

Pareto-adaptive ε -dominance*

Alfredo G. Hernández-Díaz¹, Luis V. Santana-Quintero²,
Carlos A. Coello Coello² and Julián Molina³

¹Department of Quantitative Methods,
Pablo de Olavide University, Seville, Spain

²CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Ingeniería Eléctrica,
Sección Computación

Av. IPN No. 2508, Col. San Pedro Zacatenco,
México D.F. 07360, México

³Department of Applied Economics (Mathematics),
University of Málaga, Málaga Spain
agarher@upo.es, lvspenny@hotmail.com,
ccoello@cs.cinvestav.mx, julian.molina@uma.es

Abstract

Efficiency has become one of the main concerns in evolutionary multiobjective optimization during recent years. One of the possible alternatives to achieve a faster convergence is to use a relaxed form of Pareto dominance that allows to regulate the granularity of the approximation of the Pareto front that we wish to achieve. One of such relaxed forms of Pareto dominance that has become popular in the last few years is ε -dominance, which has been mainly used as an archiving strategy in some multi-objective evolutionary algorithms. Despite its advantages, ε -dominance has some limitations. In this paper, we propose a mechanism that can be seen as a variant of ε -dominance, which we call Pareto-adaptive ε -dominance (*pa* ε -dominance). Our proposed approach tries to overcome the main limitation of ε -dominance: the loss of several nondominated solutions from the hypergrid adopted in the archive because of the way in which solutions are selected within each box.

* Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos A. Coello Coello and Julián Molina, "Pareto-adaptive ε -dominance", Technical Report EVOCINV-02-2006, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México, March 2006.

1 Introduction

Laumanns et al. (2002) proposed a relaxed form of dominance for multi-objective evolutionary algorithms (MOEAs), named ε -dominance. This mechanism acts as an archiving strategy to ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found. Laumanns et al. (2002) proposed an extension of the classical Pareto-dominance relation so that a point $f \in \mathbb{R}^m$ not only dominates those points with lower fitness in all their objectives but also all points close enough to f (i.e., those with a distance to f less than an ε). This value, ε , could be provided by the decision maker to control the size of the solution set. Nevertheless, because of the geometrical characteristics of the Pareto-optimal set (concavity, convexity, curvature, torsion, disconnected segments, etc.) are usually unknown to the decision maker, we can lose a high number of good solutions if the ε value is badly chosen.

Despite the obvious usefulness of ε -dominance, this mechanism has several drawbacks from which the main one has to do with the difficulties to compute an appropriate value of ε that provides the number of nondominated solutions that the user wants. Another important limitation of this mechanism is the fact that it loses solutions lying on segments of the Pareto front that are almost horizontal or almost vertical, as well as the extreme points of the Pareto front.

This paper provides an extension of ε -dominance that addresses the problems indicated above. The remainder of this paper is organized as follows: In Section 2, we provide the basic definitions associated to ε -dominance, as well as a brief description of its main limitations. Section 3 contains the detailed description of our proposed scheme. The experimental setup used to validate our proposed approach is provided in Section 4. Our scheme is incorporated into a multi-objective evolutionary algorithm, and its results are compared with respect to the same algorithm using ε -dominance and with respect to the ε -MOEA (Deb et al., 2005a). Our results are presented and analyzed in Section 5. Finally, some conclusions and possible paths for future research are provided in Section 6.

2 ε -dominance

Laumanns et al. (2002) proposed two different methods/schemes to implement ε -dominance: the additive and the multiplicative approaches. We assume all objectives are to be minimized. Then, given a vector $f \in \mathbb{R}^m$ and $\varepsilon > 0$, for the additive scheme f is said to ε -dominate all points in the set

$$\{g \in \mathbb{R}^m : f_i - \varepsilon \leq g_i, \text{ for all } i = 1, \dots, m\}$$

whereas for the multiplicative scheme f is said to ε -dominate all points in the set

$$\{g \in \mathbb{R}^m : f_i(1 - \varepsilon) \leq g_i, \text{ for all } i = 1, \dots, m\}.$$

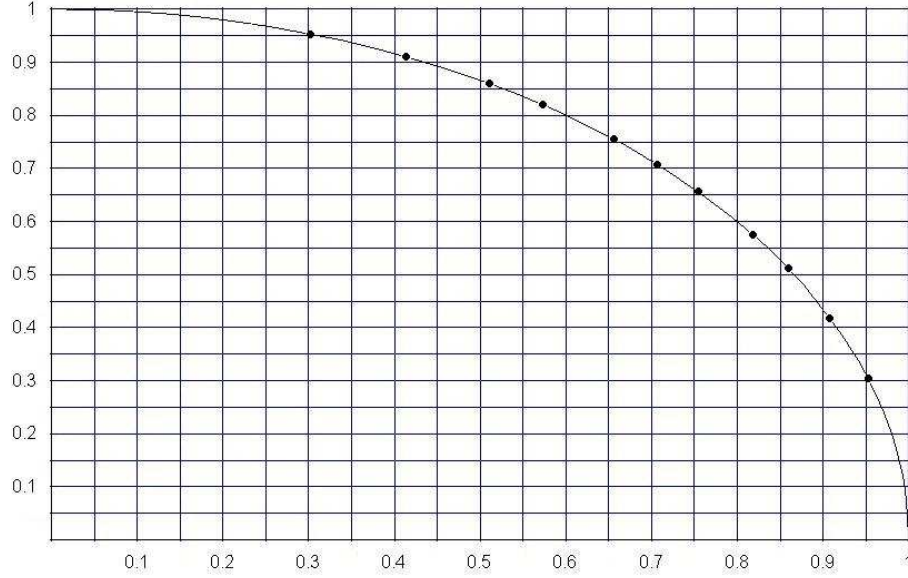


Figure 1: Uniform grid with 400 boxes (maximum capacity of 20 points) for the curve $x^2 + y^2 = 1$. This grid allows a maximum of 11 points (the other 9 points are lost) because the extreme points are easily ε -dominated.

Although the above definitions assume the same ε value for all the objectives, they can be easily generalized to consider a different value for each of objective. In order to do this, we only have to take an ε_i for each $i \in \{1, 2, \dots, m\}$. Without loss of generality, we assume that $1 \leq f_i \leq K$, for all i .

So, both schemes generate a hyper-grid in the objective functions space with $\left(\frac{K-1}{\varepsilon}\right)^m$ boxes in the additive scheme and $\left(\frac{-\log K}{\log(1-\varepsilon)}\right)^m$ for the multiplicative one. As ε -dominance only allows one point in each box, these grids could accommodate a maximum of $\left(\frac{K-1}{\varepsilon}\right)^{m-1}$ non ε -dominated points for the additive scheme and $\left(\frac{-\log K}{\log(1-\varepsilon)}\right)^{m-1}$ non ε -dominated points for the multiplicative scheme. Another possibility would be to ask the decision maker for the number of desired solutions and adjust the ε values in order to achieve that number. For example, if the decision maker wants T points in the Pareto front, for the additive scheme we can easily compute the value $\varepsilon = (K-1)/T^{\frac{1}{m-1}}$, that will generate a hyper-grid with a maximum capacity of T points non ε -dominated. Similarly, this leads to an $\varepsilon = 1 - K^{-T^{\frac{1}{1-m}}}$ for the multiplicative scheme.

ε -dominance has been found to be an efficient mechanism to maintain diversity in multiobjective optimization problems without losing convergence properties towards the Pareto-optimal set (Deb et al., 2003; Deb et al., 2005a; Reyes Sierra and Coello Coello, 2005). However, this scheme has some limitations such as the following:

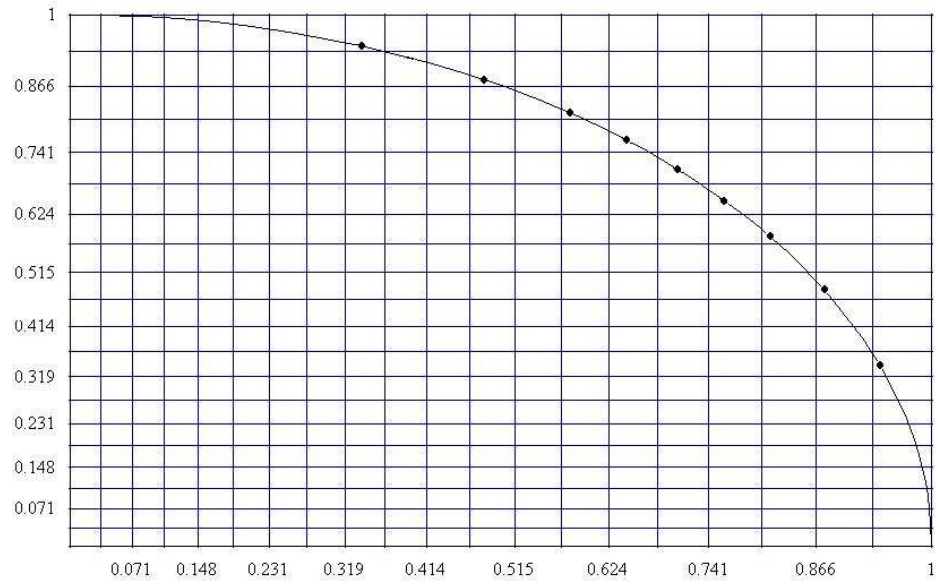


Figure 2: Non-uniform grid with 400 boxes (maximum capacity of 20 points) for the curve $x^2 + y^2 = 1$. In this case, because the front is concave, the grid only allows a maximum of 9 points, losing again both extreme points of the Pareto front.

1. We can lose a high number of efficient solutions if the decision maker does not take into account (or does not know beforehand) the geometrical characteristics of the true Pareto front of the problem to be solved.
2. It is normally the case that we lose the extreme points of the Pareto front, as well as points located in segments of the Pareto front that are almost horizontal or vertical, as shown in Figure 1.
3. The upper bound for the number of points allowed by a grid is not easy to achieve. For a non-adaptive grid, the upper bound is only achieved when the real Pareto front is linear.
4. When adopting a multiplicative scheme, the size of the region ε -dominated by the point $f \in \mathbb{R}^m$ depends on the f_i values. Then, the size of this region is larger in the cases where the f_i values increase. For the same reason, if the f_i values are close to zero, ε -dominance would be similar to the traditional Pareto-dominance. This kind of grid is not suitable, for instance, for concave Pareto fronts (see Figure 2).

3 Pareto-adaptive ε -dominance

In order to address some of the problems previously described, we propose an alternative scheme for the additive ε -dominance. Our proposal is called Pareto-adaptive- ε -dominance (*pa* ε -dominance). This scheme maintains the good properties of ε -dominance while overcoming its main limitations.

In our proposal, we consider not only a different ε value for each objective but also the vector $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)$ associated to each $f = (f_1, f_2, \dots, f_m) \in \mathbb{R}^m$ depending on the geometrical characteristics of the Pareto-optimal front. In other words, we consider different intensities of dominance for each objective according to the position of each point along the Pareto front. Then, the size of the boxes will be adapted depending on the area in the objective functions space so that boxes will be smaller where needed (normally at the extremes of the Pareto front), and larger in other less problematic parts of the front.

For this aim, each Pareto front (that we will assume normalized: $0 \leq f_i \leq 1$ for any i) will be associated to one curve of the following family

$$\{x^p + y^p = 1 : 0 \leq x, y \leq 1, 0 < p < \infty\}.$$

for bi-objective optimization problems, or

$$\{x^p + y^p + z^p = 1 : 0 \leq x, y, z \leq 1, 0 < p < \infty\}$$

for three dimensional problems. These families have the following property: For $p > 1$, the curve (or surface) is concave and the bigger the p value the longer the almost horizontal (and almost vertical) parts of the front; and, for $p < 1$, the curve (surface) is convex and the lower the p value the longer the almost horizontal (and almost vertical) stretches in the front. Finally, for $p = 1$ we get

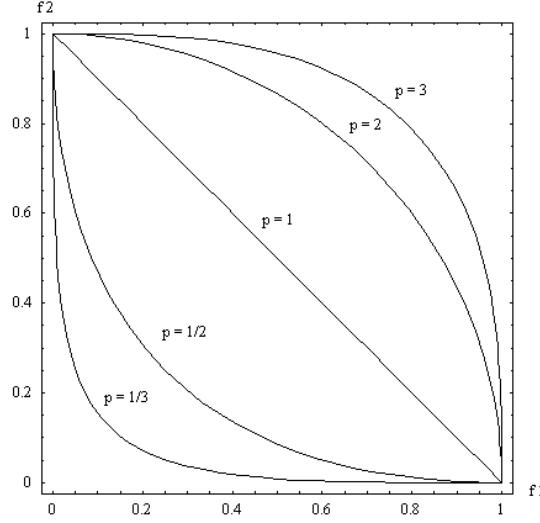


Figure 3: Curves in the reference set for $p = \frac{1}{3}, \frac{1}{2}, 1, 2, 3$. The ε values we have to consider for $p = 2$ and $p = 3$ have to be different because $x^3 + y^3 = 1$ has longer horizontal and vertical stretches than $x^2 + y^2 = 1$. The same happens for $p = \frac{1}{3}$ and $p = \frac{1}{2}$.

the linear front $x + y = 1$. For this last value, it will be shown that our scheme coincides with the additive ε -dominance. Thus, our proposal generalizes the ε -dominance concept introduced by Laumanns et al. (Laumanns et al., 2002) (just taking $p = 1$ in (1)).

In Figure 3 we show five different curves of this family for $p \in \{\frac{1}{3}, \frac{1}{2}, 1, 2, 3\}$.

In order to decide the value of p , we need an initial Pareto front approximation, denoted by F , that will determine which value of p fits better to our front. This is, we will use F to be the model where the p -curve should fit. Then, the number of efficient points included in F can be critical for the final performance, because if the value of p is not appropriate, then the grid will not be appropriate neither. Obviously, the higher the number of efficient points in F the better performance of the grid generated. On the other hand, if we want to maintain the diversity properties of ε -dominance, we should generate the first grid as soon as possible. For example, for a grid with a maximum capacity of 100 vectors, different experiments performed by the authors indicate that the best results are obtained when the number of points in F is between 75 and 125 (we set it at 100 for our experiments).

To compute the value of p , we calculate the area (hypervolume) under the polygonal line (surface) formed by points in F (see Section 3.3 for further details). Once we know this area, we estimate the value of $p \in (0, +\infty)$ by means of an interpolation process. We choose p when the area under $x^p + y^p = 1$ is as similar

to the F hypervolume as desired (this precision is set beforehand).

Although we are assuming that the Pareto front is symmetrical, this method could be generalized using the sets

$$\{x^p + y^q = 1 : 0 \leq x, y \leq 1, 0 < p, q < \infty\}$$

and

$$\{x^p + y^q + z^r = 1 : 0 \leq x, y, z \leq 1, 0 < p, q, r < \infty\}.$$

Nevertheless, the association procedure is more unstable, as it depends on F to a higher degree and the error for the estimated p and q values could be large.

3.1 ε computation

Once the p value is estimated and the number T of points desired by the decision maker is known, we compute the sizes of the boxes for each objective $i \in \{1, 2, \dots, m\}$, that is, the vector $\varepsilon^i = (\varepsilon_1^i, \varepsilon_2^i, \dots, \varepsilon_T^i)$.

We use geometric sequences to do this: we compute these values according to a geometric sequence depending on p , T and the size of the first box for each dimension, ε_1^i , so that, for $n \geq 2$,

$$\varepsilon_n^i = \frac{\varepsilon_{n-1}^i}{p^{v_i}} = \frac{\varepsilon_{n-2}^i}{(p^{v_i})^2} = \dots = \frac{\varepsilon_1^i}{(p^{v_i})^{n-1}} \quad (1)$$

where v_i controls the *speed of variation* of the ε values in order to get a uniform distribution in the Pareto front.

Then, for each objective $i \in \{1, 2, \dots, m\}$ we have to estimate the size of the first box, ε_1^i , and the speed v_i . To this end, we propose the following system of non-linear equations, for each i ,

$$\left. \begin{aligned} \sum_{n=1}^T \varepsilon_n^i &= 1 \\ \sum_{n=1}^{T/2} \varepsilon_n^i &= \frac{1}{2^{1/p}} \end{aligned} \right\}. \quad (2)$$

The first equation represents the fact that the sum of the sizes of all boxes must be equal to the range of f_i . The second equation tries to spread the obtained efficient points along the front and forces the accommodation of $T/2$ nondominated points in one half of the objective i , and the remaining $T/2$ points in the other half. Taking into account that $x^p + y^p = 1$ is symmetric, it is easy to obtain the middle point: $(\frac{1}{2^{1/p}}, \frac{1}{2^{1/p}})$.

As both series in (2) are geometric, it follows that

$$\left. \begin{aligned} \sum_{n=1}^T \varepsilon_n^i &= \sum_{n=1}^T \frac{\varepsilon_1^i}{(p^{v_i})^{n-1}} = \varepsilon_1^i \frac{1 - (\frac{1}{p^{v_i}})^T}{1 - \frac{1}{p^{v_i}}} = \varepsilon_1^i \frac{p^{T v_i} - 1}{(p^{v_i} - 1)p^{(T-1)v_i}} = 1 \\ \sum_{n=1}^{T/2} \varepsilon_n^i &= \sum_{n=1}^{T/2} \frac{\varepsilon_1^i}{(p^{v_i})^{n-1}} = \varepsilon_1^i \frac{1 - (\frac{1}{p^{v_i}})^{T/2}}{1 - \frac{1}{p^{v_i}}} = \varepsilon_1^i \frac{p^{\frac{T}{2} v_i} - 1}{(p^{v_i} - 1)p^{(\frac{T}{2}-1)v_i}} = \frac{1}{2^{1/p}} \end{aligned} \right\}. \quad (3)$$

Then, the solutions of (3) are

$$\left. \begin{aligned} \varepsilon_1^i &= \frac{(p^{v_i}-1)p^{(T-1)v_i}}{p^{Tv_i}-1} \\ \left(1 - 2^{\frac{1}{p}}\right) p^{Tv_i} + 2^{\frac{1}{p}} p^{\frac{T}{2}v_i} - 1 &= 0 \end{aligned} \right\}. \quad (4)$$

But ε_1^i is already calculated in the first equation and it does not appear in the second one. So, we only have to solve the second equation in (4). Due to its nonlinearity, we propose to solve it using a numerical method, for example, a dichotomy method (Rao, 1996). Although this is not the fastest numerical method available, we decided to use it because of the simplicity of its implementation and the easy that results to control the precision required. Along our experiments, we applied a dichotomy method for v_i in the interval $[0.001, 0.1]$ because we set $T = 100$ and $\frac{1}{12} \leq p \leq 12$.

3.2 Box Index Vector

As in the original ε -dominance, the dominance relation is generalized among boxes. That is, at most one element is kept in each box and this representative vector can only be replaced by a dominating one. To this end, we associate to each vector $f \in \mathbb{R}^m$ a *box index vector* $b(f) = (b_1, \dots, b_m) \in \mathbb{Z}^m$. So, in a first level, the algorithm always maintains a set of nondominated boxes (this is, a set of nondominated box index vectors). And in a second level, if two vectors share the same box, the representative vector is eliminated if the other one dominates it.

In order to calculate the box index vector of $f = (f_1, f_2, \dots, f_m)$, we take b_i to be the only integer so that

$$\sum_{n=1}^{b_i} \varepsilon_n^i \leq f_i < \sum_{n=1}^{b_i+1} \varepsilon_n^i.$$

for all $i \in \{1, 2, \dots, m\}$. Again, because both series are geometric, the above inequalities are equivalent to

$$\varepsilon_1^i \frac{p^{v_i} - \left(\frac{1}{p^{v_i}}\right)^{b_i-1}}{p^{v_i} - 1} \leq f_i < \varepsilon_1^i \frac{p^{v_i} - \left(\frac{1}{p^{v_i}}\right)^{b_i}}{p^{v_i} - 1}.$$

If we assume that $p^{v_i} - 1 > 0$, it is equivalent to

$$p^{v_i} - \left(\frac{1}{p^{v_i}}\right)^{b_i-1} \leq \frac{f_i (p^{v_i} - 1)}{\varepsilon_1^i} < p^{v_i} - \left(\frac{1}{p^{v_i}}\right)^{b_i}.$$

Then, by successive (elementary) operations, we have the following equivalent expressions

$$\ln \left(\frac{1}{p^{v_i}}\right)^{b_i-1} \geq \ln \left(p^{v_i} - \frac{f_i (p^{v_i} - 1)}{\varepsilon_1^i}\right) > \ln \left(\frac{1}{p^{v_i}}\right)^{b_i}.$$

$$(b_i - 1) \ln \left(\frac{1}{p^{v_i}} \right) \geq \ln \left(p^{v_i} - \frac{f_i(p^{v_i} - 1)}{\varepsilon_1^i} \right) > b_i \ln \left(\frac{1}{p^{v_i}} \right)$$

$$b_i - 1 \leq \frac{\ln \left(p^{v_i} - \frac{f_i(p^{v_i} - 1)}{\varepsilon_1^i} \right)}{\ln \left(\frac{1}{p^{v_i}} \right)} < b_i.$$

Finally, we choose

$$b_i(f) = \left\lceil \frac{\log \left(\frac{\varepsilon_1^i p^{v_i} - (p^{v_i} - 1) f_i}{\varepsilon_1^i} \right)}{\log \left(\frac{1}{p^{v_i}} \right)} + 1 \right\rceil.$$

It is easy to check that the same b_i is obtained if $p^{v_i} - 1 < 0$.

In that way, although the whole objective function space is discretized into boxes, the nondominated vectors are allocated into boxes whose box index vectors range from $(0, 0, \dots, 0)$ to $(T - 1, T - 1, \dots, T - 1)$. Nevertheless, if vectors outside of the above limits are found, we must include them in the grid (if their box index vectors are non- $pa\varepsilon$ -dominated) in one of the two following ways:

1. Update the grid re-computing new box limits. In this case, a new p value would be also calculated. This does not ensure the convergence property of ε -dominance (see (Laumanns et al., 2002)) and the behavior could be worse.
2. Do not change any of the box limits (the assignment of the elements to the boxes must remain the same). This guarantees the same convergence properties of ε -dominance but the number of nondominated points could be larger than T . In this case, a larger ε value can be chosen, but the grid would have to be updated again.

In our proposed approach we follow the second choice shown above, and, once the grid is generated, its boundaries are never modified. Note however, that the grid depends on the quality of the first set of nondominated points, F , as such set determines the value of p . The best performance is attained if there is no need to re-adjust this initial grid. We only update the grid when some of the coordinates of the new box index vector is sufficiently far to require it, this is, when $b_i < -3$ or $b_i > T + 3$ for some i . The best results have been obtained when the first grid is generated once F contains at least 100 nondominated points, as the hyper-grid is almost never re-adjusted with this setting. This minimum value has been empirically derived after numerous experiments for the curves $x^p + y^p = 1$. In all the cases that the authors empirically tested, with values close to 100, the value of p that was obtained was very close to the simulated curve.

Finally, if two vectors f and g share the same box (so, $b(f) = b(g)$) and neither dominates the other, we choose the one closer (using Euclidean distance, for

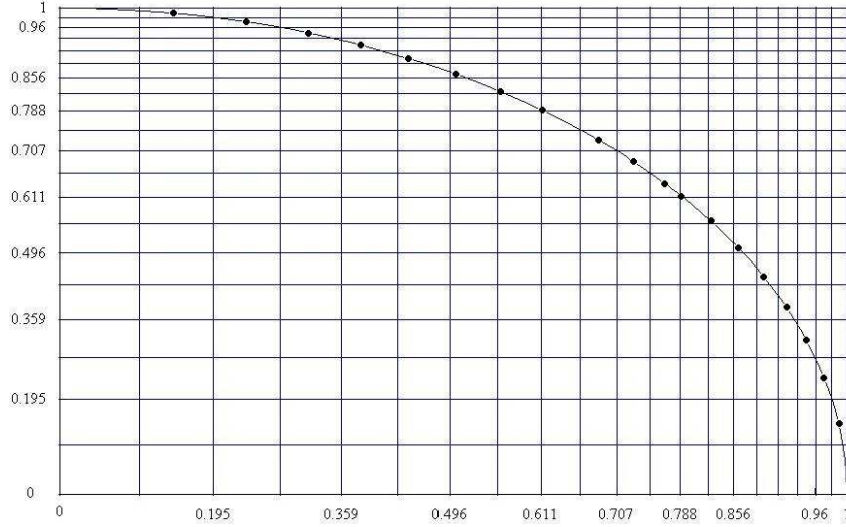


Figure 4: Alternative grid with 400 boxes (maximum capacity of 20 points) using $p\alpha\epsilon$ -dominance for the curve $x^2 + y^2 = 1$. In this case the grid allows a maximum of 19 points.

example) to the lower lefthand corner of the box, denoted by $c(b) = (c_1, \dots, c_m)$. In order to calculate c_i , we sum the size of all previous boxes, that is

$$c_i = \sum_{n=1}^{b_i} \epsilon_n^i = \epsilon_1^i \frac{p^{v_i b_i}}{(p^{v_i} - 1)p^{v_i(b_i-1)}}$$

for all $i = 1, 2, \dots, m$.

In Figure 4 we can see the grid obtained for $x^2 + y^2 = 1$. The figure clearly indicates how the grid adapts the size of the boxes as needed.

3.3 Algorithm for the Hypervolume

As we mentioned above, each Pareto front is associated to one curve in

$$\{x^p + y^p = 1 : 0 \leq x, y \leq 1, 0 < p < \infty\}$$

by estimating the area (hypervolume) under the polygonal line (surface) formed by the points in F in objective function space.

Let us assume a bi-objective optimization problem and $F = \{f^j = (f_1^j, f_2^j) : j = 1, 2, \dots, |F|\}$ is the set of nondominated vectors obtained before generating the hyper-grid. Obviously, if we rank points in F in ascending order of magnitude in the first objective, f_2 values are ranked in descending order. Then, the area under the polygonal, $A(F)$, is calculated by the mean value of the following

Algorithm 1 The $\text{pa}\epsilon$ -dominance algorithm - Part 1

Require: $T \leftarrow$ number of solutions given by user

```
1: procedure  $\text{PA}\epsilon\text{-DOMINANCE\_GRID}$  ( $P_{nP\text{oints}}, nP\text{oints}, nO\text{bjectives}$ )
2:    $\text{maxValue}_f[i] \leftarrow P_{nP\text{oints}}$   $\triangleright$  maximum values per each nObjectives
3:    $\text{minValue}_f[i] \leftarrow P_{nP\text{oints}}$   $\triangleright$  minimum values per each nObjectives
4:   if  $nO\text{bjectives} == 2$  then
5:      $\text{area} \leftarrow \text{Area}(P_{nP\text{oints}}, nP\text{oints})$   $\triangleright$  Calculate area
6:      $p \leftarrow \text{get\_P}(\text{area}, nO\text{bjectives})$ 
7:   else
8:      $\text{hyper} \leftarrow \text{HyperVolume}(P_{nP\text{oints}}, nP\text{oints}, nO\text{bjectives})$   $\triangleright$ 
       Hypervolume
9:      $p \leftarrow \text{get\_P}(\text{hyper}, nO\text{bjectives})$ 
10:  end if
11:   $v \leftarrow \text{Speed\_variation}(p, T)$   $\triangleright$  Calculate Speed Variation
12:   $\text{aux} \leftarrow \{(p^v - 1) \cdot p^{T-1}\} / \{p^{(T \cdot v)} - 1\}$ 
13:  for  $i \leftarrow 0, nO\text{bjectives}$  do  $\triangleright$  Calculate First  $\epsilon_1$  in each dimension
14:     $\epsilon_1^i \leftarrow \text{abs} \|\text{maxValue}_f[i] - \text{minValue}_f[i]\| \cdot \text{aux};$ 
15:  end for
16: end procedure

17: procedure  $\text{GET\_P}(\text{hyper}, nO\text{bjectives})$ 
18:  if  $nO\text{bjectives} == 2$  then
19:     $\text{fp} \leftarrow \text{openFile } p.\text{txt}$   $\triangleright$  generated for 2 objectives
20:  else
21:     $\text{fp} \leftarrow \text{openFile } 3p.\text{txt}$   $\triangleright$  generated for 3 objectives
22:  end if
23:  repeat  $\triangleright$ 
24:     $\text{line} \leftarrow \text{ReadNextLine}(\text{fp})$   $\triangleright$  read line. eg: {0.5, 0.1667}
25:    if  $\text{hyper} < \text{line}_2$  then  $\triangleright$   $\text{line}_2 = \text{Second value of line. eg: 0.1667}$ 
26:      return  $\text{lastline}_1 + \frac{(\text{hyper} - \text{lastline}_2) \cdot (\text{line}_1 - \text{lastline}_1)}{(\text{line}_2 - \text{lastline}_2)}$ 
27:    end if
28:     $\text{lastline} \leftarrow \text{line}$ 
29:  until  $\neg (\text{endofFile } fp)$ 
30: end procedure

31: procedure  $\text{SPEED\_VARIATION}(p, T)$ 
32:   $\text{low} \leftarrow 0.001$ 
33:   $\text{up} \leftarrow 1.0$ 
34:   $\text{lowsign} \leftarrow \text{Dichotomy\_fun}(p, t, \text{low});$ 
35:  repeat
36:     $\text{medium} \leftarrow ((\text{up} + \text{low}) / 2.0)$ 
37:     $\text{auxsign} \leftarrow \text{Dichotomy\_fun}(p, t, \text{medium})$ 
38:    if  $(\text{auxsign} == \text{lowsign})$  then
39:       $\text{low} \leftarrow \text{medium}$ 
40:    else
41:       $\text{up} \leftarrow \text{medium}$ 
42:    end if
43:  until  $((1 - 0.001) > 14)$  11
44:  return  $((\text{up} + \text{low}) / 2.0);$ 
45: end procedure
```

Algorithm 2 The pae-dominance algorithm - Part 2

```
1: procedure DICHOTOMY_FUN( $p, T, v$ )
2:    $\text{fun} \leftarrow ((1 - 2^{(1/p)}) \cdot p^{(T \cdot x)} + 2^{(1/p)} \cdot p^{(T \cdot x/2)} - 1$ 
3:   if ( $0 < v$ ) then
4:     return 1
5:   else
6:     return 0
7:   end if
8: end procedure

9: procedure AREA( $P_{nPoints}, nPoints$ )
10:   $\text{area} \leftarrow 0$ 
11:   $\text{upperArea} \leftarrow 0$ 
12:   $\text{lowerArea} \leftarrow 0$ 
13:  Sort  $P_{nPoints}$  in the first objective value
14:  for  $i \leftarrow 1, nPoints$  do  $\triangleright$  Calculate both areas among the  $nPoints$ 
    solutions
15:     $\text{lowerArea} \leftarrow \text{lowerArea} + \text{abs}((P_{i+1}.f[0] - P_i.f[0]) \cdot (P_{i+1}.f[1] -$ 
       $P_{nPoints-1}.f[1]))$ 
16:     $\text{upperArea} \leftarrow \text{upperArea} + \text{abs}((P_{i+1}.f[0] - P_i.f[0]) \cdot (P_i.f[1] -$ 
       $P_{nPoints-1}.f[1]))$ 
17:  end for
18:   $\text{area} = (\text{lowerArea} + \text{upperArea}) / 2$   $\triangleright$  Area between upper and lower
    areas
19:  return area
20: end procedure

21: procedure HYPERVOLUME( $P_{nPoints}, nPoints, nObjectives$ )
22:   $\text{hypervolume} \leftarrow 0$ 
23:  Sort  $P_{nPoints}$  in the third objective value
24:  for  $j \leftarrow 1, nPoints$  do
25:    Project  $P_{nPoints}$  onto two-objective values
26:    Check nondominated solutions in  $P_{nPoints}$   $\triangleright$  (as maximization
      problem)
27:     $\text{area} \leftarrow \text{Area}\{P_{nPoints}, nPoints\}$ 
28:     $\text{depth} \leftarrow P_j - P_{j-1}$ 
29:     $\text{hypervolume} \leftarrow \text{hypervolume} + (\text{area} * \text{depth})$ 
30:    Remove  $P_j$  from  $P_{nPoints}$ 
31:  end for
32:  return hypervolume
33: end procedure
```

lower, $LA(F)$, and upper, $UA(F)$, approximation areas:

$$LA(F) = \sum_{i=1}^{|F|-1} (f_1^{i+1} - f_1^i) f_2^{i+1},$$

and

$$UA(F) = \sum_{i=1}^{|F|-1} (f_1^{i+1} - f_1^i) f_2^i.$$

From these areas, $A(F)$ is

$$A(F) = \frac{LA(F) + UA(F)}{2}.$$

For the three-dimensional case, the difficulty increases because points cannot be fully ranked. So, we propose the following procedure:

Initially, the $nPoints$ points are sorted by their values in the third objective value. These values are then used to make *slices*. Each slice has a hypervolume in the first 2 objectives (Area). This area is calculated and it is multiplied by its depth in the third objective; then, the values obtained are summed up to obtain the total hypervolume of the $nPoints$ points.

Each slice in the hypervolume contains a different number of points, because at each slice, it is removed the lowest value point in the third objective. However, not all the points in each slice contribute to the Area in that slice. Some points may be dominated in the first two objective values and contribute nothing. So, it is important to re-check dominance (as a maximization problem) in the first two objective values (not including the third objective) by each slice to calculate the hypervolume (Area). This procedure it is illustrated in Algorithm 2 (see line 21).

4 Validation of our Proposed Approach

In order to validate our proposed $pa\epsilon$ -dominance, we adopted three algorithms: Two of them use ϵ -dominance, and in one of them, such a mechanism is replaced by our $pa\epsilon$ -dominance to make the third algorithm. This will allow us to show also the performance of a same algorithm with and without $pa\epsilon$ -dominance. The three multi-objective evolutionary algorithms adopted for our experimental study are the following:

1. **ϵ -MyDE**: This approach was proposed by Santana-Quintero and Coello (2005), and it consists of an extension of the differential evolution algorithm (Storn and Price, 1997) used to solve multi-objective optimization problems. The operators typically adopted in differential evolution are incorporated into this approach (Price et al., 2005), but the algorithm is extended with a secondary population which is used to retain the non-dominated solutions obtained during the evolutionary process. Also, ϵ -dominance is incorporated in order to get a well-distributed set of solutions along the Pareto front.

Parameter	ϵ -MyDE	ϵ -MOEA	pa ϵ -MyDE
P	100	100	100
NP	100 (approx)	100(approx)	100 (approx)
G_{max}	75	75	75
P_c	0.95	1.0	0.95
P_m	1/nVar	1/nVar	1/nVar
F	0.5	nr	0.5

nr = not required

Table 1: Parameters used by the algorithms compared.

2. **ϵ -MOEA**: This approach was proposed by Deb et al. (Deb et al., 2003; Deb et al., 2005a), and it consists of a steady-state genetic algorithm which maintains an archive of nondominated individuals. Note however, that this algorithm does not use the Pareto dominance relation when updating the archive. Instead, it uses the ϵ -dominance relation. One parent is selected from the main population and other from the archive. Then, an offspring is produced and it is allowed to enter into the archive if ϵ -dominates at least one of elements of the archive, and if no archive member ϵ -dominates it.
3. **pa ϵ -MyDE**: This is a modification of the ϵ -MyDE approach indicated above, in which we include pa ϵ -dominance instead of the regular ϵ -dominance concept.

Table 1 summarizes the parameter settings adopted for all the algorithms compared. In Table 1, P refers to the population at each generation, G_{max} is the total number of generations (or iterations) to be performed. Note that all the algorithms perform the same number of objective function evaluations: 7,500 for all test problems. NP is the number of solutions expected by each algorithm; this parameter is controlled by the value of ϵ . F is a parameter applicable only to differential evolution. P_c and P_m are the crossover and mutation rates, respectively.

4.1 Test functions and metrics

We chose five continuous (unconstrained) test problems with different geometrical characteristics for our experimental study. Note that our choice of problems was directed by the geometrical characteristics of the Pareto fronts rather than by the difficulty of solving each test problems, since our goal is to show the advantages of our pa ϵ -dominance scheme over the original ϵ -dominance.

The problems selected are the following: Deb11 (convex and bimodal) and Deb52 (the Pareto front is concave) from (Deb, 1999); Kursawe's problem (Kursawe, 1991) (the Pareto front is disconnected); ZDT1 (multimodal problem)

Test Function	NObj	NVar	Type	Characteristics
Deb11	2	2	$\min(f_1, f_2)$	Convex, bimodal
Deb52	2	2	$\min(f_1, f_2)$	Concave
Kursawe	2	3	$\min.(f_1, f_2)$	Disconnected
ZDT1	2	30	$\min.(f_1, f_2)$	Multimodal
DTLZ2	3	12	$\min(f_1, f_2, f_3)$	Concave, three-dimensional

Table 2: **NObj** denotes the number of objectives, **NVar** the number of decision variables, **Type** specifies the type of optimization problem (maximization or minimization) and **Characteristics** provides a summary of the geometrical characteristics of the Pareto front.

from (Zitzler et al., 2000); and DTLZ2 from (Deb et al., 2005b) (a three-objective problem). Tables 2 and 3 show further details of these problems.

The main goal of pae -dominance is to obtain as many Pareto optimal solutions as possible (up to the maximum capacity of the grid), but within a homogeneous spread. Thus, the performance measures adopted in our study are focused on such aspects:

Number of points: It shows us how far the number of solutions found is from the maximum capacity of the grid. In all our experiments, the grid was defined with a capacity of 100 points. So, the closer to 100 that an algorithm gets, the better the value of this performance measure.

Chi-Square-Like Deviation Measure: This metric was proposed by Srinivas and Deb (1994) to measure the diversity of the set of solutions obtained. Solutions are compared with respect to a uniformly distributed set of the true Pareto front. Let P be the set of vectors uniformly distributed along the Pareto-optimal front and F the set of solutions to be compared. Then, for each $i \in \{1, \dots, |P|\}$, let us denote by n_i the number of solutions in F whose distance from i is less than δ (δ is set beforehand and we use Euclidean distance). Then, the deviation is measured like a Chi-square distribution such as

$$\chi = \sqrt{\sum_{i=1}^{|P|+1} \left(\frac{n_i - \bar{n}_i}{\sigma_i} \right)^2}.$$

The ideal distribution is achieved when all of the neighborhoods of points in P have the same number of vectors, that is, if for each point in P there are $\bar{n}_i = |F|/|P|$ points in F whose distance from this vector is less than δ . Then $\chi = 0$. The variance σ_i^2 is proposed to be $\sigma_i^2 = \bar{n}_i(1 - \frac{\bar{n}_i}{|F|})$, for all $i \in \{1, 2, \dots, |P|\}$. Index $i = |P| + 1$ is used for those points that are far from all points in P . For this index, $\bar{n}_{|P|+1} = 0$ and $\sigma_{|P|+1}^2 = |F|(1 - \frac{1}{|P|})$ are also proposed in (Srinivas and Deb, 1994). Then, it easy to see that

Test Function	Objectives	Bounds
Deb11	$f_1(x_1) = x_1$ $f_2(x_1, x_2) = \frac{1}{x_1} \left(2 - e^{-\left(\frac{x_2-0.2}{0.004}\right)^2} - 0.8e^{-\left(\frac{x_2-0.6}{0.4}\right)^2} \right)$	$0 \leq x_i \leq 1$ $i = 1, 2$
Deb52	$f_1(x_1) = 1 - e^{-4x_1} \sin^4(10\pi x_1)$ $f_2(x_1, x_2) = g(x_2) * h(x_1)$, where $g(x_2) = 1 + x_2^2$ and $h(x_1) = \begin{cases} 1 - \left(\frac{f_1(x_1)}{g(x_2)}\right)^{10} & \text{if } f_1(x_1) \leq g(x_2) \\ 0 & \text{otherwise.} \end{cases}$	$0 \leq x_i \leq 1$ $i = 1, 2$
Kursawe	$f_1(x_1, x_2) = \sum_{i=1}^2 -10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x_1, x_2) = \sum_{i=1}^2 \left(x_i ^{0.8} + 5 \sin(x_i^3) \right)$	$-5 \leq x_i \leq 5$ $i = 1, 2, 3$
ZDT1	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}, g) = 1 - \sqrt{f_1/g(\vec{x})}$ where: $g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$	$n = 30$ $0 \leq x_i \leq 1$ $i = 1, 2, \dots, 30$
DTLZ2	$f_1(\vec{x}) = \cos(\frac{\pi}{2}x_1) \cos(\frac{\pi}{2}x_2)(1 + g(\vec{x}))$ $f_2(\vec{x}) = \cos(\frac{\pi}{2}x_1) \sin(\frac{\pi}{2}x_2)(1 + g(\vec{x}))$ $f_3(\vec{x}) = \sin(\frac{\pi}{2}x_1)(1 + g(\vec{x}))$ where $g(\vec{x}) = \sum_{i=1}^{12} (x_i - 0.5)^2$	$0 \leq x_i \leq 1$ $i = 1, \dots, 12$

Table 3: Objective functions and bounds of the decision variables for each of the test problems adopted for our experimental study.

$0 \leq \chi < \infty$ and the lower the χ value the better the distribution of F with respect to P . The parameter δ depends on P and it is crucial for the final χ value. Neighborhoods must be disjoint, so we take δ as a half of the minimum distance between two points in P .

Spread: Deb et al. (2002) proposed the metric Δ with the idea of measuring both progress towards the Pareto-optimal front and the extent of spread. To this end, if P is a subset of the Pareto-optimal front, Δ is defined as follows

$$\Delta = \frac{\sum_{i=1}^m d_i^e + \sum_{i=1}^{|F|} |d_i - \bar{d}|}{\sum_{i=1}^m d_i^e + |F| \bar{d}}.$$

where d_i^e denotes the distance between the i -th coordinate for both extreme points in P and F , and d_i measures the distance of each point in F to its closer point in F . For our experiments, we use the crowding distance for d_i (see (Deb, 2001) for more details on this distance). Nevertheless, other types of measures could be used for d_i .

From the above definition, it is easy to conclude that $0 \leq \Delta \leq 1$ and the lower the Δ value, the better the distribution of solutions. A perfect distribution, that is $\Delta = 0$, means that the extreme points of the Pareto-optimal front have been found and d_i is constant for all i .

Standard Deviation of the Crowding Distances: Trying to get more information related with the crowding distance, we include its standard deviation:

$$SDC = \sqrt{\frac{1}{|F|} \sum_{i=1}^{|F|} (d_i - \bar{d}_i)^2}$$

Now, $0 \leq SDC \leq \infty$ and the lower the value of SDC , the better the distribution of vectors in F . \bar{d}_i is the mean value of all d_i . Again, a perfect distribution, that is $SDC = 0$, means that d_i is constant for all i .

5 Discussion of Results

In this section, we compare the performance of our proposed $pa\epsilon$ -dominance using the aforementioned algorithms and test functions. In Tables 4, 5, 6, 7 and 8, for each performance measure considered, its mean value, standard deviation and the maximum and minimum value over 30 independent runs. We emphasize the best values using **boldface**.

Table 4 shows the results for the first problem considered (Deb11). It is worth mentioning that $pa\epsilon$ -MyDE achieved the best results in this case, not only regarding the distribution of solutions, but also with respect to the number of solutions retained (its average was 99.8 from a maximum of 100). In fact, the $pa\epsilon$ -MyDE obtained the best results with respect to all the performance

Algorithm		No. of points	Chi-Square	Spread	Crowding
$pa\epsilon$ -MyDE	Mean	99.833	7.714	0.230	0.009
	SDev	1.003	0.209	0.016	0.001
	Max	101	8.255	0.273	0.011
	Min	98	7.280	0.191	0.009
ϵ -MyDE	Mean	46.433	8.125	0.490	0.042
	SDev	1.086	0.216	0.016	0.001
	Max	50	8.668	0.526	0.047
	Min	45	7.832	0.464	0.038
ϵ -MOEA	Mean	49.900	8.581	0.559	0.038
	SDev	6.730	1.680	0.120	0.003
	Max	60	11.010	0.734	0.041
	Min	45	7.317	0.461	0.033

Table 4: Mean, standard deviation, maximum and minimum values over 30 runs for the first test problem (Deb11).

measures considered. In Figure 5, we show the Pareto fronts obtained by the three algorithms. This figure graphically shows that our approach ($pa\epsilon$ -MyDE) has benefited from adopting $pa\epsilon$ -dominance instead of ϵ -dominance.

Table 5 shows the results for the second test problem (Deb52). In this case, due to an almost horizontal region in the Pareto front, ϵ -dominance loses a big number of points. Although the number of points generated by $pa\epsilon$ -MyDE is also far from 100, it finds more than twice the number of points obtained by the two other algorithms adopting ϵ -dominance. Again, the $pa\epsilon$ -MyDE obtained the best results with respect to all the performance measures considered. In Figure 6, we show the Pareto fronts obtained by the three algorithms. Notice that $pa\epsilon$ -MyDE was able to find the extreme points despite the difficult geometrical characteristics of this Pareto front.

Table 6 shows the results for the third test problem (Kursawe). In this case, the performance measures are very similar for the three approaches compared, although our $pa\epsilon$ -MyDE outperformed the others with respect to two of them. The reason for this similar performance is that the p value associated to this problem is close to 1 and, as previously mentioned, ϵ -dominance and $pa\epsilon$ -dominance are almost the same as the p value gets close to 1. The number of points found is around 60 because this front is disconnected. In Figure 7 we show the Pareto fronts obtained by the three methods. In this case, $pa\epsilon$ -dominance and ϵ -dominance generate similar grids.

Table 7 shows the results for the fourth test problem (ZDT1). Again, our $pa\epsilon$ -MyDE obtained the best results with respect to all the performance measures considered. Again, the number of points retained is close to 100 and their distribution is quite good. In Figure 8 we show the Pareto fronts obtained by the three methods. Notice that $pa\epsilon$ -MyDE was able to find the extreme points despite the almost vertical region of this Pareto front.

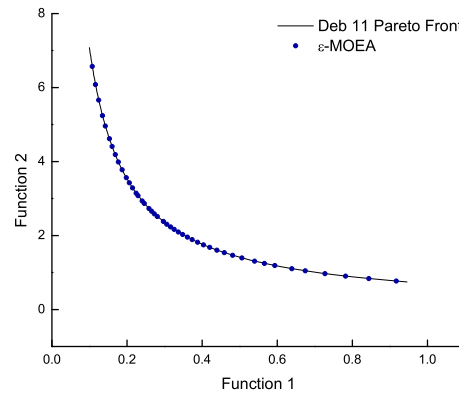
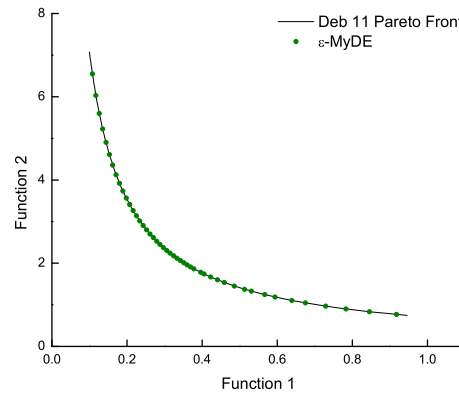
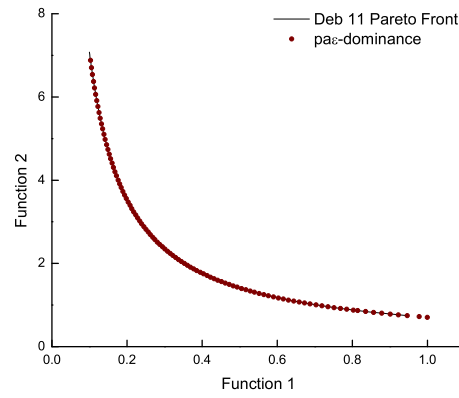


Figure 5: Efficient solutions generated by pae -MyDE (top), ϵ -MyDE (middle) and ϵ -MOEA (bottom) for the first test problem (Deb11).

Algorithm		No. of points	Chi-Square	Spread	Crowding
$pa\epsilon$ -MyDE	Mean	75.033	6.875	0.377	0.056
	SDev	1.494	0.1942	0.046	0.006
	Max	78	7.461	0.474	0.068
	Min	72	6.510	0.328	0.048
ϵ -MyDE	Mean	34.800	8.385	0.502	0.073
	SDev	0.833	0.100	0.031	0.007
	Max	37	8.646	0.585	0.083
	Min	33	8.165	0.452	0.058
ϵ -MOEA	Mean	33.600	8.494	0.534	0.105
	SDev	0.663	0.188	0.049	0.001
	Max	35	8.799	0.580	0.106
	Min	33	8.105	0.390	0.101

Table 5: Mean, standard deviation, maximum and minimum values over 30 runs for the second test problem (Deb52).

Algorithm		No. of points	Chi-Square	Spread	Crowding
$pa\epsilon$ -MyDE	Mean	63.733	6.113	0.351	0.036
	SDev	1.711	0.275	0.025	0.001
	Max	68	6.739	0.404	0.040
	Min	60	5.645	0.303	0.034
ϵ -MyDE	Mean	60.933	6.682	0.300	0.035
	SDev	1.031	0.207	0.022	0.001
	Max	63	7.303	0.339	0.039
	Min	59	6.401	0.241	0.033
ϵ -MOEA	Mean	57.433	6.653	0.327	0.035
	SDev	0.761	0.224	0.016	0.001
	Max	59	7.273	0.358	0.037
	Min	56	6.298	0.287	0.033

Table 6: Mean, standard deviation, maximum and minimum values over 30 runs for the third test problem (Kursawe).

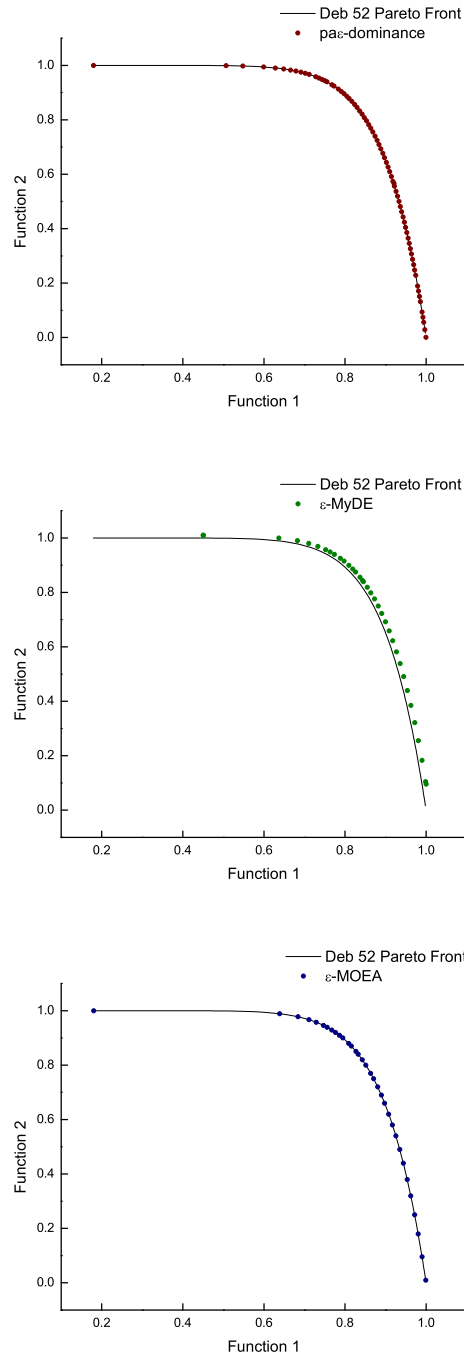


Figure 6: Efficient solutions generated by $pa\varepsilon$ -MyDE (top), ε -MyDE (middle) and ε -MOEA (bottom) for the second test problem (Deb52).

Algorithm		No. of points	Chi-Square	Spread	Crowding
$pa\epsilon$ -MyDE	Mean	99.366	4.684	0.158	0.009
	SDev	2.057	0.650	0.013	0.001
	Max	98	5.788	0.200	0.011
	Min	90	3.401	0.141	0.006
ϵ -MyDE	Mean	77.233	5.611	0.211	0.014
	SDev	1.605	0.317	0.015	0.002
	Max	81	6.392	0.240	0.023
	Min	74	5.067	0.181	0.010
ϵ -MOEA	Mean	75.266	5.975	0.220	0.012
	SDev	0.512	0.189	0.007	0.0005
	Max	77	6.505	0.242	0.014
	Min	75	5.675	0.205	0.011

Table 7: Mean, standard deviation, maximum and minimum values over 30 runs for the fourth problem (ZDT1).

Finally, Table 8 shows the results for the fifth test problem (DTLZ2). Again, best mean values are obtained by $pa\epsilon$ -dominance. In Figure 9 we show the Pareto fronts obtained. The new grid finds more points specially on the extreme areas of the Pareto front.

6 Conclusions and Future Work

In this paper, we have proposed an alternative approach for the ϵ -dominance (which we call $pa\epsilon$ -dominance) due to Laumanns et al. (2002). In our proposed scheme, we considered different ϵ -dominance regions depending on the geometrical characteristics of the Pareto-optimal front. In order to do this, each Pareto front is associated to one curve of the family

$$\{x^p + y^p = 1 : 0 \leq x, y \leq 1, 0 < p < \infty\}.$$

for bi-objective optimization problems, or

$$\{x^p + y^p + z^p = 1 : 0 \leq x, y, z \leq 1, 0 < p < \infty\}$$

for three dimensional problems. This way, we take advantage of the positive aspects of ϵ -dominance (already shown), while addressing some of its limitations.

On the one hand, $pa\epsilon$ -dominance finds a higher number of efficient points because the size of the boxes are adjusted specially in those areas where the Pareto front needs less solutions in any of its dimensions (almost horizontal or vertical regions of the Pareto front). Also, these solutions are better uniformly distributed along the Pareto front because the new grid balances the size of the boxes being more precise in those areas of the objective function space in which more solutions are needed.

Algorithm		No. of points	Chi-Square	Spread	Crowding
$pa\epsilon$ -MyDE	Mean	82.633	11.403	0.461	0.043
	SDev	16.113	0.442	0.060	0.011
	Max	123	12.673	0.598	0.076
	Min	61	10.775	0.367	0.026
ϵ -MyDE	Mean	69.433	11.7144	0.560	0.060
	SDev	3.008	0.127	0.065	0.010
	Max	77	11.300	0.664	0.081
	Min	64	10.786	0.444	0.046
ϵ -MOEA	Mean	60.033	11.838	0.523	0.067
	SDev	2.168	0.092	0.049	0.007
	Max	64	11.199	0.633	0.086
	Min	56	10.688	0.426	0.057

Table 8: Mean, standard deviation, maximum and minimum values over 30 runs for the fifth test problem (DTLZ2).

Three evolutionary multiobjective algorithms are used to show the effectiveness of our proposed scheme: ϵ -MyDE and ϵ -MOEA, which use ϵ -dominance as their diversification mechanism and $pa\epsilon$ -MyDE, which consists of the ϵ -MyDE approach, but adopting $pa\epsilon$ -dominance instead of ϵ -dominance.

In order to assess the performance of our proposed $pa\epsilon$ -dominance, we solved five test problem with different geometrical characteristics and used three standard metrics designed to measure diversity properties and one more measure related to the number of points found. In all cases, $pa\epsilon$ -dominance has been shown more efficient in getting a higher number of nondominated solutions with a better spread. Thus, we conclude that $pa\epsilon$ -dominance is an advantageous alternative to ϵ -dominance, particularly when the Pareto front has geometrical characteristics that cause difficulties to ϵ -dominance.

As part of our future work, we plan to generalize our proposal, so that we can drop our symmetry hypothesis assumed in the curves of the form $x^p + y^p = 1$. This would certainly be more unstable than the current proposal, but we believe that such instability can be controlled using a different way of determining the values of p , and q (and r , if dealing with a three-objective problem), for a given Pareto front. Disconnected Pareto front also require a more in-depth analysis, since they deserve a special treatment when using relaxed forms of Pareto dominance such as ϵ -dominance or our proposed scheme. Additionally, we are also looking into ways of using our proposed scheme to handle the user's preferences in an interactive way (Coello Coello, 2000; Coello Coello et al., 2002).

Acknowledgements

The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Section of the Electrical Engineering Department at CINVESTAV-IPN. The third author gratefully acknowledges support from CONACyT through project 42435-Y.

References

- Coello Coello, C. A. (2000). Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *2000 Congress on Evolutionary Computation*, volume 1, pages 30–37, Piscataway, New Jersey. IEEE Service Center.
- Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York. ISBN 0-3064-6762-3.
- Deb, K. (1999). Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230.
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK. ISBN 0-471-87339-X.
- Deb, K., Mohan, M., and Mishra, S. (2003). Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In Fonseca, C. M., Fleming, P. J., Zitzler, E., Deb, K., and Thiele, L., editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 222–236, Faro, Portugal. Springer. Lecture Notes in Computer Science. Volume 2632.
- Deb, K., Mohan, M., and Mishra, S. (2005a). Evaluating the ϵ -Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501–525.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005b). Scalable Test Problems for Evolutionary Multiobjective Optimization. In Abraham, A., Jain, L., and Goldberg, R., editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA.
- Kursawe, F. (1991). A Variant of Evolution Strategies for Vector Optimization. In Schwefel, H. P. and Männer, R., editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science Vol. 496*, pages 193–197, Berlin, Germany. Springer-Verlag.

- Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282.
- Price, K. V., Storn, R. M., and Lampinen, J. A. (2005). *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin. ISBN 3-540-20950-6.
- Rao, S. S. (1996). *Engineering Optimization*. John Wiley & Sons, third edition.
- Reyes Sierra, M. and Coello Coello, C. A. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ -Dominance. In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 505–519, Guanajuato, México. Springer. Lecture Notes in Computer Science Vol. 3410.
- Santana-Quintero, L. V. and Coello Coello, C. A. (2005). An Algorithm Based on Differential Evolution for Multiobjective Problems. In Dagli, C. H., Buczak, A. L., Enke, D. L., Embrechts, M. J., and Ersoy, O., editors, *Smart Engineering System Design: Neural Networks, Evolutionary Programming and Artificial Life*, volume 15, pages 211–220, St. Louis Missouri, USA.
- Srinivas, N. and Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248.
- Storn, R. and Price, K. (1997). Differential Evolution - A Fast and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359.
- Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195.

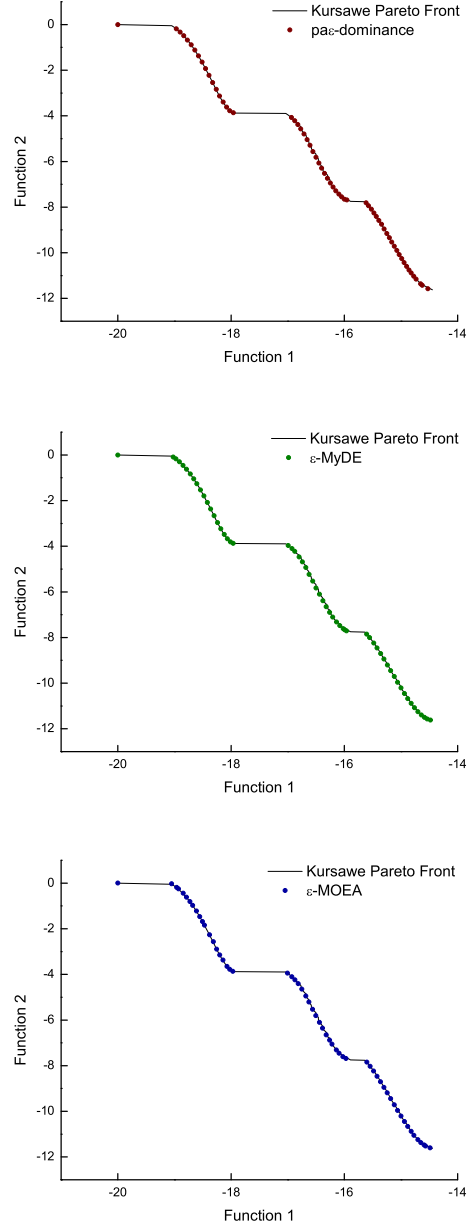


Figure 7: Efficient solutions generated by pae -MyDE (top), ε -MyDE (middle) and ε -MOEA (bottom) for the third test problem (Kursawe).

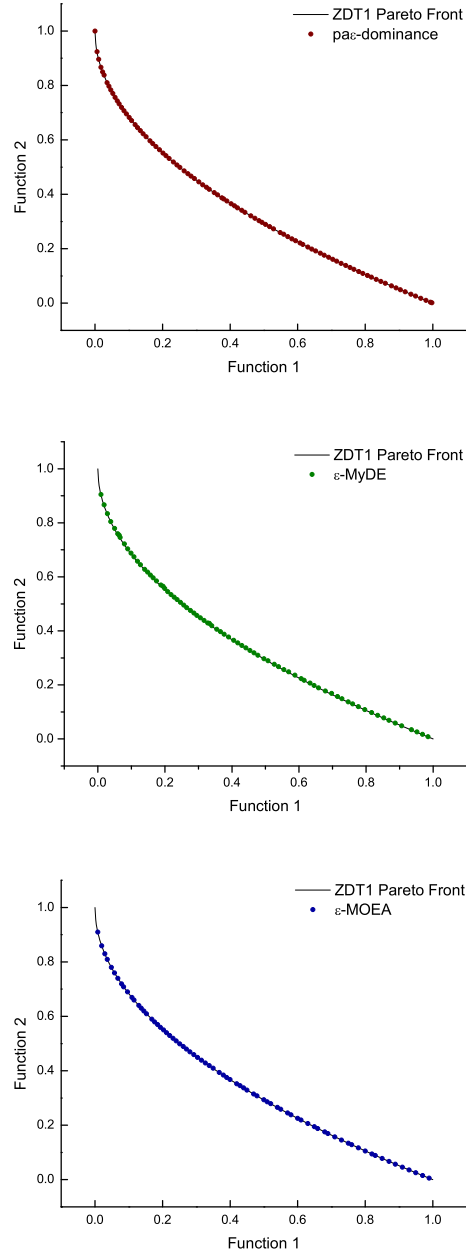


Figure 8: Efficient solutions generated by $p\epsilon$ -MyDE (top), ϵ -MyDE (middle) and ϵ -MOEA (bottom) for the fourth test problem (ZDT1).

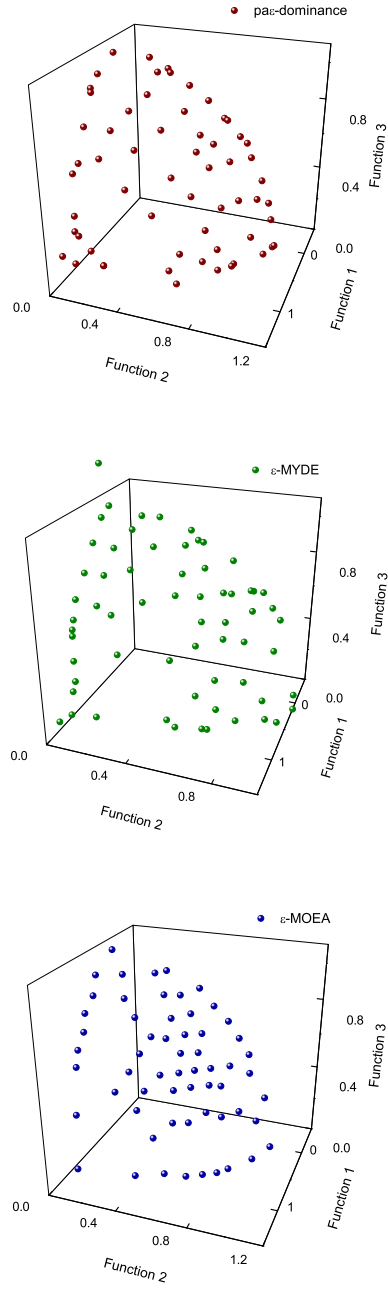


Figure 9: Efficient solutions generated by $pa\varepsilon$ -MyDE (top), ε -MyDE (middle) and ε -MOEA (bottom) for the fifth test problem (DTLZ2).