coello Coello
Arturo Hernández Aguirre
Eckart Zitzler (Eds.)

LNCS 3410

Evolutionary Multi-Criterion Optimization

Third International Conference, EMO 2005
Guanajuato, Mexico, March 2005
Proceedings

Multi-objective MaxiMin Sorting Scheme

E. J. Solteiro Pires[1], P. B. de Moura Oliveira[2], and J. A. Tenreiro Machado[3]

[1] Universidade de Trás-os-Montes e Alto Douro, Dep. de Engenharia Electrotécnica,
Quinta de Prados, 5000–911 Vila Real, Portugal
http://www.utad.pt/~epires
[2] Universidade de Trás-os-Montes e Alto Douro, CETAV,
Quinta de Prados, 5000–911 Vila Real, Portugal
{epires,oliveira}@utad.pt
http://www.utad.pt/~oliveira
[3] Instituto Superior de Engenharia do Porto, Dep. de Engenharia Electrotécnica,
Rua Dr. António Bernadino de Almeida, 4200-072 Porto, Portugal
jtm@dee.isep.ipp.pt
http://www.dee.isep.ipp.pt/~jtm

Abstract. Obtaining a well distributed non-dominated Pareto front is one of the key issues in multi-objective optimization algorithms. This paper proposes a new variant for the elitist selection operator to the NSGA-II algorithm, which promotes well distributed non-dominated fronts. The basic idea is to replace the crowding distance method by a maximin technique. The proposed technique is deployed in well known test functions and compared with the crowding distance method used in the NSGA-II algorithm. This comparison is performed in terms of achieved front solutions distribution by using distance performance indices.

1 Introduction

Multi-objective techniques using genetic algorithms (GAs) have been increasing in relevance as a research area. In 1989, Goldberg [1] suggested the use of a GA to solve multi-objective problems and since then other investigators have been developing new methods, such as multi-objective genetic algorithm (MOGA) [2], non-dominated sorted genetic algorithm (NSGA) [3] and niched Pareto genetic algorithm (NPGA) [4], among many other variants [5].

Achieving a well-spread and well-diverse Pareto solution front can be a time consuming computational problem, associated with multi-objective evolutionary algorithms (MOEAs). A good background review about the use of bounded archive population in MOEAs can be found in [6]. The computational complexity is directly related with the level of diversity and distribution the MOEAs aims to obtain. The higher this level, the larger computational power will be required. Indeed, as it was stated in [7] "For example, NSGA-II uses a crowding approach which has a computational complexity of $O(N \log N)$, where N is the population size. On the other hand, SPEA uses a clustering approach which has computational complexity of $O(N^3)$". Also it was found that while for two objectives problems the difference in terms of the achieved solution diversity with NSGA-II and SPEA is not significant, for three objectives problem the SPEA proved to be clearly better, but at the expensive of a higher computational load.

C. A. Coello Coello et al. (Eds.): EMO 2005, LNCS 3410, pp. 165–175, 2005.

Thus, new computational schemes which can find good distributed Pareto fronts with reasonable computational effort are a highly desired feature.

Maximin is a well known method used in classic multi-attribute problems [8] and in the game theory [9, 10]. Recently Balling [11] proposed a multi-objective optimization technique based on a fitness function derived from using the maximin strategy [9] and Li [12] used the maximin fitness in a particle swarm multi-objective optimizer.

Bearing these ideas in mind, this paper, proposes a sorting scheme to select the best solutions in order to promote its diversity within MOEAs. In each generation, the achieved set is initially formed by the best solutions for each objective. Then, the achieved population is completed, one solution at a time, by the maximum of the minimal norm between a solution and the set of solutions already selected.

The article is organized as follows: section 2 describes the proposed method. Section 3 presents the MOEA settings, test functions and performance indices used for performance comparison. Section 4 shows the results and analysis of experiments carried out with maximin sorting scheme. Finally, section 5 outlines the main conclusions.

2 MaxiMin Sorting Scheme

This section presents the maximin sorting algorithm to render the following generation, having a good solution distribution.

The problem addressed by the proposed sorting scheme (maximin) is the selection of the best distributed N solutions from an original population with size M ($M > N$). As it is well known, this is a very useful feature in elitism MOEAs when it is necessary to choose the solutions which passes to the following generation. Similarly to the NSGA-II, the proposed algorithm is based on non-dominated fronts [13]. However, when the last allowed front is being considered, and there are more solutions in the last front than the remaining slots in the population, a maximin function is called to select the last solutions, in spite of using the crowding distance.

The main idea behind the maximin sorting scheme is to select the solutions in order to decrease the large gap areas existing in the already selected population. For example, let us consider the non-dominated solutions in figure 1. Initially the extreme solutions are selected $S \equiv \{a, b\}$. Then, solution c is selected because it has the greater distance to the set S. After that, solutions d and e are selected into the set $S \equiv \{a, b, c\}$, for the same reason. The process is repeated until the S set is completed.

The maximin sorting scheme is depicted in algorithm 1, assuming a minimization problem. Table 1 presents the notation used in algorithm 1. In each generation the new population is merged with the archive population, resulting in set T, from with the new archive is obtained by applying the proposed algorithm (lines 0–1). After that, the algorithm selects the best solutions for each objective (lines 2-4) into the S set. Then the non-dominated front is removed from the non-selected population T into S set until the last allowed front being considered does not fit into the S set (lines 5-9). Therefore, the square of the distance, c_i (1a), between each non-dominated solution and the set of solutions already selected, S, is evaluated, selecting the solution whose distance to the set is greater (1b). Every time a solution enter to the set S the cost c_i, of non-dominated solutions, is reevaluated (lines 10-19). This process ends when the S set is completed.

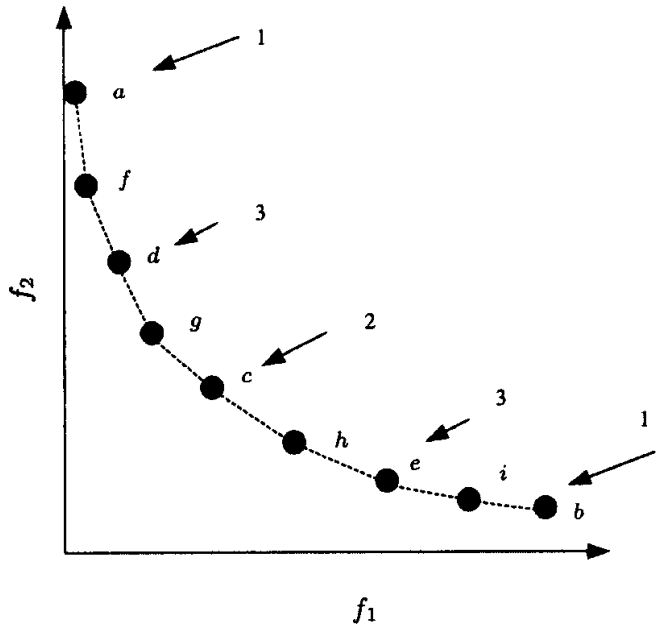


Fig. 1. Non-dominated front solutions

Table 1. Description of variables and functions used in maximin algorithm

	Description
D	Offspring population
A, T	Auxiliary populations
S	Archive or Parent population
k	Actual maximum
a_{c_i}	Minimum euclidian norm between solution i and set S
getMin(X,i)	Remove from set X the solution whose objective i is minimal
getMaxCi(X)	remove the solution whose norm value c_i is maximum
getND(X)	Remove all non-dominated solutions from set X

$$c_{a_j} = \min_{s_i \in S} \| f_{a_j} - f_{s_i} \| \tag{1a}$$

$$S = S \cup \max_{a_j \in A} a_j \tag{1b}$$

We verify that the different bulk of this algorithm, in relation to the crowding method (lines 10–19), requires at most $O(N^2)$ computations, which is worst than the $O(N \log N)$ computations required by the crowding distance scheme.

3 Test Problems and Performance Indices

In this section the maximin sorting scheme performance is compared to the crowding distance performance used in the NSGA-II. In order to study the diversity obtained by the two algorithms, they will be studied using some functions and performance indices.

Algorithm 1 MaxiMin Sorting Scheme

0: $T = S \cup D$
1: $S = \varnothing$
2: *for* $i = 1$ *to* N_{obj} *do*
3: $S = S \cup \text{getMin}(T, i)$
4: *end for*
5: $A = \text{getND}(T)$
6: *while* ($\#S + \#A \leq pop_{\text{size}}$)
7: $S = S \cup A$
8: $A = \text{getND}(T)$
9: *end while*
10: *for* $j = 1$ *to* $\#A$ *do*
11: $c_{a_j} = \min_{s_i \in S}\{\|f_{a_j} - f_{s_i}\|\}$
12: *end for*
13: *while* ($\#S < pop_{\text{size}}$)
14: $k = \text{getMaxCi}(A)$
15: $S = S \cup k$
16: *for* $l = 1$ *to* $\#A$ *do*
17: $c_{a_l} = \min\{\|f_{a_l} - f_k\|, c_{a_l}\}$
18: *end for*
19: *end while*

3.1 Genetic Settings

To compare the results a gray binary MOEA is used, with 24 bits by parameter, crossover probability $p_c = 0.8$, mutation provability $p_m = 1/l$ (l is the string length), niching parameter $\sigma_{\text{share}} = 0.5$, dominance pressure of 100%, population size of 100 strings. Moreover, the algorithm is executed in 1000 generations per test.

3.2 Functions Used

The test bed is formed by a total of five functions. Three functions $\{F_1, F_2, F_3\}$, proposed by Zitzler *et al.* [14] as ZDT1 (2), ZDT2 (3) and ZDT3 (4), each with two objectives $\{f_1, f_2\}$. Two other functions $\{F_4, F_5\}$, used by Deb *et al.* in [15] as DTLZ2 (5) and DTLZ4, each with three objectives $\{f_1, f_2, f_3\}$.

$$F_1 = \begin{cases} f_1(X) & = x_1 \\ g(X) & = 1 + 9 \sum_{i=2}^{m} \frac{x_i}{m-1} \\ h(f_1, g) & = 1 - \sqrt{\frac{f_1}{g}} \\ f_2(X) & = g(X) h(f_1, g) \end{cases} \tag{2}$$

$$F_2 = \begin{cases} f_1(X) & = x_1 \\ g(X) & = 1 + 9 \sum_{i=2}^{m} \frac{x_i}{m-1} \\ h(f_1, g) & = 1 - \left(\frac{f_1}{g}\right)^2 \\ f_2(X) & = g(X) h(f_1, g) \end{cases} \tag{3}$$

$$F_3 = \begin{cases} f_1(X) &= x_1 \\ g(X) &= 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1} \\ h(f_1, g) &= 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1) \\ f_2(X) &= g(X)h(f_1, g) \end{cases} \tag{4}$$

$$F_4 = \begin{cases} f_1(X) = [1 + g(X)]\cos(x_1\pi/2)\cos(x_2\pi/2) \\ f_2(X) = [1 + g(X)]\cos(x_1\pi/2)\sin(x_2\pi/2) \\ f_3(X) = [1 + g(X)]\sin(x_1\pi/2) \\ g(X) = 1 + 9\sum_{i=3}^{m}(x_i - 0.5)^2 \end{cases} \tag{5}$$

Function F_5 is similar to F_4 but it has a meta-variable mapping $x_i \rightarrow x_i^{\alpha}$ with $\alpha = 100$. The vector X is formed by m parameters, $m_j = \{30, 30, 30, 12, 12\}, j = 1, \ldots, 5$, for $F_j = \{F_1, F_2, F_3, F_4, F_5\}$, with each value $x_i \in [0, 1], i = 1, \ldots, m_j$.

3.3 Performance Indices

To study the solution diversity the following indices are used: the spacing index (SP) [16] (6), the distance-based distribution index (Δ') [17] (7) and the minimal distance graph (MDG) (8). In all indices $\overline{d}$ is the average of d_i distances.

$$SP(S) = \sqrt{\frac{1}{\#S - 1}\sum_{i=1}^{\#S}(d_i - \overline{d})^2} \tag{6a}$$

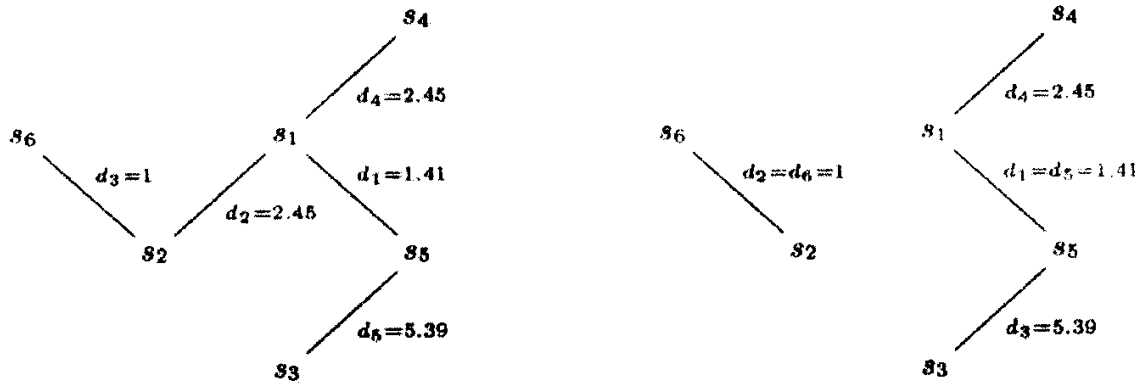
$$d_i = \min_{s_k \in S \wedge s_k \neq s_i} \sum_{m=1}^{M} |f_m(s_i) - f_m(s_k)| \tag{6b}$$

The distance d_i of Δ' index is evaluated by euclidian distance between consecutive solutions in S. Therefore, this index can only be used in two objective problems.

$$\Delta'(S) = \sum_{i=1}^{\#S-1} \frac{|d_i - \overline{d}|}{\#S - 1} \tag{7}$$

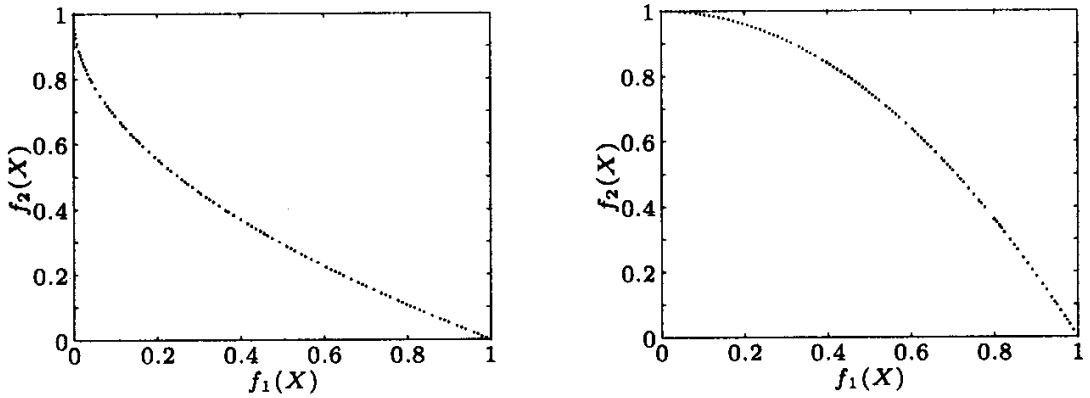
The MDG performance index is calculated based on the minimal distances, d_i, that links all the non-dominated solutions of the population. For example, considering the solutions $\{s_1 = (3, 5, 7), s_2 = (2, 6, 5), s_3 = (7, 7, 2), s_4 = (5, 4, 8), s_5 = (4, 5, 6), s_6 = (1, 6, 5)\}$ the minimal distances that link all the solutions are: $\{\overline{s_1 s_2}, \overline{s_1 s_4}, \overline{s_1 s_5}, \overline{s_2 s_6}, \overline{s_3 s_5}\}$, as it is illustrated in figure 2(a). The MDG index has some advantage over the SP index because it never uses the same distance more than once and relates all solutions together (see figure 2). Moreover, the MDG index can be used in any dimensional space in spite of the Δ' index.

$$MDG(S) = \sqrt{\frac{1}{\#S - 2}\sum_{i=1}^{\#S-1}(d_i - \overline{d})^2} \tag{8}$$



(a) Minimal Distance Graph (b) SP index distances

Fig. 2. Distances used by MDG and SP indices



(a) F_1 optimization front (b) F_2 optimization front

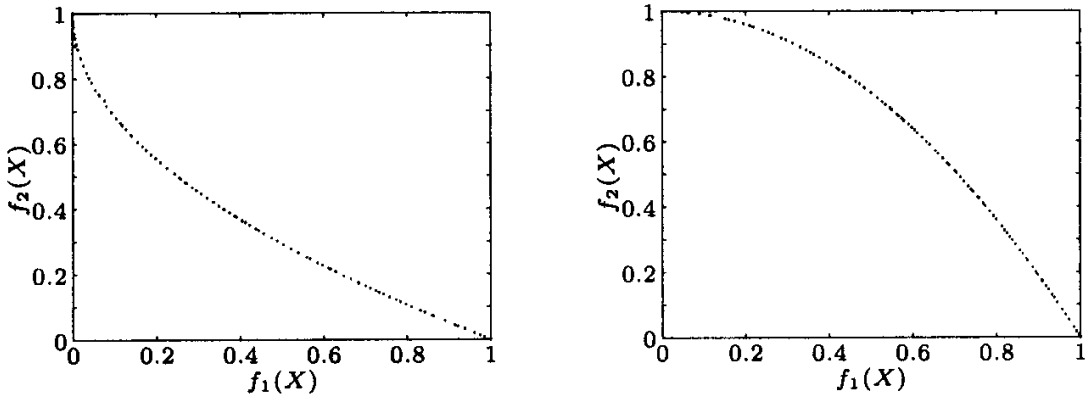
Fig. 3. F_1 and F_2 optimization with maximin sorting scheme

4 Simulation Results

To compare the maximin sorting scheme and the crowding distance technique several simulations were conducted involving $n = 101$ runs each. For the results analysis are considered the median, average, standard deviation, and the minimum and the maximal solutions obtained.

Table 2. F_1 test results

	Maximin sorting scheme			Crowding distance		
	SP	Δ'	MDG	SP	Δ'	MDG
Median	**0.005**	**0.067**	**0.005**	0.008	0.075	0.007
Average	**0.005**	**0.067**	**0.005**	0.008	0.075	0.007
Std dev	0.000	0.001	0.000	0.001	0.003	0.001
Min	**0.004**	**0.064**	**0.005**	0.006	0.068	0.006
Max	0.006	0.069	0.006	0.009	0.083	0.009



(a) F_1 optimization (b) F_2 optimization

Fig. 4. F_1 and F_2 optimization with crowding distance

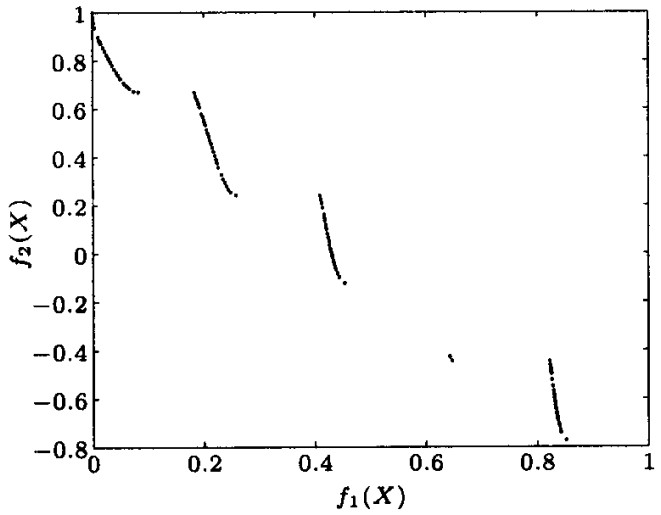
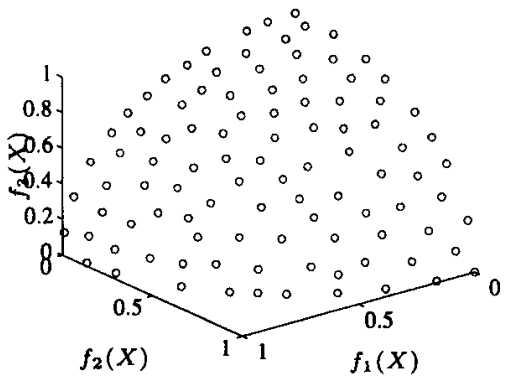


Fig. 5. A F_3 optimization front with maximin sorting scheme

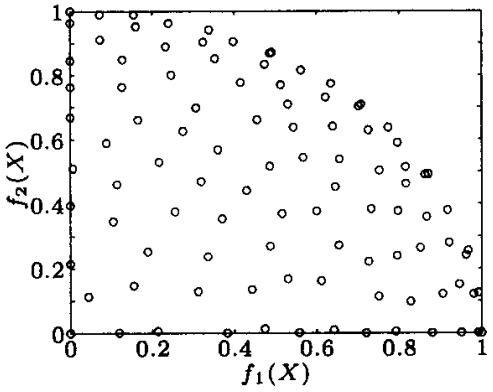
Table 3. F_2 test results

	Maximin sorting scheme			Crowding distance		
	SP	Δ'	MDG	SP	Δ'	MDG
Median	**0.005**	**0.067**	**0.005**	0.008	0.075	0.007
Average	**0.005**	**0.067**	**0.005**	0.008	0.075	0.007
Std dev	0.000	0.001	0.000	0.001	0.002	0.001
Min	**0.004**	**0.064**	**0.005**	0.006	0.066	0.006
Max	0.006	0.070	0.006	0.010	0.086	0.009

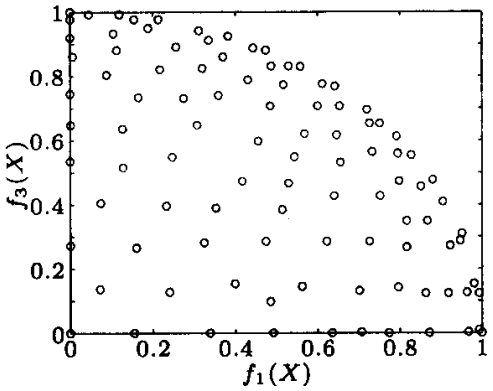
For the F_1 and F_2 optimizations both schemes obtain good diversity (figures 3 and 4). Nevertheless, the maximin sorting scheme achieved better performance indices (tables 2 and 3), reaching better results, even though, the worst results obtained with the maximin sorting scheme are in the same range of the best results with the crowding distance. Due to the proximity of achieved indices values, it is difficult to conclude about the



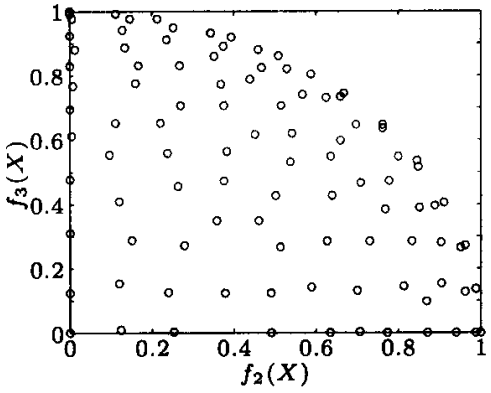
(a) F_4 optimization front



(b) $f_1 \times f_2$ plane projection



(c) $f_1 \times f_3$ plane projection



(d) $f_2 \times f_3$ plane projection

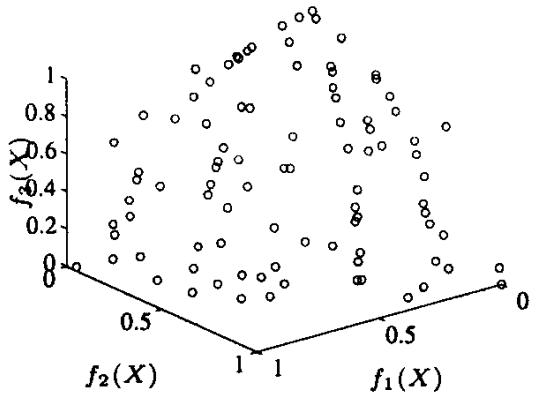
Fig. 6. F_4 optimization with maximin sorting scheme

Table 4. F_3 test results

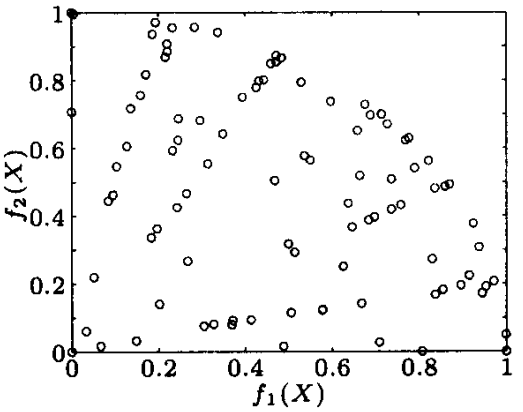
	Maximin sorting scheme			Crowding distance		
	SP	Δ'	MDG	SP	Δ'	MDG
Median	0.014	0.122	0.039	**0.008**	**0.121**	**0.034**
Average	0.015	**0.121**	0.037	**0.011**	**0.121**	**0.035**
Std dev	0.011	0.006	0.006	0.007	0.005	0.006
Min	**0.004**	**0.109**	**0.027**	0.006	0.111	**0.027**
Max	0.052	0.138	0.049	0.052	0.138	0.054

superiority of any algorithm. Thus, this results were subjected to the Mann-Whitney test which shows that the proposed method is significantly better for $p > 0.05$.

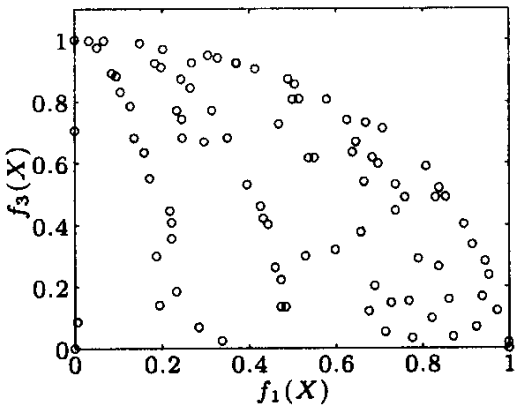
For F_3 the crowding distance method presents better performance indices (see table 4). However, the maximin sorting scheme reaches the best results in terms of the achieved distribution. The reason for this is due to the maximin algorithm has difficulty in converging for some tests to the entire non-dominate fronts (see figure 5).



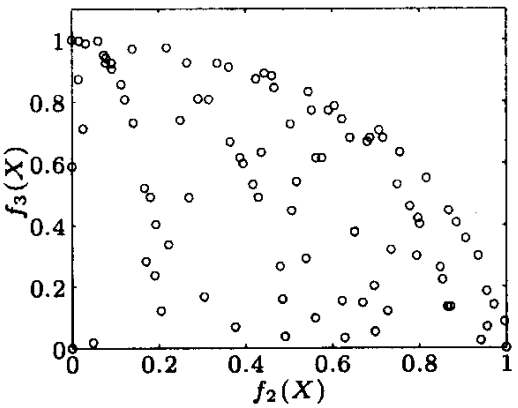
(a) F_4 optimization front



(b) $f_1 \times f_2$ plane projection

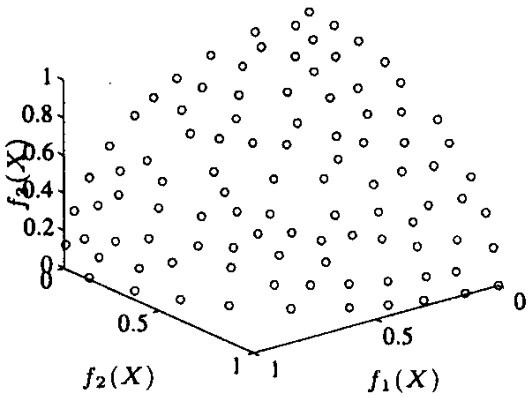


(c) $f_1 \times f_3$ plane projection

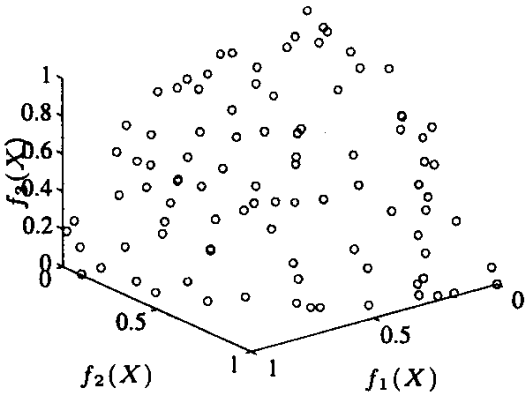


(d) $f_2 \times f_3$ plane projection

Fig. 7. F_4 optimization with crowding distance



(a) Maximin sorting scheme



(b) Crowding distance

Fig. 8. F_5 optimization

Table 5. F_4 test results

	Maximin sorting scheme			Crowding distance		
	SP	Δ'	MDG	SP	Δ'	MDG
Median	**0.028**	–	**0.017**	0.060	–	0.048
Average	**0.028**	–	**0.017**	0.060	–	0.047
Std dev	0.003	–	0.002	0.005	–	0.003
Min	**0.021**	–	**0.013**	0.044	–	0.040
Max	0.038	–	0.025	0.073	–	0.055

Table 6. F_5 test results

	Maximin sorting scheme			Crowding distance		
	SP	Δ'	MDG	SP	Δ'	MDG
Median	**0.029**	–	**0.019**	0.060	–	0.048
Average	**0.029**	–	**0.020**	0.061	–	0.049
Std dev	0.004	–	0.004	0.006	–	0.003
Min	**0.020**	–	**0.015**	0.049	–	0.043
Max	0.048	–	0.043	0.078	–	0.060

For F_4 and F_5 the maximin scheme leads to much better results, as can be seen by figures 6 to 8 and tables 5 and 6. In these simulations the worst test case obtained by the maximin scheme is superior to the best case obtained by the crowding distance method.

5 Conclusions and Further Work

A maximin sorting algorithm to select non-dominated solutions which complete the new population was proposed. The new algorithm was integrated in a MOEA in order to test its capacity to generate well distributed non-dominated Pareto fronts. This technique was deployed in well known test functions and the results compared with the ones obtained by using the NSGA-II crowding distance method. The analysis was made considering some distance performance indices and points out that the algorithm reaches a set of non-dominated solutions, along the Pareto front, with a good spread, outperforming the crowding method in most of the test functions, particularly when involving three objectives. The proposed algorithm is more demanding in terms of computational load. However, the improvement gained in terms of diversity compensates this cost.

In the nearby future the proposed method will be compared with other techniques such as the C-NSGAII, ϵ-MOEA, and SPEA. These tests will incorporate more performance criteria and access the execution computational time.

Acknowledgment

This work is partially supported by the grant Prodep III (2/5.3/2001) from FSE.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison – Wesley (1989)
2. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multi-objective optimization. Evolutionary Computation Journal **3** (1995) 1–16
3. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley-Interscience Series in Systems and Optimization. (2001)
4. Horn, J., Nafploitis, N., Goldberg, D.: A niched pareto genetic algorithm for multi-objective optimization, Proceedings of the First IEEE Conference on Evolutionary Computation (1994) 82–87
5. Coello, C., Carlos, A.: A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information Systems **1** (1999) 269–308
6. Knowles, J.D., Corne, D.W., Fleischer, M.: Bounded archiving using the lebesgue measure. In: CEC – Congress on Evolutionary Computation, Canberra, Australia (2003)
7. Kalyanmoy Deb, Manikanth Mohan, S.M.: Towards a quick computation of well-spread pareto-optimal solutions. In Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L., eds.: Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 8-11, 2003, Proceedings. Volume 2632 of Lecture Notes in Computer Science., Springer (2003) 222–236
8. K. Paul Yoon, C.L.H.: Multiple Attribute Decision Making : An Introduction (Quantitative Applications in the Social Sciences). SAGE Publications (1995)
9. Luce, R.D., Raiffa, H.: Introduction and Critical Survey. John Wiley & Sons, Inc, New York (1957)
10. Rawls, J.: A Theory of Justice. Oxford University Press, London (1971)
11. Balling, R.: The maximin fitness function; multi-objective city and regional planning. In Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L., eds.: Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 8-11, 2003, Proceedings. Volume 2632 of Lecture Notes in Computer Science., Springer (2003) 1–15
12. Li, X.: Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function. In Deb, K.e.a., ed.: Proceeding of Genetic and Evolutionary Computation Conference 2004 (GECCO'04). Volume LNCS 3102 of Lecture Notes in Computer Science, Springer-Verlag., Seattle, USA (2004) 117–128
13. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, LTD (2001)
14. Zitzler, E., D.K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8** (2000) 173–195
15. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In Fogel, D.B., El-Sharkawi, M.A., Yao, X., Greenwood, G., Iba, H., Marrow, P., Shackleton, M., eds.: Proceedings of the 2002 Congress on Evolutionary Computation CEC2002, IEEE Press (2002) 825–830
16. Schott, J.R.: Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, Massachusetts (1995)
17. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., eds.: Parallel Problem Solving from Nature – PPSN VI. Volume 1917 of LNCS., Berlin, Springer (2000) 849–858