An Adaptive Sharing Elitist Evolution Strategy for Multiobjective Optimization

Lino Costa lac@dps.uminho.pt
Departamento de Produção e Sistemas, Universidade do Minho, Campus de Gualtar, 4710 Braga, Portugal

Pedro Oliveira pno@dps.uminho.pt
Departamento de Produção e Sistemas, Universidade do Minho, Campus de Gualtar, 4710 Braga, Portugal

Abstract
Almost all approaches to multiobjective optimization are based on Genetic Algorithms (GAs), and implementations based on Evolution Strategies (ESs) are very rare. Thus, it is crucial to investigate how ESs can be extended to multiobjective optimization, since they have, in the past, proven to be powerful single objective optimizers. In this paper, we present a new approach to multiobjective optimization, based on ESs. We call this approach the Multiobjective Elitist Evolution Strategy (MEES) as it incorporates several mechanisms, like elitism, that improve its performance. When compared with other algorithms, MEES shows very promising results in terms of performance.

Keywords
Evolution Strategies, Multiobjective Optimization, Elitism.

1 Introduction

Solving multiobjective engineering problems is a very difficult task since, in general, the objectives conflict across a high-dimensional problem space. Thus, the interaction between the multiple objectives gives rise to a set of compromise solutions, known as the Pareto-optimal solutions. Over the past decade, the application of evolutionary algorithms to multiobjective optimization has been investigated by several authors, such as Schaffer (1985), Fonseca and Fleming (1993), Horn et al. (1994), Srinivas and Deb (1995) and Zitzler and Thiele (1999). Almost all approaches are based on Genetic Algorithms (GAs) (Goldberg, 1989) which were extended in order to track multiobjective problems. Implementations based on Evolution Strategies (ESs) (Rechenberg, 1994) are rare, such as the algorithm proposed by Kursawe (1990) and Knowles and Corne (2000). However, the latter approach does not use some traditional features of ESs, namely, the real coding of decision variables and the adaptation of step sizes for mutation. Thus, it is crucial to investigate how to extend ESs to multiobjective optimization, since, in the past, they prove to be powerful single objective optimizers. This paper presents a new approach to multiobjective optimization based on ESs. In the new algorithm, an effort was made to maintain the main features of traditional ESs as single objective optimizers. In particular, real representation is used, together with self adaptation of step sizes and recombination. Several mechanisms, like elitism, have been introduced in order to improve the performance of the algorithm, as previously suggested by Zitzler et al. (2000) and Van Veldhuizen and Lamont (2000). Using a new adaptive sharing scheme

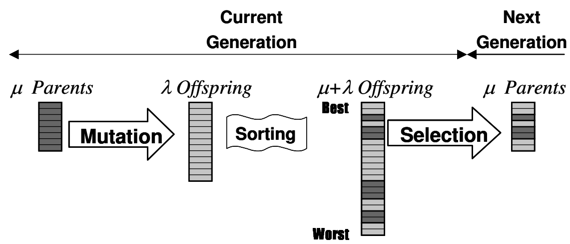

Figure 1: The $(\mu + \lambda)$ Evolution Strategy

together with geometric selection has proven to be useful in controlling and guiding the search.

In section 2, a short introduction to ES is presented. Section 3 describes the implementation of the Multiobjective Elitist Evolution Strategy (MEES). Next, the results of the application to several problems are presented, as well as some comparisons with other multiobjective algorithms. Finally, some conclusions and future work are addressed.

2 Evolution Strategies

Evolution Strategies are search procedures that mimic the natural evolution of a species in natural systems. They were first reported by Rechenberg (1964, 1994) and later by Schwefel (1981,1995). ESs were developed to solve single objective optimization problems. Like GAs, they work with populations of candidate solutions, requiring only data based on the objective function and constraints, and no derivatives or other auxiliary knowledge. However, ESs work directly with the real representation of the decision variables and the transitions rules are deterministic (in particular, selection is a deterministic procedure). Traditionally, the search for new points was based on one single operator, the mutation operator. However, more recently, a recombination operator was introduced. One of the most important features of ESs is that they use adaptive step sizes for mutation.

Figure 1 illustrates the $(\mu + \lambda)$-ES. The (μ, λ)-ES is similar, differing basically on the selection procedure. Thus, in the $(\mu + \lambda)$-ES, at a given generation, there are μ parents, and λ offspring generated by mutation. For the (μ, λ)-ES, μ must be inferior to λ whereas, for the $(\mu + \lambda)$-ES, λ can be any positive integer. Mutation creates new points by adding random normal distributed quantities with mean zero and variance σ_i^2. It is important to note that, for each decision variable i, an individual standard deviation σ_i is used (controlling the step sizes). Then, the $\mu + \lambda$ members are sorted according to their objective function values. Finally, the best μ of all the $\mu + \lambda$ members become the parents of the next generation (i.e., the selection takes place between the $\mu + \lambda$ members). On the other hand, in (μ, λ)-ES, the best μ of the λ members generated become the parents of the next generation (i.e., the selection takes place between the λ members).

For many problems, $\lambda/\mu \approx 7$ is suggested (Schwefel, 1995). For single objective optimization problems, the typical initial values for the standard deviations σ_i can be expressed by equation (1), where Δx is a rough measure of the distance from the optimum and n is the dimension of the problem (the number of decision variables of the problem).

$$\sigma_i^{(0)} = \frac{\Delta x}{\sqrt{n}} \tag{1}$$

During the search, the step sizes for mutation are adapted. Several self-adaptation schemes are possible. One possibility is to actualize the standard deviations σ_i (for each decision variable) according to the equation (Schwefel, 1995):

$$\sigma_i^{(g+1)} = \sigma_i^{(g)} e^{z_i} e^{z} \tag{2}$$

where g is the generation counter, $z_i \sim N(0, \Delta\sigma^2)$, $z \sim N(0, \Delta\sigma'^2)$ and $\Delta\sigma$ and $\Delta\sigma'$ are parameters of the algorithm. Typically, acceptable values for $\Delta\sigma$ and $\Delta\sigma'$ are $1/\sqrt{n}$.

Schwefel (1995) reported a remarkable acceleration in the search process, as well as the facilitation of self-adaptation of parameters by introducing a recombination operator. Basically, the recombination operator consists of, before mutation, recombining a set of chosen parents to find a new solution. A given number ρ $(1 \leq \rho \leq \mu)$ of parents are randomly chosen for recombination. When $\rho = 1$ then there is no recombination. Thus, the nomenclature for ESs can now be extended, and ESs with recombination are usually referred to as $(\mu/\rho + \lambda)$-ES or $(\mu/\rho, \lambda)$-ES. Two types of recombination are, mainly, considered: intermediate and discrete recombination.

In the intermediate recombination, the components of the offspring are obtained by calculating the average of the corresponding components of parents (randomly selected from the population). Thus, for ρ chosen parents (randomly selected from population), the offspring x_{Int_rec} is given by:

$$x_{Int\_rec} = \frac{1}{\rho}\sum_{i=1}^{\rho} x_i = \left[\begin{array}{ccc} \frac{1}{\rho}\sum_{i=1}^{\rho} x_{i,1} & \cdots & \frac{1}{\rho}\sum_{i=1}^{\rho} x_{i,n} \end{array}\right]^T \tag{3}$$

If $\rho = \mu$, this procedure tends to generate recombinated offspring around the centroid of the population.

In the discrete recombination, each component of the offspring is chosen from one of the ρ parents at random. Thus, for ρ chosen parents (randomly selected from population), the offspring x_{Dis_rec} is given by

$$x_{Dis\_rec} = \left[\begin{array}{ccc} x_{u_1,1} & \cdots & x_{u_n,n} \end{array}\right]^T \quad \text{with} \quad u_1 \in [1,\rho], \ldots, u_n \in [1,\rho] \tag{4}$$

where the generated random values u_i, for $i = 1, \ldots, n$, select from which of the ρ parents the value of decision variable i is copied. This procedure allows different combinations of the values of the decision variables from existing solutions in the population.

2.1 Multiobjective Optimization

In multiobjective optimization there is no single optimum, since several objectives must be considered. Thus, there are several solutions which are not comparable, usually referred to as Pareto-optimal solutions. A multiobjective minimization problem with n variables and m objectives can be formulated as, without loss of generality, $\min y = f(x) = (f_1(x), ..., f_m(x))$ where $x = (x_1, ..., x_n) \in X$ and $y = (y_1, ..., y_m) \in Y$. For a multiobjective minimization problem, a solution a is said to dominate a solution b, if and only if, $\forall j \in \{1, ..., m\} : f_j(a) \leq f_j(b)$ and $\exists l \in \{1, ..., m\} : f_l(a) < f_l(b)$. A solution a is said to be non-dominated regarding a set $X' \subseteq X$ if and only if, there is no solution in X' which dominates a. The solution a is Pareto-optimal if and only if a is non-dominated regarding X.

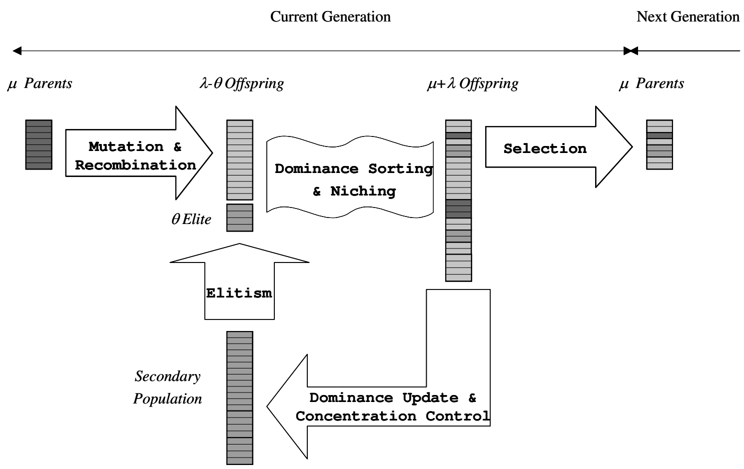

Figure 2: The Multiobjective Elitist Evolution Strategy

Therefore, each solution can either be represented on the variable space (X) or on the objective space (Y). The set of all non-dominated solutions constitutes the Pareto-optimal set. Thus, the main features for a multiobjective algorithm is to find a good and balanced approximation to the Pareto-optimal set. As stated in the Introduction, early algorithms were based on Genetic Algorithms, while approaches based on evolution strategies have been rare.

The algorithm described in the following section maintains the main features of ESs and introduces new mechanisms like elitism and sharing, in order to improve its performance for multiobjective optimization, such as step size adaptation as well as real coded representation.

3 A Multiobjective Elitist Evolution Strategy

Figure 2 illustrates the Multiobjective Elitist Evolution Strategy (MEES). This approach differs from conventional ESs with respect to the selection operator emphasizing the non-domination of solutions. Non-domination is tested at each generation in the selection phase, thus defining an approximation to the Pareto optimal set. On the other hand, a sharing method is used to distribute the solutions in the population over the Pareto-optimal region. The usual deterministic selection was also modified in order to track multiobjective optimization. The real representation of the decision variables, mutation and recombination operators remain as usual. The step sizes for mutation were adapted with a non-isotropic self-adaptation scheme as in equation (2).

3.1 Fitness Assignment

The fitness assignment follows the non-domination sorting algorithm proposed by Srinivas and Deb (1995). Thus, for each generation, all non-dominated solutions of the λ or $\mu + \lambda$ solutions will constitute the 1st front. To these solutions a fitness value of 1 is assigned. In order to maintain diversity, a sharing scheme is then applied to the fitness values of these solutions (Deb and Goldberg, 1989). Thus, the fitness value of each solution is multiplied by a quantity, called niche count, proportional to the num-

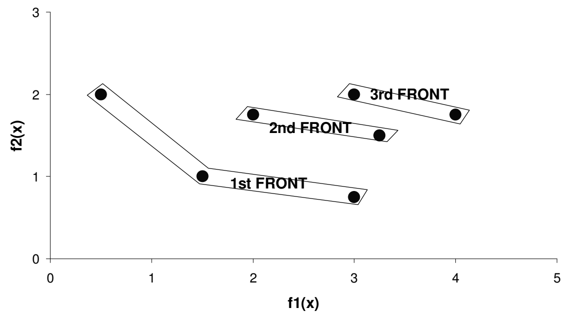

Figure 3: Distribution of Solutions in Fronts

ber of solutions having a distance inferior to a parameter, the σ_{share}. All distances are measured in the objective space. Thereafter, the solutions of the 1st front are ignored temporarily, and the remaining solutions are processed. To the second level of non-dominated solutions is assigned a fitness value equal to the worst computed fitness value from the solutions in the 1st front plus 1. Next, the fitness value of each solution in the 2nd front is multiplied by the respective niche count value. This process is repeated until all the λ or $\mu + \lambda$ solutions are assigned a fitness value. This fitness assignment process will emphasize the non-domination of solutions, since the fitness values of all solutions in the 1st front will have a value inferior to all the fitness values of solutions in the 2nd front, and so on. Moreover, the co-existence of multiple non-dominated solutions is encouraged by the sharing scheme (Figure 3).

3.2 Sharing Adaptation

Niching was pioneered by Goldberg (1989). However, there are problems with setting the niche radius. These have been identified and discussed by Deb and Goldberg (1989) and Fonseca and Fleming (1993). In particular, adaptive schemes have been proposed by Fonseca and Fleming (1998). However, these have some problems when applied in multiobjective optimization. So, a new adaptive niching scheme is presented for sharing on the objective space.

The working of the sharing mechanism depends mainly on the parameter σ_{share}. This parameter must be set carefully in order to obtain a well distributed set of solutions approximating the Pareto optimal set. The parameter σ_{share} is the maximum distance between solutions necessary to form a well distributed front. Therefore, this parameter depends on the number of desirable distinct solutions in the front and the upper and lower bounds of the solution space. Two distinct approaches for comparing solutions can be considered: solutions can be compared in the variable or objective spaces. Several arguments exist that prefer one approach (decision variable distance) to the other (objective function distance). The likely existence of multiple solutions on the decision space with the same objective function value, might favor the first approach, if it is considered that diversity on the decision space is of primary importance; however, the Pareto set is defined on the objective function values and, thus, it might be of greater importance to have diversity, i.e., a well-balanced distribution on the objective function space.

Since solutions in the population are ranked in terms of dominance defining sev-

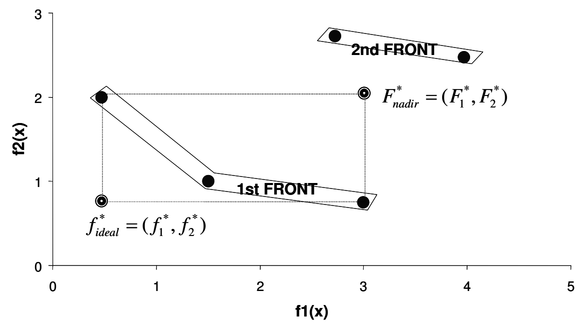

Figure 4: Extreme Solutions in Objective Space

eral fronts (Figure 3), it is convenient to consider distinct values of σ_{share} according to the number of solutions in each front. Thus, the particular values of σ_{share} calculated for each of the K fronts will take into account:

- the distribution of solutions in the front along the objective space;
- the number of solutions, n_k, in the front k.

Each front has n_k solutions and the minimum values, f_1^*, f_2^* and f_m^*, and maximum values, F_1^*, F_2^* and F_m^*, for the m objectives are known. Thus, the 'ideal solution', $f^*_{ideal} = (f_1^*, f_2^*, ..., f_m^*)$ and the 'nadir solution', $F^*_{nadir} = (F_1^*, F_2^*, ..., F_m^*)$ can be computed (Figure 4). The Euclidian distance between these two solutions can be calculated as:

$$d_{norm} = \sqrt{\sum_{j=1}^{m} (f_j^* - F_j^*)^2} \tag{5}$$

As a first approach, in order to estimate the parameter σ_{share} this distance can be divided in $n_k - 1$ parts. The estimated value for σ_{share}, σ'_{share}, will be half of this quantity (the radius) and is given by:

$$\sigma'_{share} = \begin{cases} \frac{d_{norm}}{2\sqrt[m-1]{n_k - 1}} & \text{if} \quad n_k > 1 \\ 0 & \text{if} \quad n_k = 1 \end{cases} \tag{6}$$

It is also possible to define the distances between the extreme solutions for each objective, $\overline{f}_1 = (f_1^*, F_2^*, ..., F_m^*)$, $\overline{f}_2 = (F_1^*, f_2^*, ..., F_m^*)$ and $\overline{f}_m = (F_1^*, F_2^*, ..., f_m^*)$, to the 'ideal solution':

$$d_{l,ideal} = \sqrt{\sum_{j=1, j \neq l}^{m} (f_j^* - F_j^*)^2} \tag{7}$$

The distances from the extreme solutions and the 'nadir solution' are given by:

$$d_{l,nadir} = d_{m-l,ideal} = \sqrt{\sum_{j=1, j \neq m-l}^{m} (f_j^* - F_j^*)^2} \tag{8}$$

The distances from the extreme points and the 'ideal solution' can be used to compute another estimative of σ_{share}. This extreme estimative is stronger than the one given by (5) in the sense that it tends to reduce the distance between solutions in objective space, and, for $n_k > 1$ (if $n_k = 1$ then $\sigma''_{share} = 0$), is given by:

$$\sigma''_{share} = \frac{\sum_{j=1}^{m} d_{j,ideal}}{2 \sqrt[m-1]{n_k - 1}} \tag{9}$$

These two estimations, σ'_{share} and σ''_{share}, can be combined as:

$$\sigma_{share} = \alpha\sigma'_{share} + (1-\alpha)\sigma''_{share} = \frac{\alpha d_{norm} + (1-\alpha)\sum_{j=1}^{m} d_{j,ideal}}{2 \sqrt[m-1]{n_k - 1}} \tag{10}$$

where α is a parameter. If α is 1 or 0 then the distance between solutions in objective space tends to be increased or reduced. With intermediate values of α, compromised values of σ_{share} are obtained.

3.3 Selection Operator

Two selection schemes are proposed, one simpler and one more complicated. In the simplest form, at each generation, only μ from the λ or $\mu + \lambda$ solutions are selected for the next generation. Two situations were considered:

- if the number of solutions in the 1st front, n_1, is not greater than μ, then a deterministic selection is performed;
- Otherwise, if n_1 is greater than μ, then a tournament selection is performed.

Deterministic selection consists of, after sorting the λ or $\mu + \lambda$ offspring according to their fitness values, selecting the μ best (the ones with lower fitness values). This selection is obviously similar to the traditional selection of ESs, in the sense that only the best individuals will be present on the next generation. On the other hand, when the number of solutions in the 1st front is high (greater than μ) then a selection scheme is adopted that guarantees that all non-dominated solutions have a possibility of being present in the next generation. This selection consists of, after sorting the λ or $\mu + \lambda$ offspring, performing a tournament between solutions of the 1st front. The tournament consists of picking two individuals from the offspring and then the best one is selected. However, this selection scheme has two main drawbacks:

- when $n_1 \leq \mu$, the sharing mechanism has no real effect for solutions in the 1st front;
- it is not possible to control the selection pressure in order to preserve, for instance, diversity during the search.

A new selection can be developed in order to avoid these drawbacks. The main idea is, for each front, to select distinct quantities of solutions for the next generation. The quantities of solutions for each front are chosen in such way that the number of solutions selected from the 1st front will be greater than the number of solutions selected from the 2nd front, and so on. If a geometric progression, as used in Deb and Goel (2001), is considered then the quantity of solutions Q_k for front k is given by:

$$Q_k = \mu \frac{r^{K-k}}{\left(\frac{1-r^K}{1-r}\right)} = \mu(1-r)\frac{r^{K-k}}{1-r^K} \tag{11}$$

Q_k		K									
		1	2	3	4	5	6	7	8	9	10
	1	100	67	57	53	52	51	50	50	50	50
	2		33	29	27	26	25	25	25	25	25
	3			14	13	13	13	13	13	13	13
	4				7	6	6	6	6	6	6
k	5					3	3	3	3	3	3
	6						2	2	2	2	2
	7							1	1	1	1
	8								0	0	0
	9									0	0
	10										0

Table 1: Variation of Q_k for $r = 2$

where K is the total number of fronts and r is the ratio of variation of Q_f between fronts (if r equals to 2 then Q_1 will be approximately twice Q_2 and so on). Note that, for all $K \geq 2$, if $r > 1$ then it is guaranteed that $Q_1 > Q_2$. More generally, for all $1 < k < K-1$, if $r > 1$ then $Q_k \geq Q_{k+1}$. The values of Q_k for $r = 2$, considering different values of K, are presented in Table 1.

However, it is important to adapt the value of r during the search since at the beginning of the search, the values of n_1 are usually small and if r is kept constant, the value computed for Q_1 can become too high. To solve this problem, the value of r can be adapted in the following way:

$$r = 1 + (spress - 1)\frac{n_1}{\mu} \tag{12}$$

where $spress$ ($spress > 1$) is a parameter denoting the selection pressure. It is clear that, if $n_1 \simeq \mu$ then $Q_1 \simeq spressQ_2$. When n_1 is small compared with μ then the value of r is approximately 1. This is a common situation at the beginning of the search. As n_1 increases during the search, the value of r is also increased, and tends to the value of $spress$ when $n_1 \simeq \mu$. If $n_1 > \mu$ then $r > spress$ and Q_1 will tend to μ.

After the computation of r, the quantities Q_k are calculated for all fronts. Then, two situations were considered:

- if the number of solutions in the kth front, n_k, is not greater than Q_k, then a deterministic selection is performed, i.e., all the n_k solutions of the kth front are selected;
- Otherwise, if n_k is greater than Q_k, then a tournament selection is performed between the solutions of the kth front.

As previously described, the deterministic selection consists of, after sorting the λ or $\mu + \lambda$ offspring according to their fitness values, selecting the μ best (the ones with lower fitness values). However, since n_k is inferior than Q_k, only n_k solutions are selected. In this case, the remaining solutions $Q_k - n_k$, are added to the quantity of solutions for the next front. Thus, for front $k + 1$, the quantity of solutions effectively considered Q'_{k+1}, will be $Q_{k+1} - (Q_k - n_k)$ if $Q_k - n_k > 0$ and Q_{k+1}, otherwise. If n_k is greater than Q_k, a tournament scheme is used to select, from each front k, the quantity, Q_k, of solutions for the next generation. As before, this tournament consists of picking 2 solutions from the offspring and choosing the best one. This selection scheme is illustrated by Table 2. For the situation considered in this table, the objective is to select 100 solutions from the 200 solutions present in 10 fronts. Since, it is considered that r is equal to 2, the number of solutions for fronts Q_k is given by the last column of Table 1.

k	n_k	Q_k	$Q_k - n_k$	Q'_k	Comment
1	85	50	< 0	50	50 solutions are selected from 85 by tournament
2	50	25	< 0	25	25 solutions are selected from 50 by tournament
3	12	13	1	12	12 solutions are deterministically selected
4	15	6	< 0	7	7 solutions are selected from 15 by tournament
5	12	3	< 0	3	3 solutions are selected from 12 by tournament
6	9	2	< 0	2	2 solutions are selected from 9 by tournament
7	8	1	< 0	1	1 solutions are selected from 8 by tournament
8	5	0	< 0	0	no solution is selected
9	3	0	< 0	0	no solution is selected
10	1	0	< 0	0	no solution is selected
	200			100	

Table 2: Example of selection for $r = 2$

This selection scheme, which allows the selection pressure to be controlled, will be referred as "$+_{spress}$" or "$,_{spress}$" selection. The simplest selection scheme described in Section 2 will be referred as "$+$" or "$,$" selection. Note that, when $spress$ is greater than μ, the "$+_{spress}$" or "$,_{spress}$" selection schemes tend to perform similarly to the "$+$" or "$,$" selection schemes.

Deb and Goel (2001) suggested a controlled selection for controlling the extent of elitism by a fixed user-defined parameter. The results of the application of the NSGA-II with controlled elitism to a number of test problems indicated a better convergence property than the original NSGA-II. However, in this approach the value of r is kept constant during the search. In the beginning of the search the value of r may be inadequately high. The "$+_{spress}$" or "$,_{spress}$" selection schemes avoid this inconvenience by adapting the value of r during the search according to the number of solutions in the first front. This adaptation scheme is intended to prevent premature convergence.

3.4 Elitist Scheme

The elitist technique is based on a separate population, the secondary population (SP) composed of all (or a part of) potential Pareto optimal solutions found so far during the search process. In this sense, SP is completely independent of the main population and, at the end of the entire search, it contains the set of all non-dominated solutions generated so far. It should be noted that $(\mu + \lambda)$-ES is inherently elitist.

A parameter θ is introduced in order to control the elitism level. This parameter states the maximum number of non-dominated solutions of SP, the so-called elite, that will be introduced in the main population. These non-dominated solutions will effectively participate in the search process. If the number of solutions in SP (n_{SP}) is greater or equal than θ, then θ non-dominated solutions are randomly selected from SP to constitute the elite. Otherwise, only n_{SP} non-dominated solutions are selected from SP to constitute the elite. In the latter case, the elite will only have n_{SP} members.

In its simplest form, for all generations, the new potential Pareto optimal solutions found are stored in SP. The SP update implies the determination of Pareto optimality of all solution stored so far, in order to eliminate those that became dominated. As the size of SP grows, the time to complete this operation may become significant. So, in order to prevent growing computation times, in general, a maximum SP size is imposed. Thus, the algorithm consists of, for all generations, storing, in SP, each Pareto optimal solution x_{nd} found in the main population if:

1. all solutions in SP are different from x_{nd};

2. none of the solutions in SP dominates x_{nd}.

Next, all solutions in SP that became dominated are eliminated. As mentioned, as the size of SP increases, the execution time and memory requirements also increase. So, it is convenient to keep relatively small sizes of SP. In this sense, the previous algorithm can be modified accordingly. A new parameter d is introduced, stating the minimum desirable distance in objective space between potential Pareto optimal solutions in SP. So, the algorithm is modified by introducing the following step:

3. the distance from x_{nd} to any of the non-dominated solutions in SP is greater than d (Euclidean distance measured on objective space).

Thus, parameter d defines a region of indifference with respect to non-domination. The order by which the candidate solutions are inserted in SP, influences the approximation to the Pareto front. However, this influence can be overcome by successive runs with decreasing values of parameter d, thus defining approximations to the front with increasing precision.

3.5 Time Complexity

The MEES algorithm uses a parent population of μ individuals and, at each generation, a population of λ offspring is generated. Moreover, a secondary population (SP) of maximum size n_{SP} archiving non dominated solutions is maintained. If an $(\mu/\rho + \lambda)$-ES is considered then, at each generation, since a non-dominated sorting mechanism is applied to $N = \mu + \lambda$ individuals, the number of comparisons required is $O(mN^3)$. The niching mechanism (in the objective space) applied to the N individuals of the parent and offspring populations requires in the worst case $O(mN^2)$ comparisons. The SP update implies $O(n_{SP}mN)$ comparisons, since each new non dominated solution found in parent or offspring populations (in the worst case N individuals) is compared in terms of dominance with all the solutions in SP (in the worst case n_{SP} solutions). If it is assumed that $n_{SP} \simeq N$ then the $O(mN^2)$ evaluations are required. The overall complexity is given by $O(mN^3) + O(mN^2) + O(mN^2)$, i.e., $O(mN^3)$.

4 Results

4.1 Test Problems

The multiobjective problems were chosen from a number of significant past studies in the area (Zitzler et al., 2000; Deb et al., 2000). All problems have two objective functions and no constraints. Table 3 describes these problems, showing the number of variables, their bounds, the Pareto-optimal solutions known, and the nature of the Pareto-optimal front for each problem.

The MEES was applied to each problem with a reasonable set of values for the parameters (no effort was made in finding the best parameter setting for each problem). The initial values for standard deviations (step sizes) and parameters for its self adaptation during the search were the suggested values for ESs in single objective optimization; i.e., the initial values for σ_i were given by equation 1 with $\Delta x = (x_u - x_l)/2$ (where x_u and x_l are the upper and lower bounds of variable x_i) and $\Delta\sigma = \Delta\sigma' = 1/\sqrt{n}$. The points in the initial population were generated randomly. Several scenarios were considered in order to study the effects of the recombination operator, the selection mechanism, the elitism and d parameter. Thus, for each scenario all parameter values were kept constant except the feature under study (interaction between parameters was not studied in this phase).

Problem	Variable bounds	Objective functions	Optimal solutions	Comments
ZDT1 $n = 30$	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., n$	convex
ZDT2 $n = 30$	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (x_1/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., n$	nonconvex
ZDT3 $n = 30$	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - \frac{x_i}{g(x)} \sin(10\pi x_1)]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., n$	convex, disconnected
ZDT4 $n = 10$	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, ..., n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., n$	nonconvex
ZDT5 $n = 11$	$x_1 \in \{0, 1\}^{30}$ $x_i \in \{0, 1\}^5$ $i = 2, ..., n$	$f_1(x) = 1 + u(x_1)$ $f_2(x) = g(x)(1/x_1)$ $g(x) = \sum_{i=2}^{n}[v(u(x_i))]$ $v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{if} u(x_i) < 5 \\ 1 & \text{if} u(x_i) \geq 5 \end{cases}$ $u(x_i)$ gives the number of ones of x_i (unitation)	$x_1 \in \{0, 1\}^{30}$ $u(x_i) = 0$ $i = 2, ..., n$	convex, deceptive
ZDT6 $n = 10$	$[0, 1]$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(4\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., n$	nonconvex, nonuniformly spaced

Table 3: Multiobjective Problems

4.2 Metrics of Performance

Comparing different multiobjective optimization algorithms is substantially more complex than for the case of single objective optimizers, because the optimization goal itself consists of finding a non-dominated set of solutions that is:

- a good approximation to the true Pareto optimal set (the distance between the approximation and the true sets should be minimized);
- a well distributed set in the objective space.

Several attempts can be found in literature to express the above statements by means of quantitative metrics. The metric considered here is described by Knowles and Corne (2000) and is based on a statistical method proposed by Fonseca and Fleming (1995). The basic idea is to compare the resulting approximations to the Pareto surface from different algorithms by defining attainment surfaces through joining lines. An attainment surface divides the objective space into two regions: one containing solutions that are non-dominated by the solutions returned by the algorithm, and another containing all the points that are dominated. A collection of sampling lines that intersect the attainment surfaces is considered in order to determine which algorithm outperforms or it is outperformed (for each sampling line, an algorithm outperforms another if the intersection of the sampling line with its attainment surface is closer to the origin). Then, for several executions of the algorithms, a statistical test based on the Mann-Whitney rank-sum test is applied to the previous collected data. The results of a comparison can be presented in a pair $[a, b]$, where a is the percentage of the objective space on which algorithm A was found statistically superior to B, and b gives the similar percentage for algorithm B. Thus, a is the percentage of the objective space where algorithm A is 'unbeaten' and, b is the percentage of the objective space where

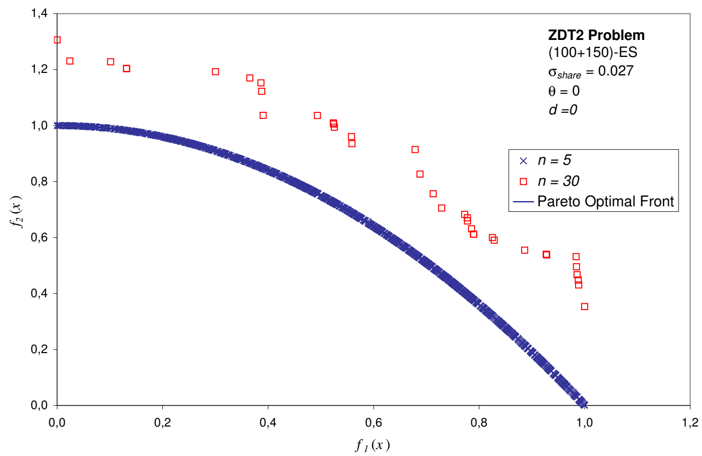

Figure 5: Results for problem ZDT2 with 5 and 30 variables

algorithm B is 'unbeaten'. So, typically, if $a \approx b \approx 100\%$ then the algorithms A and B have similar results. For all results presented in the paper the statistical significance is at the 5% level and 1000 sampling lines were used.

4.3 Influence of Recombination

The MEES without the recombination operator seems to have difficulties in obtaining a well distributed set of non-dominated solutions when applied to multiobjective problems with a high number of variables. This is illustrated by Figure 5, which represents the non-dominated solutions obtained, in one single run, for problem ZDT2 with 5 and 30 variables for an (100+150)-ES without any recombination and, with $\sigma_{share} = 0.027$, $d = 0$ and $\theta = 0$. The stopping criterion was to terminate the execution after 250 generations. It is clear that a good definition of the approximation to the Pareto-optimal set was obtained for problem ZDT2 with 5 variables. However, for 30 variables, the results are poor, in the sense, that the solutions are far from the true Pareto-optimal front and, they are not uniformly distributed in the objective space.

Since MEES without recombination seems to perform poorly for large dimensional multiobjective problems, several scenarios of MEES with recombination were tested. Scenarios that combine the most used recombination schemes were considered:

- without any recombination (NOrec scenario);
- intermediate recombination on variables and standard deviations (IIrec scenario);
- intermediate recombination on variables and discrete recombination on standard deviations (IDrec scenario);
- discrete recombination on variables and intermediate recombination on standard deviations (DIrec scenario);
- discrete recombination on variables and standard deviations (DDrec scenario).

ZDT1	IIrec	IDrec	DIrec	DDrec
NOrec	[100,3.8]	[100,2.2]	[2.7,100]	[2.7,100]
IIrec	-	[100,7.2]	[3.6,100]	[3.7,100]
IDrec	-	-	[2.1,100]	[2.1,100]
DIrec	-	-	-	[56.7,100]

Table 4: Influence of Recombination (problem ZDT1)

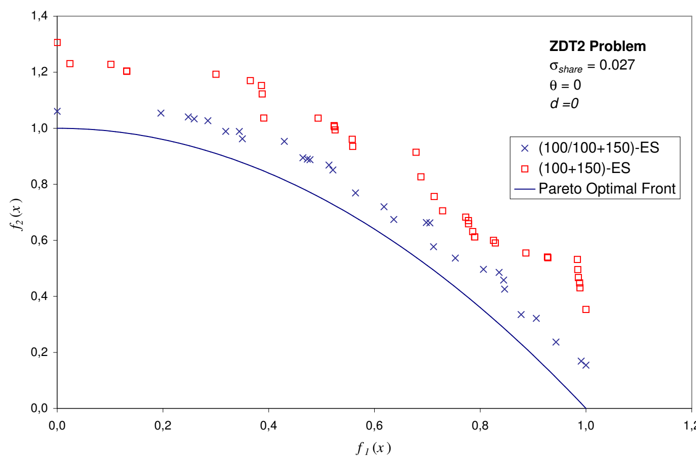

Figure 6: Results for problem ZDT2 with and without recombination

For scenarios with recombination, an (100/100+150)-ES was applied (obviously, an (100+150)-ES was considered when no recombination exists) with $\sigma_{share} = 0.027$, $d = 0$ and $\theta = 0$. As before, the stopping criterion was to terminate the execution after 250 generations. For each scenario, the MEES was executed 30 times. Table 4 presents the results obtained for all scenarios for problem ZDT1 (for each pair $[a, b]$, a and b refer, respectively, to the row and column algorithms in the table). All scenarios were compared in pairs using the statistical technique as previously described. It is clear that the best results were obtained for the DDrec scenario, i.e., when discrete recombination is applied to decision variables and standard deviations.

The MEES with discrete recombination on variables and standard deviations (an (100/100+150)-ES) can now be compared with the performance of MEES without any recombination for problem ZDT2. The comparison is illustrated by Figure 6, which represents the non-dominated solutions obtained in one single run after 250 generations. The approximation to the Pareto-optimal front obtained with the MEES with recombination, was far better than the one obtained without any recombination.

4.4 Influence of Elitism

In order to study the influence of the elitism level, an (100/100+150)-ES with discrete recombination on variables and standard deviations was applied to problem ZDT1. The same values for the parameters were considered with the exception of θ, which was varied from 0 to 100 as in Table 5. The d parameter was fixed equal to 0 in order to guarantee that in SP all non-dominated solutions found during the search are present.

ZDT1	$\Theta = 10$	$\Theta = 20$	$\Theta = 50$	$\Theta = 100$
$\Theta = 0$	[0.3,100]	[0.3,100]	[0.3,100]	[0.3,100]
$\Theta = 10$	-	[100,32.1]	[100,13.5]	[100,13.5]
$\Theta = 20$	-	-	[96.9,83.9]	[96.9,83.9]
$\Theta = 50$	-	-	-	[100,100]

Table 5: Influence of elitism (problem ZDT1)

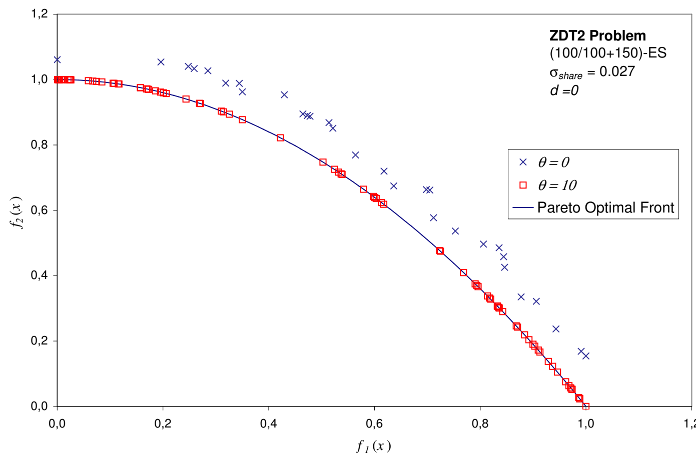

Figure 7: Results for problem ZDT2 for $\theta = 0$ and $\theta = 10$

This table shows that for increasing values of θ there is a degradation of the performance of the algorithm, due to the lack of diversity in the main population. However, it is also clear that the best results were obtained with elitism. Furthermore, in general, the best results were obtained with $\theta = 10$. The comparison between different levels of elitism is illustrated by Figure 7, which represents the non-dominated solutions obtained in one single run, after 250 generations, for problem ZDT2, with $\theta = 0$ and $\theta = 10$. It is clear that the approximation to the Pareto-optimal front obtained with $\theta = 10$ was far better than with $\theta = 0$.

4.5 Influence of Adaptive Sharing and Selection

Problem ZDT1 was also considered to investigate the effect of adaptive sharing and selection on the performance of the MEES. As before, an (100/100+150)-ES with discrete recombination on variables and standard deviations was considered. The same values for parameters were considered, except the values of σ_{share}. The "+" selection scheme was also compared with "$+_{spress}$" selection scheme varying the selection pressure $spress$. Table 6 presents the results varying the value of the selection pressure $spress$. It is clear that, the simplest "+" selection scheme is beaten by "$+_5$", "$+_{10}$" and "$+_{15}$" selection schemes. Furthermore, for this problem, the best results were obtained with the "$+_5$" selection scheme.

Table 7 shows the results obtained for problem ZDT1 for $\sigma_{share} = 0.014$, $\sigma_{share} = 0.027$ and using adaptive sharing ($\alpha = 0$, $\alpha = 0.5$ and $\alpha = 1$). In this table only the "+" selection was considered. In Table 8, comparisons are made with the results obtained

ZDT1	$+_{2}$	$+_{5}$	$+_{10}$	$+_{15}$	$+_{20}$	$+_{25}$	$+_{50}$	$+_{90}$	$+_{100}$
+	[100,3.2]	[92.5,96.1]	[93.6,97.0]	[97.0,97.6]	[98.5,95.0]	[98.1,93.1]	[99.0,91.8]	[96.7,92.6]	[99.0,87.3]
$+_{2}$	-	[3.2,100]	[3.2,100]	[3.2,100]	[3.2,100]	[3.2,100]	[3.2,100]	[3.2,100]	[3.2,100]
$+_{5}$	-	-	[94.0,93.1]	[96.1,94.0]	[98.1,88.2]	[97.4,89.6]	[99.1,85.5]	[98.1,90.8]	[99.4,77.7]
$+_{10}$	-	-	-	[96.6,95.6]	[98.0,92.0]	[97.1,89.3]	[97.9,85.5]	[97.6,91.2]	[98.9,81.2]
$+_{15}$	-	-	-	-	[97.6,91.7]	[98.5,90.9]	[98.4,89.5]	[97.3,92.5]	[99.6,83.4]
$+_{20}$	-	-	-	-	-	[96.6,96.3]	[97.5,93.2]	[95.7,95.5]	[97.5,88.5]
$+_{25}$	-	-	-	-	-	-	[97.6,92.3]	[94.8,95.2]	[98.0,90.8]
$+_{50}$	-	-	-	-	-	-	-	[94.7,98.3]	[97.2,92.8]
$+_{90}$	-	-	-	-	-	-	-	-	[98.5,89.5]

Table 6: Influence of selection pressure (problem ZDT1)

		+			
ZDT1		$\sigma_{sh} = 0.027$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
+	$\sigma_{sh} = 0.014$	[92.6,97.3]	[92.9,97.8]	[92.9,97.8]	[96.0,96.0]
	$\sigma_{sh} = 0.027$	-	[95.5,93.5]	[96.5,93.8]	[99.0,93.7]
	$\alpha = 0$	-	-	[94.3,93.4]	[94.9,93.7]
	$\alpha = 0.5$		-	-	[94.6,93.6]

Table 7: Influence of adaptive sharing (problem ZDT1)

for the "$+_5$" selection.

Tables 7 and 8 compare the solutions with different σ_{share} values and the adaptive sharing scheme. It is well known that some difficulty exists regarding the fixation of a value for σ_{share}. In this work several values of σ_{share} were tested. The best results were obtained for $\sigma_{share} = 0.027$. The adaptive sharing scheme avoids the need to search for an initial value for σ_{share}. The comparison shows that, although the best results for this problem were obtained for $\sigma_{share} = 0.027$, the proposed scheme does not differ markedly for α values close to zero. Furthermore, the adaptive scheme is best when compared to the other σ_{share} values tested. It should be noted that some experimentation is needed in order to find the best values for α. However, it can be seen, by the results obtained, that the algorithm is more sensitive to changes in σ_{share} than to changes in the α values.

4.6 Comparison of Different Multiobjective Evolutionary Approaches

The Multiobjective Elitist Evolution Strategy (MEES) was compared with nine algorithms on test problems ZDT1 to ZDT6. Some of these results were published by Zitzler et al. (2000) (http://www.tik.ee.ethz.ch/~zitzler/testdata.html) and Knowles and Corne (1999) (http://iridia.ulb.ac.be/~jknowles/multi/PAES.html). The algorithms considered here are:

- RAND: A random search algorithm;
- FFGA: Fonseca and Fleming's multiobjective EA;
- NPGA: The Niched Pareto Genetic Algorithm;
- HLGA: Hajela and Lin's weighted-sum based approach;
- VEGA: The Vector Evaluated Genetic Algorithm;
- NSGA: The Nondominated Sorting Genetic Algorithm;

ZDT1		$+_5$				
		$\sigma_{sh} = 0.014$	$\sigma_{sh} = 0.027$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
$+$	$\sigma_{sh} = 0.014$	[91.2,98.0]	[85.7,99.4]	[92.9,97.9]	[94.6,95.5]	[92.3,98.3]
	$\sigma_{sh} = 0.027$	[94.4,94.9]	[92.5,96.1]	[97.5,95.8]	[97.1,93.3]	[94.9,94.8]
	$\alpha = 0$	[93.9,95.1]	[91.9,96.5]	[95.2,95.5]	[95.9,94.9]	[94.0,94.8]
	$\alpha = 0.5$	[93.9,95.8]	[89.7,96.6]	[95.0,97.4]	[96.0,95.5]	[95.2,93.9]
	$\alpha = 1$	[91.4,97.0]	[88.5,98.3]	[92.0,95.3]	[96.2,97.3]	[91.6,95.2]
$+_5$	$\sigma_{sh} = 0.014$	-	[92.2,95.0]	[95.6,93.3]	[99.1,93.5]	[95.9,92.4]
	$\sigma_{sh} = 0.027$	-	-	[97.2,91.7]	[97.7,89.5]	[96.8,89.7]
	$\alpha = 0$	-	-	-	[97.4,93.6]	[94.7,94.4]
	$\alpha = 0.5$	-	-	-	-	[93.8,97.6]

Table 8: Influence of adaptive sharing and selection pressure (problem ZDT1)

- SOEA: A single-objective evolutionary algorithm using weighted-sum aggregation;
- SPEA: The Strength Pareto Evolutionary Algorithm;
- PAES: The Pareto Archived Evolution Strategy;
- NSGA2: The Nondominated Sorting Genetic Algorithm-II.

For MEES, an (100/100+150)-ES with discrete recombination in variables and standard deviations was considered. The MEES was applied without and with elitism ($MEES_0$ and $MEES_{10}$, respectively). The d and σ_{share} parameters were fixed equal to 0 and 0.027, respectively. The stopping criterion was to terminate the search after 100 generations. As described with more detail in (Zitzler et al., 2000), for algorithms FFGA, NPGA, HLGA, VEGA, NSGA, SOEA and SPEA, the population size was 100 (for SPEA the population size was 80 with an external non-dominated set of 20 points). The crossover and mutation rates were 0.8 and 0.01, respectively. The niching parameter was fixed in 0.48862. Several variants of the PAES were considered by Knowles and Corne (1999) for comparing their algorithm with other approaches. However, in this study, only PAES, with an external population (the archive) of 98 individuals, is considered. Two variants of the algorithm are considered: the PAES with standard binary representation of the variables (PAES) and the PAES with Gray coding of the variables ($PAES_{Gray}$). The mutation rate was 1%. The parameter for crowding was $l = 5$ (defining 1024 hypercubes). The NSGA2 was applied with real-coded representation of the decision variables ($NSGA2_r$) as well as with binary-coded representation of the decision variables ($NSGA2_b$) as proposed by Deb et al. (2000). For both approaches the crossover probability was 0.9. The mutation probability for $NSGA2_r$ was $1/n$ while for $NSGA2_b$ was $1/b$ (where n is the number of decision variables and b is the binary string length). Note that, for $NSGA2_b$, the total binary string length was $b = 30n$ since each decision variable was coded using 30 bits. For $NSGA2_r$, the distribution indices for crossover and mutation were equal to 20. It should be noted that MEES uses real representation and some of the algorithms with which it is compared use binary coding. For all algorithms (except for $MEES_0$ and $MEES_{10}$), the maximum number of generations was 250. All algorithms were executed 30 times for each test problem and, for each run, the set of all non-dominated solutions generated during the entire search was taken as the outcome of one optimization run (off-line performance). The number of objective function evaluations was the same for all algorithms (approximately, 25000 evaluations) except for $MEES_0$ and $MEES_{10}$ (15100 evaluations).

ZDT1	RAND	FFGA	NPGA	HLGA	VEGA	NSGA	SOEA
$MEES_0$	[100,0]	[100,9.2]	[100,10]	[100,8.3]	[100,11.4]	[100,1.3]	[75.1,37.4]
RAND	-	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	-	-	[72.2,78.7]	[21.5,93.7]	[15.2,100]	[9.6,100]	[9.3,100]
NPGA	-	-	-	[27.6,96.6]	[12.5,100]	[10.5,100]	[10.1,100]
HLGA	-	-	-	-	[75.2,77.5]	[8.8,100]	[8.4,100]
VEGA	-	-	-	-	-	[12,100]	[11.5,100]
NSGA	-	-	-	-	-	-	[1.3,100]

Table 9: Comparison between non elitist algorithms (problem ZDT1)

ZDT1	PAES	$PAES_{Gray}$	SPEA	$NSGA2_r$	$NSGA2_b$	$MEES_{10}$
$MEES_0$	[48.8,51.4]	[86.5,50.5]	[1.7,100]	[2.1,100]	[2.4,99.1]	[1.7,100]
RAND	[43.9,57.1]	[44.5,56.3]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	[45.8,54.8]	[42.6,54.5]	[9.2,100]	[10.1,100]	[11.9,91.5]	[9.2,100]
NPGA	[46.8,53.4]	[47.1,53.3]	[9.9,100]	[10.4,100]	[12.5,91.3]	[9.9,100]
HLGA	[47.5,53]	[48.3,52.7]	[8.3,100]	[11.7,100]	[17.1,94.8]	[8.3,100]
VEGA	[47.4,53]	[47.8,52.8]	[11.3,100]	[12.5,100]	[14.7,89.7]	[11.2,100]
NSGA	[48.4,51.8]	[49.9,51.6]	[1.3,100]	[2,100]	[2.6,99.4]	[1.3,100]
SOEA	[48.9,51.3]	[64.6,48.9]	[1,100]	[0,100]	[0,100]	[0,100]
PAES	-	[100,21.7]	[51.3,48.9]	[0,100]	[45.9,94.5]	[12.6,100]
$PAES_{Gray}$	-	-	[17.7,100]	[0,100]	[0,100]	[0,100]
SPEA	-	-	-	[2.1,100]	[2.7,98.6]	[1.9,100]
$NSGA2_r$	-	-	-	-	[100,0]	[100,28.4]
$NSGA2_b$	-	-	-	-	-	[9.7,98.5]

Table 10: Comparison between elitist and non elitist algorithms (problem ZDT1)

Tables 9 to 19 present the results of the comparison of these algorithms with MEES. Two classes of algorithms can be distinguished, those that do not use elitism (FFGA, NPGA, HLGA, VEGA, NSGA, SOEA and $MEES_0$) and, those that use, explicitly, elitism in the search (SPEA, PAES, $PAES_{Gray}$, $NSGA2_r$, $NSGA2_b$ and $MEES_{10}$). Thus, considering only the results obtained with non elitist algorithms, the best results were obtained by $MEES_0$ for all test problems considered except for problem ZDT5. Moreover, $MEES_0$ has outperformed all other algorithms including the elitist approaches in problem ZDT4. The $MEES_0$ also exhibits a good performance on problem ZDT6 outperforming $MEES_{10}$. Comparing the results obtained by the two approaches using real representation of the decision variables, it is clear that $NSGA2_r$ has outperformed the $MEES_{10}$ on problems ZDT1 and ZDT2; however, the opposite has occurred for problem ZDT3. The performance of $MEES_{10}$ on problem ZDT5 was clearly inferior to the other algorithms.

ZDT2	RAND	FFGA	NPGA	HLGA	VEGA	NSGA	SOEA
$MEES_0$	[100,0]	[100,6.7]	[100,6.6]	[100,16.3]	[100,3.3]	[100,3.5]	[100,0]
RAND	-	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	-	-	[8.3,100]	[9.1,100]	[8.5,100]	[7.3,100]	[6.9,100]
NPGA	-	-	-	[61.4,75.7]	[29.3,88.3]	[8.1,100]	[7.1,100]
HLGA	-	-	-	-	[25.7,100]	[19,100]	[17.2,100]
VEGA	-	-	-	-	-	[4.6,100]	[3.7,100]
NSGA	-	-	-	-	-	-	[4.5,100]

Table 11: Comparison between non elitist algorithms (problem ZDT2)

ZDT2	PAES	$PAES_{Gray}$	SPEA	$NSGA2_r$	$NSGA2_b$	$MEES_{10}$
$MEES_0$	[49,51.3]	[75.7,50]	[17.8,97.3]	[1.5,100]	[1.8,100]	[1.5,100]
RAND	[43.9,57.1]	[46.1,54.4]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	[45.8,54.8]	[47.4,53]	[6.8,100]	[6.8,100]	[8,100]	[6.6,100]
NPGA	[47.4,52.8]	[48.5,51.7]	[6.7,100]	[6.4,100]	[6.6,100]	[6.4,100]
HLGA	[47.6,52.7]	[48.7,51.6]	[16.5,100]	[16.1,100]	[16.3,100]	[15.9,100]
VEGA	[47.9,52.2]	[49,51.3]	[3.4,100]	[3.4,100]	[3.5,100]	[3.2,100]
NSGA	[48.4,51.7]	[49.5,50.9]	[3.6,100]	[3.6,100]	[3.7,100]	[3.4,100]
SOEA	[48.8,51.4]	[50.5,50.3]	[0,100]	[0,100]	[0,100]	[0,100]
PAES	-	[100,46.3]	[51.3,48.9]	[0,100]	[48.2,76.8]	[41.2,97.1]
$PAES_{Gray}$	-	-	[49.2,96.2]	[0,100]	[0,100]	[0,100]
SPEA	-	-	-	[1.9,100]	[2.2,100]	[1.9,100]
$NSGA2_r$	-	-	-	-	[100,0.6]	[99,2.7]
$NSGA2_b$	-	-	-	-	-	[67.1,94.9]

Table 12: Comparison between elitist and non elitist algorithms (problem ZDT2)

ZDT3	RAND	FFGA	NPGA	HLGA	VEGA	NSGA	SOEA
$MEES_0$	[100,0]	[100,8.2]	[100,5.3]	[100,13.7]	[100,6.6]	[100,2.4]	[66.8,52.8]
RAND	-	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	-	-	[21.8,100]	[21.6,94]	[11.4,100]	[8.4,100]	[8.2,100]
NPGA	-	-	-	[26.9,91.3]	[16.4,91.9]	[5.5,100]	[5.3,100]
HLGA	-	-	-	-	[55.9,82.9]	[14.3,100]	[13.9,100]
VEGA	-	-	-	-	-	[6.9,100]	[6.7,100]
NSGA	-	-	-	-	-	-	[19.5,83.8]

Table 13: Comparison between non elitist algorithms (problem ZDT3)

ZDT3	PAES	$PAES_{Gray}$	SPEA	$NSGA2_r$	$NSGA2_b$	$MEES_{10}$
$MEES_0$	[63.4,48.8]	[71.3,36.3]	[2.4,100]	[2.7,100]	[2.8,98.4]	[2.7,100]
RAND	[48.2,58.7]	[44.9,55.5]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	[44.3,56.1]	[46.1,54.1]	[8.2,100]	[11.5,100]	[12.5,94.2]	[11.5,100]
NPGA	[45,55.2]	[46.8,53.5]	[5.3,100]	[5.7,100]	[7.5,96.3]	[5.7,100]
HLGA	[45.8,54.6]	[47.7,52.9]	[13.7,100]	[15.6,100]	[17.6,92.2]	[15.6,100]
VEGA	[45.8,54.5]	[47.5,52.9]	[6.6,100]	[7.3,100]	[8,96.8]	[7.3,100]
NSGA	[50,52.3]	[62.6,51.9]	[2.4,100]	[2.7,100]	[3.4,98.4]	[2.7,100]
SOEA	[62.1,49.9]	[75.7,35.7]	[2.2,99.3]	[0,100]	[0,100]	[0,100]
PAES	-	[100,3]	[45.4,65.3]	[0.4,100]	[32.5,85.8]	[10.8,100]
$PAES_{Gray}$	-	-	[24.6,99.6]	[0,100]	[0,100]	[0,100]
SPEA	-	-	-	[0.9,100]	[1.1,99.5]	[0.9,100]
$NSGA2_r$	-	-	-	-	[100,0.3]	[30.6,100]
$NSGA2_b$	-	-	-	-	-	[28.4,100]

Table 14: Comparison between elitist and non elitist algorithms (problem ZDT3)

ZDT4	RAND	FFGA	NPGA	HLGA	VEGA	NSGA	SOEA
$MEES_0$	[100,0]	[100,6]	[100,6.6]	[100,32.7]	[100,9.8]	[100,16.7]	[100,0]
RAND	-	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	-	-	[6.5,100]	[17,100]	[6.3,100]	[6.1,100]	[6.9,100]
NPGA	-	-	-	[42.8,95.8]	[7.7,100]	[6.9,100]	[22.5,100]
HLGA	-	-	-	-	[87.6,47.4]	[33.1,100]	[87.7,46.7]
VEGA	-	-	-	-	-	[10.6,100]	[94.7,18.4]
NSGA	-	-	-	-	-	-	[100,0]

Table 15: Comparison between non elitist algorithms (problem ZDT4)

ZDT4	PAES	$PAES_{Gray}$	SPEA	$NSGA2_r$	$NSGA2_b$	$MEES_{10}$
$MEES_0$	[100,0]	[100,0]	[100,13.7]	[96.5,58.1]	[100,13.2]	[100,27.9]
RAND	[19.6,91.6]	[22,84.7]	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	[13.9,90.4]	[15.3,88.6]	[6.1,100]	[6,100]	[6.1,100]	[6,100]
NPGA	[33.4,70.5]	[20.3,83.8]	[6.9,100]	[6.6,100]	[7,100]	[6.6,100]
HLGA	[100,8.4]	[45.6,55.8]	[33.1,100]	[32.7,100]	[33.3,100]	[32.7,100]
VEGA	[100,56.2]	[31.2,71.8]	[10.6,100]	[9.8,100]	[11.2,100]	[9.8,100]
NSGA	[100,0]	[42.1,60]	[82.2,100]	[16.7,100]	[100,18.2]	[16.7,100]
SOEA	[48.1,59.8]	[27.7,72.3]	[0,100]	[0,100]	[4,100]	[0,100]
PAES	-	[32,100]	[0,100]	[0,100]	[0,100]	[0,100]
$PAES_{Gray}$	-	-	[60.6,42.9]	[0,100]	[64.7,38.3]	[0,100]
SPEA	-	-	-	[13.7,100]	[100,16.2]	[13.7,100]
$NSGA2_r$	-	-	-	-	[100,13.2]	[99.1,54.7]
$NSGA2_b$	-	-	-	-	-	[13.2,100]

Table 16: Comparison between elitist and non elitist algorithms (problem ZDT4)

ZDT5	RAND	FFGA	NPGA	HLGA	VEGA	NSGA	SOEA
$MEES_0$	[100,31.7]	[10.4,95.8]	[0,100]	[23.8,78.1]	[0,100]	[0,100]	[0,100]
RAND	-	[1.6,100]	[0,100]	[20.9,81]	[0,100]	[0,100]	[0,100]
FFGA	-	-	[82.1,72.7]	[75.8,44.1]	[68.4,41.9]	[32.5,71]	[1.8,100]
NPGA	-	-	-	[79.3,49.6]	[65.1,41.8]	[37.3,64.7]	[12.6,100]
HLGA	-	-	-	-	[57.3,46.6]	[39.1,64.3]	[0,100]
VEGA	-	-	-	-	-	[0,100]	[0,100]
NSGA	-	-	-	-	-	-	[17.8,93.1]

Table 17: Comparison between non elitist algorithms (problem ZDT5)

ZDT5	PAES	SPEA	$NSGA2_b$	$MEES_{10}$
$MEES_0$	[0,100]	[0,100]	[0,100]	[0,100]
RAND	[0,100]	[0,100]	[0,100]	[0,100]
FFGA	[47.9,57.7]	[7.2,100]	[2.7,100]	[55.8,53]
NPGA	[51.5,54.6]	[14.5,100]	[12.6,100]	[54,54.2]
HLGA	[53,51.6]	[0,100]	[0,100]	[50.9,51.8]
VEGA	[67.6,99.8]	[0,100]	[0,100]	[56.4,91.6]
NSGA	[100,0]	[54.3,100]	[38.6,100]	[100,0]
SOEA	[100,0]	[93,36.7]	[78.9,45.2]	[100,0]
PAES	-	[100,0]	[0,100]	[92.9,58.6]
SPEA	-	-	[73.8,100]	[100,0]
$NSGA2_b$	-	-	-	[100,0]

Table 18: Comparison between elitist and non elitist algorithms (problem ZDT5)

ZDT6	RAND	FFGA	NPGA	HLGA	VEGA	NSGA	SOEA
$MEES_0$	[81.1,19.5]	[76.4,26]	[67.7,33.3]	[63.8,36.9]	[64.6,35.7]	[58.8,41.5]	[52.8,47.2]
RAND	-	[6.2,100]	[0,100]	[0,100]	[6.6,100]	[0,100]	[0,100]
FFGA	-	-	[0,100]	[0,100]	[17.1,100]	[0,100]	[0,100]
NPGA	-	-	-	[19,100]	[38.6,65.1]	[13.5,100]	[11.5,100]
HLGA	-	-	-	-	[100,11.2]	[4.9,100]	[4,100]
VEGA	-	-	-	-	-	[0,100]	[0,100]
NSGA	-	-	-	-	-	-	[8,100]

Table 19: Comparison between non elitist algorithms (problem ZDT6)

ZDT6	PAES	PAES_{Gray}	SPEA	NSGA2_r	NSGA2_b	MEES_{10}
MEES_0	[45.7,73.1]	[49.8,50.4]	[51.3,49.2]	[71.3,79.4]	[100,13.9]	[85.1,47]
RAND	[6.6,97.5]	[0,100]	[0,100]	[19.5,81.1]	[19.7,80.9]	[19.5,81.1]
FFGA	[12.1,100]	[0,100]	[0,100]	[26,76.4]	[26.3,76.1]	[2.6,76.4]
NPGA	[23.2,78.9]	[0,100]	[9.4,100]	[22.7,78.7]	[23.2,78.3]	[22.7,78.7]
HLGA	[26.7,80.2]	[0,100]	[3.5,100]	[25.1,76]	[25.6,75.4]	[25,76]
VEGA	[26.2,100]	[0,100]	[0,100]	[34.8,65.4]	[35.5,64.8]	[34.8,65.4]
NSGA	[32.9,67.7]	[1.1,100]	[5.2,100]	[15.5,85.3]	[16,85]	[15.5,85.3]
SOEA	[61,39]	[10.8,95.5]	[0,100]	[8.9,91.1]	[9.1,90.9]	[8.9,91.1]
PAES	-	[67.6,37.7]	[41.1,60]	[69.2,41.3]	[100,0]	[80.6,29]
PAES_{Gray}	-	-	[93,15.4]	[47.7,52.7]	[48.3,52]	[47.4,52.7]
SPEA	-	-	-	[17.1,85.4]	[17.1,85.4]	[17.1,85.4]
NSGA2_r	-	-	-	-	[100,0]	[87.4,25.7]
NSGA2_b	-	-	-	-	-	[0,100]

Table 20: Comparison between elitist and non elitist algorithms (problem ZDT6)

	MEES_0	NSGA	SOEA	PAES	SPEA	NSGA2_r	NSGA2_b	MEES_{10}
ZDT1	-36.8	-84.7	-48.6	-32.2	+0.2	+95.1	+35.6	+71.4
ZDT2	-25.8	-83.0	-58.2	-24.8	-2.9	+98.4	+42.0	+54.4
ZDT3	-37.4	-78.8	-47.8	-40.9	+2.2	+74.8	+38.7	+89.1
ZDT4	+81.0	+2.0	-86.8	-84.0	+6.1	+66.1	-32.9	+48.6
ZDT5	-100.0	+19.6	+77.6	-11.0	+10.5	-	+59.0	-55.7
ZDT6	+16.2	-63.9	-34.0	+28.7	+1.0	+51.7	-23.9	+24.1

Table 21: Global Performance

Table 21 summarizes the results obtained by different algorithms for all problems. The values in the table, for each algorithm, are the average of the differences between the percentage of the objective space that is 'unbeaten' when compared with the remaining algorithms. Thus, for each problem the algorithms with best performance are those that present higher tabled values. From this table, it is clear that, in general, elitism is useful in guiding the search. Moreover, the real representation of the decision variables allowed, globally, better results to be achieved. The best non elitist algorithm seems to be MEES_0, which outperformed all the other approaches for problem ZDT4. The most consistent results were obtained by the NSGA2_r algorithm for all problems with the exception of problems ZDT3 and ZDT4. The MEES_{10} outperformed the NSGA2_r on problem ZDT3.

5 Conclusions and Future Work

In this work, a new Elitist Evolution Strategy for multiobjective optimization was presented. This approach incorporates the main features of traditional single objective Evolution Strategies, such as real representation of the decision variables and self-adaptation of step sizes.

However, new features were introduced that distinguish this approach, i.e., an adaptive sharing, a selection operator controlling the selection pressure $spress$, a parameter θ defining the level of elitism, a parameter d that defines the concentration of solutions along the approximation to the Pareto front in the secondary population.

With respect to adaptive sharing, the proposed scheme avoids the need to search for a "best" value for the parameter σ_{share}. The selection operator incorporates solutions from different fronts, through a geometric progression, therefore allowing greater diversity while controlling the selection pressure. The study of the number of solutions from the secondary population that are introduced in the search, allowed an adequate level of elitism to be defined. The parameter d, while stating the minimal desirable distance in the objective space, controls the size of the secondary population.

The algorithm was tested on several test problems in order to investigate the influence of several factors on its performance, as well as, to compare its performance with other multiobjective evolutionary approaches. As expected, the results indicated that recombination and elitism are essential for obtaining good approximations to the Pareto-optimal front. The Multiobjective Elitist Evolution Strategy without elitism ($MEES_0$) outperformed the other non elitist approaches (HLGA, VEGA and NSGA) for all the test problems considered with the exception of problem ZDT5. The Multiobjective Elitist Evolution Strategy with elitism ($MEES_{10}$) outperformed the $NSGA2_r$ in one problem. However, $NSGA2_r$ performed better for problems ZDT1, ZDT2 and ZDT6. It should be noted that MEES requires more function evaluations than the other algorithms under consideration. In spite of the positive results obtained using MEES, the effect of recombination, elitism and selection must be studied on other problems.

Future work will concentrate on the study of the algorithm parameters and on controlling the density of points in the approximation to the Pareto-optimal front as well as the extension of Pareto front based on the variable space. Moreover, the MEES will be compared to other approaches such as PESA (Corne et al., 2000), Micro-GA (Coello Coello and Toscano, 2001) and MOPSO (Coello Coello and Lechuga, 2002). Further investigation will be carried out with other test problems including constraints.

References

Coello Coello, C. and Lechuga, M. (2002). Mopso: A proposal for multiple objective particle swarm optimization. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1051–1056, Piscataway, New Jersey. IEEE Service Center.

Coello Coello, C. and Toscano, G. (2001). A micro-genetic algorithm for multiobjective optimization. In Zitzler, E., Deb, K., Thiele, L., Coello, C. C., and Corne, D., editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 126–140. Lecture Notes in Computer Science No. 1993, Springer-Verlag.

Corne, D., Knowles, J., and Oates, M. (2000). The pareto envelope-based selection algorithm for multiobjective optimization. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J., and Schwefel, H.-P., editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848. Springer.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858.

Deb, K. and Goel, T. (2001). Controlled elitist non-dominated sorting genetic algorithms for better convergence. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 67–81.

Deb, K. and Goldberg, D. (1989). An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, USA.

Fonseca, C. and Fleming, P. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Ed. San Mateo, CA: Morgan Kauffman.

Fonseca, C. and Fleming, P. (1995). On the performance assessment and comparison of stochastic multiobjective optimizers. In Voigt, H.-M., Ebeling, W., Rechenberg, I., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature IV*, pages 584–593. Eds Berlin: Germany: Springer.

Fonseca, C. and Fleming, P. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part II: Application example. *IEEE Transactions on Systems, Man, and Cybernetics: Part A: Systems and Humans*, 28(1):38–47.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts.

Horn, J., Nafploitis, N., and Goldberg, D. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87.

Knowles, J. and Corne, D. (1999). Local search, multiobjective optimization and the pareto archived evolution strategy. In *Proceedings of the Third Australia-Japan Joint Workshop on Intelligent and Evolutionay Systems*, pages 209–216.

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

Kursawe, F. (1990). A variant of evolution strategies for vector optimization. In Schwefel, H.-P. and Manner, R., editors, *Parallel Problem Solving from Nature*, pages 193–197. Eds Berlin: Germany: Springer.

Rechenberg, I. (1964). Cybernetic solution path of an experimental problem. In *Royal Aircraft Establishment*, Farnborough, England. Library Translation No. 1122.

Rechenberg, I. (1994). *Evolutionsstrategie '94*. Frommann-Holzboog, Stuttgart.

Schaffer, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In Grefensttete, J., editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Ed. Hillsdale.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Wiley, New York.

Srinivas, N. and Deb, K. (1995). Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

Van Veldhuizen, D. and Lamont, G. (2000). Multiobjective evolutionary algorithms: Analysing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195.

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.