Improving the Efficiency of ϵ-dominance Based Grids

Alfredo G. Hernández-Díaz
agarher@upo.es
Department of Economics, Quantitative Methods and
Economic History. Pablo de Olavide University, Seville, Spain

Luis V. Santana-Quintero
lsantana@gmu.edu
Krasnow Institute, George Mason University
Fairfax, VA 22030, USA

Carlos A. Coello Coello
ccoello@cs.cinvestav.mx
CINVESTAV-IPN, Computer Science Department
(Evolutionary Computation Group)
México D.F., México

Julián Molina
julian.molina@uma.es
Department of Applied Economics (Mathematics)
University of Málaga, Málaga, Spain

Rafael Caballero
rafael.caballero@uma.es
Department of Applied Economics (Mathematics)
University of Málaga, Málaga, Spain

January 10, 2011

Abstract

In this paper, we deal with the problem of handling solutions in an external archive with the use of a relaxed form of Pareto dominance called ϵ-dominance and a variation of it called paϵ-dominance. These two relaxed forms of Pareto dominance have been used as archiving strategies in some multi-objective evolutionary algorithms (MOEAs). The main objective of this work is to improve the ϵ-dominance based schemes to handle nondominated solutions, or to retain nondominated solutions in an external archive. Thus, our main contribution is to add an extra objective function only at the time of accepting a nondominated solution into the external archive, in order to preserve some solutions which are normally lost when using any of the aforementioned relaxed forms of Pareto dominance. Such a proposal is inexpensive (computationally speaking) and quite effective, since it is able to produce Pareto fronts of much better quality than the aforementioned archiving techniques.

Keywords: Epsilon-Dominance; Pae-Dominance; Pareto Dominance; Multi-Objective Optimization; Evolutionary Algorithms

1 Introduction

In many disciplines, optimization problems have two or more objectives, which are normally in conflict with one another, and that we wish to optimize simultaneously. These problems are called "multi-objective", and give rise to a set of solutions (called the Pareto optimal set) that represent the best possible trade-offs among all the objectives, such that no objective can be improved without worsening another. The vectors corresponding to the solutions in the Pareto optimal set are said to be nondominated with respect to each other. The objective function values of the solutions contained in the Pareto optimal set constitute the so-called Pareto front of the problem.

In the absence of preference information from the user, all the solutions contained in the Pareto optimal set are equally good. However, since this set could be very large (or infinite, if dealing with continuous search spaces), in practice, only a few of such solutions are actually maintained. Thus, ideally, the search engine adopted to generate such nondominated solutions should be able to provide a reduced set in a way such that they are both optimal and well-distributed along the Pareto front. This will allow the decision

maker (DM) to have sufficient information to choose only one (or very few) solution(s) from that set.

In the last few years, researchers have developed powerful multi-objective optimizers based on evolutionary algorithms. Most of these multi-objective evolutionary algorithms (MOEAs) incorporate an external archive in which the nondominated solutions generated during the search are stored. A nondominated solution is allowed to enter such an external archive only if it either dominates some solution from the archive (in that case, the dominated solution is deleted) or if it is nondominated with respect to its contents. Thus, external archives impose an elitist mechanism to MOEAs, since they allow the storage of the solutions that are globally (i.e., with respect to all the solutions generated so far by the MOEA) nondominated. However, and mainly because of practical issues (as indicated before), external archives tend to be bounded in their maximum size. This has motivated the development of techniques that enforce a good distribution of solutions within a bounded external archive. The most popular approaches are: clusters [29], adaptive grids [15], crowding [3], entropy [9] and the use of relaxed forms of Pareto dominance [11, 17, 16].

Thus, the main aim of this work is to show how a relatively simple modification in the classical relaxed Pareto dominance relation adopted by filtering schemes such as ϵ-dominance [17] or paϵ-dominance [11] results in a remarkable improvement in performance. The proposed modification helps to the preservation of solutions lying at the extreme parts of the Pareto front (such solutions are normally lost when using ϵ-dominance) and the computation of an appropriate ϵ value is thus better controlled.

We propose three different schemes whose use depends on the user's preferences. These schemes can be easily implemented for any sort of Pareto front:

Scheme 1: When the user provides the number of desired nondominated solutions but such limit can be exceeded.

Scheme 2: When the user provides the number of desired nondominated solutions and is able to incorporate information about the geometric characteristics of the Pareto front.

Scheme 3: When the user provides the number of desired nondominated solutions and does not want to exceed it, but there is no information about the geometric characteristics of the Pareto front. In this case,

taking into account the minimum and the maximum capacity of the grid, the ϵ value is adjusted in order to match the expected number of solutions that the user wants.

The remainder of this paper is organized as follows. In Section 2, we present some basic concepts related to multi-objective optimization. Then, in Section 3 we present others schemes used to handle nondominated solutions. In Section 4, we present our proposed mechanism. Then, in Section 5 we show the results obtained. Finally, in Section 6 we provide our conclusions as well as some possible paths for future research.

2 Multi-Objective Optimization

Without loss of generality, the problems that we will deal with in this paper are unconstrained multi-objective optimization problems. However, the proposed method can also be used in constrained multi-objective problems, since such an approach is independent of the constraint-handling mechanism adopted by the search engine. The (unconstrained) multi-objective optimization problem (MOP) can be formally defined as the problem of finding $\vec{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)^T$ which optimizes the vector function:

$$\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x}))^T .$$

In other words, we aim to determine those points that yield the optimum values for all the k objective functions simultaneously.

Pareto dominance (assuming minimization) is formally defined as follows:

A vector $\vec{u} = (u_1, \ldots, u_k)$ is said to dominate a vector $\vec{v} = (v_1, \ldots, v_k)$ if and only if $\vec{u}$ is partially less than $\vec{v}$, i.e., $\forall i \in \{1, \ldots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\}$ such that $u_i < v_i$.

In order to say that a solution dominates another one, such a solution needs to be strictly better in at least one objective, and not worse in any of them. So, when we are comparing two different solutions, A and B, there are 3 possibilities: A dominates B, A is dominated by B or A and B are incomparable.

The formal definition of *Pareto optimality* is the following:

A solution $\vec{x_u} \in S$ is said to be Pareto optimal if and only if there is no $\vec{x_v} \in S$ for which $\vec{v} = f(\vec{x}_v) = (v_1, \ldots, v_k)$ dominates $\vec{u} = f(\vec{x}_u) = (u_1, \ldots, u_k)$.

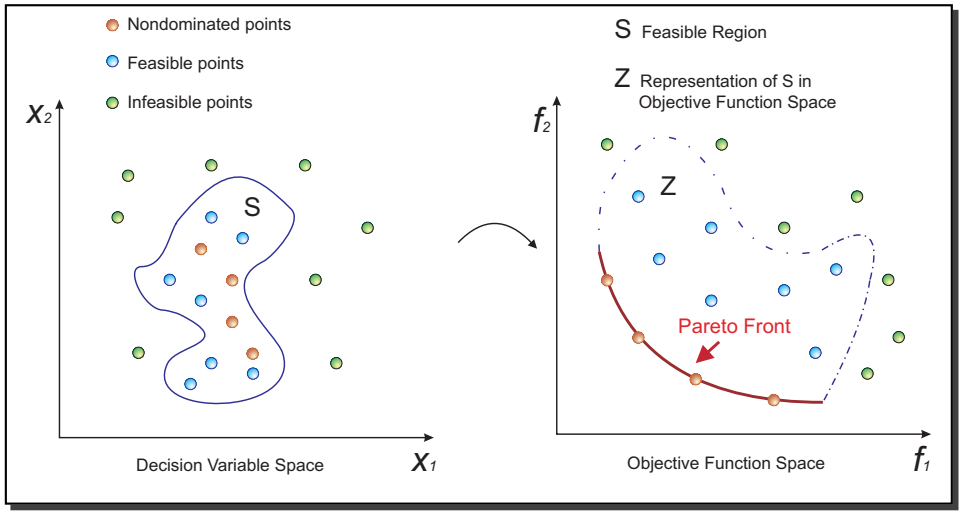


Figure 1: Mapping of the Pareto optimal solutions to the objective function space.

In words, this definition says that $\vec{x}_u$ is Pareto optimal if there exists no vector $\vec{x_v}$ which would decrease some objective without causing a simultaneous increase in at least one other objective.

This definition does not provide us a single solution (in decision variable space), but a set of solutions which form the so-called *Pareto Optimal Set* (P^*). The vectors that correspond to the solutions included in the Pareto optimal set are *nondominated*.

When all nondominated solutions are plotted in objective function space, the nondominated vectors are collectively known as the *Pareto Front* (PF^*). Formally:

$$PF^* := \{\vec{f}(\vec{x}) = (f_1(\vec{x}), \ldots, f_k(\vec{x}))|\vec{x} \in P^*\}.$$

It is, in general, impossible to find an analytical expression that defines the Pareto front of a problem, so the most common way to get the Pareto front is to compute a sufficient number of points in the feasible region, and then filter out the nondominated vectors from them.

The previous definitions are graphically depicted in Figure 1 for a general constrained MOP, showing the *Pareto front*, the *Pareto optimal set* and *dominance* relations among solutions.

3 Handling Well-Distributed Solutions in External Archives

As indicated before, over the years, a variety of mechanisms have been proposed in order to enforce a good distribution of solutions (normally in objective function space) stored in an external archive. The most popular of such mechanisms will be briefly discussed next.

3.1 Adaptive Grids

This mechanism was first incorporated in the Pareto Archived Evolution Strategy (PAES), proposed by Knowles and Corne [15]. PAES is a simple (1+1)-evolution strategy which consists of a single parent generating a single offspring through the use of mutation. PAES uses an external archive (with an upper bound on its size) that contains all the nondominated solutions generated so far. Each solution generated by PAES is a candidate to be accepted in the external archive which uses an adaptive hyper-grid in objective function space (see Figure 2) to divide it into several hyper boxes. The adaptive grid is really a space formed by hypercubes. An integer vector is used to refer to such hypercubes, where these integer vectors have as many components as objective functions has the problem to be solved. Each hypercube can be interpreted as a geographical region that contains an n number of individuals. The adaptive grid allows us to store nondominated solutions and to redistribute them when its maximum capacity is reached. In the case in which an offspring solution is nondominated by the reference set, another solution that resides in the most crowded region is removed from the external archive.

Over the years, a number of MOEAs have adopted variations of the adaptive grid (see for example [1, 2]).

3.2 Crowding Distance

This mechanism was originally proposed by Deb et. al for the Nondominated Sorting Genetic Algorithm–II (NSGA-II) [7]. NSGA-II ranks solution based on Pareto dominance (using a procedure called nondominated sorting). For each ranking level, a crowding distance between two solutions is estimated by calculating the sum of the Euclidean distances between the two neighboring

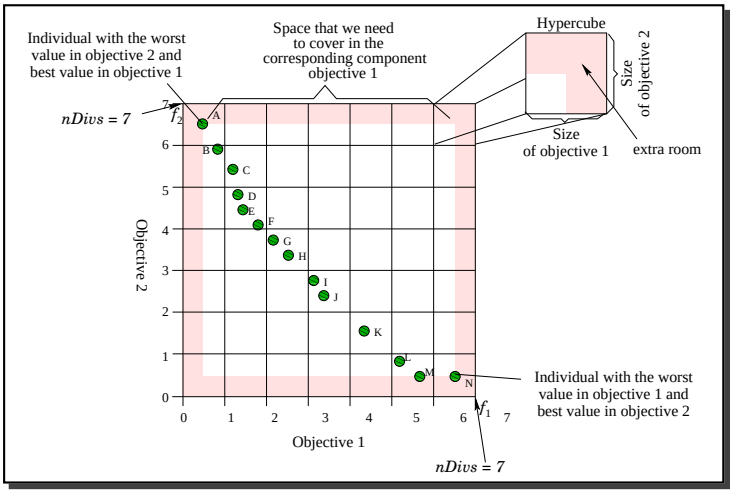


Figure 2: Adaptive grid in PAES.

solutions from either side of the solution along each of the objectives (see Figure 3).

Although NSGA-II does not use an external archive because of its $\mu + \lambda$ selection scheme (which is implicitly elitist), several researchers have adopted variations of the crowding comparison operator of NSGA-II to distribute solutions in an external archive (see for example [22, 26]).

3.3 Clustering

The Strength Pareto Evolutionary Algorithm (SPEA) proposed by Zitzler et al. [29] adopts a clustering technique called "average linkage method" [18] to prune the contents of its bounded external archive. In SPEA, the external archive participates in the selection process. Thus, if its size grows too large, it might reduce the selection pressure, which, consequently, slows down the search.

Other MOEAs have also adopted clustering techniques for maintaining diversity in their external archives and even in decision variable space (see for example [19, 14, 25]).

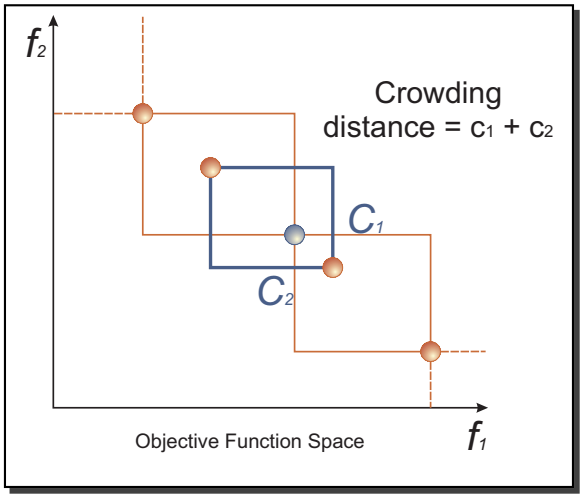


Figure 3: Crowding distance as a diversity operator used in NSGA-II.

3.4 Relaxed forms of Pareto dominance

ϵ-dominance is a relaxed form of Pareto dominance proposed by Laumanns et al. [17]. Its most common version (the additive one) is defined as follows: Let $f, g \in \mathbb{R}^k$. Then f is said to ϵ-dominate g for some $\epsilon > 0$, if and only if $\epsilon + f_i \geq g_i$, for all $i \in \{1, \ldots, m\}$.

The so-called ϵ-Pareto set is an archiving strategy that maintains a subset of generated solutions. It guarantees convergence and diversity according to well-defined criteria, namely the value of the ϵ parameter, which defines the resolution of the grid to be adopted for the secondary population. The general idea of this mechanism is to divide objective function space into boxes of size ϵ. Each box can be interpreted as a geographical region that contains a single solution. This algorithm is very attractive both from a theoretical and from a practical point of view. However, in order to achieve the best performance, it is necessary to provide the size of the box (the ϵ parameter) which is problem-dependent, and it is normally not known before executing a MOEA.

ϵ-dominance has been incorporated into several MOEAs from which the most famous is the so-called ϵ-MOEA [6].

In spite of its advantages, ϵ-dominance has several limitations, from which the following are the most important:

1. We can lose a high number of nondominated solutions if the decision

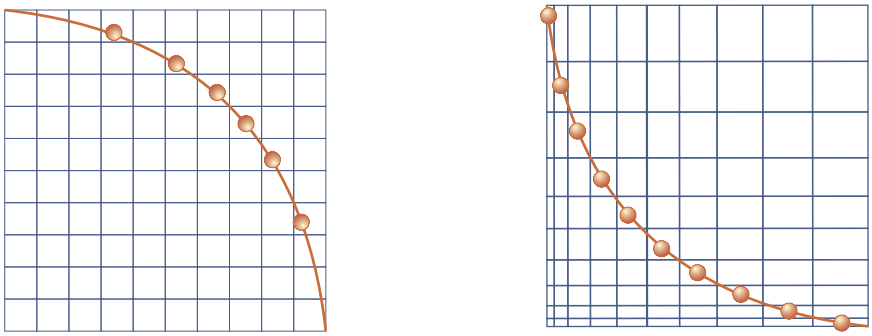


Figure 4: Uniform (left) an non-uniform (right) grids with 100 boxes (maximum capacity of 10 points) for $x^2 + y^2 = 1$ (left) and $x^{1/2} + y^{1/2} = 1$ (right). ϵ-dominance (left) allows a maximum of 6 points, whereas the paϵ-dominance grid (right) can retain the 10 solutions.

maker does not take into account (or does not know beforehand) the geometric 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 4.

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 true Pareto front is linear.

In order to address some of the problems previously described, Hernández-Díaz et. al proposed in [11] an alternative ϵ-dominance scheme, called Pareto adaptive ϵ-dominance (paϵ-dominance, for short). This scheme maintains the good properties of ϵ-dominance while overcoming its main limitations.

In that proposal, it is considered not only a different ϵ value for each objective but also the vector $\epsilon_j = \left(\epsilon_j^1, \epsilon_j^2, ..., \epsilon_j^m\right)$ associated to each f_j depends on the geometric characteristics of the Pareto optimal front. In other words, the approach takes into consideration different intensities of dominance for each objective according to the position of each point along the Pareto front.

Then, the size of the boxes is adapted depending on their corresponding area in objective function space, so that the boxes are smaller where needed (normally at the extremes of the Pareto front), and larger in other (less problematic) parts of the Pareto front, as can be seen at the right handside of Figure 4.

paϵ-dominance was originally incorporated into a hybrid MOEA based on differential evolution and rough sets called DEMORS (see [23]) and it has been adopted by other researchers (see for example [10]).

In spite of the improvements introduced by paϵ-dominance with respect to the original ϵ-dominance, this approach has some problems of its own. Namely, there are still problems in which paϵ-dominance is not able to maintain a good distribution of solutions at the very extreme parts of the Pareto front (as can be seen in Figure 11-(c) or in [11]). Evidently, this affects the distribution of solutions along the Pareto front and also has a negative effect on the performance of ϵ-dominance. Such problems were precisely the main motivation for the work reported here. Our proposal is to modify the selection mechanism shared by both ϵ-dominance and paϵ-dominance, and briefly described next.

In ϵ-dominance and paϵ-dominance, the objective function space is divided into hyper-boxes, and each solution in the ϵ-dominance grid is associated with an identification array $Box_c = (Box_{c,1}, Box_{c,2}, \ldots, Box_{c,k})$, where $Box_{c,1}, Box_{c,2}$, and $Box_{c,k}$ are integer values referring to the identification box assigned for ϵ-dominance or paϵ-dominance for each objective, where k is the number of objectives (see [17] or [11] for further details). The identification array works as a marker to identify the hyper-box in which the solution is in the ϵ-dominance grid. For instance, Figure 5 includes five points whose identification array values are $Box_{c_1} = (0, 7)$, $Box_{c_2} = (1, 5)$, $Box_{c_3} = (2, 3)$, $Box_{c_4} = (4, 2)$ and $Box_{c_5} = (6, 1)$. The minimum value of 0 is given to all the solutions whose objective function value is in the range $[0, \epsilon)$ and the value of 1 is given to the solutions whose objective function value is in the range $[\epsilon, 2 \times \epsilon)$. The maximum value N is given to the solutions that are in the range $[N\epsilon, (N+1)\epsilon) = [1, 1+\epsilon)$, only when they are at the very extreme part of the objective function, where $f = 1$.

So, each archive member a is compared with c using the procedure illustrated in Figure 6 and described next:

1. If the identification array Box_a of any archive member a dominates that of the offspring c, then it means that the offspring is ϵ-dominated

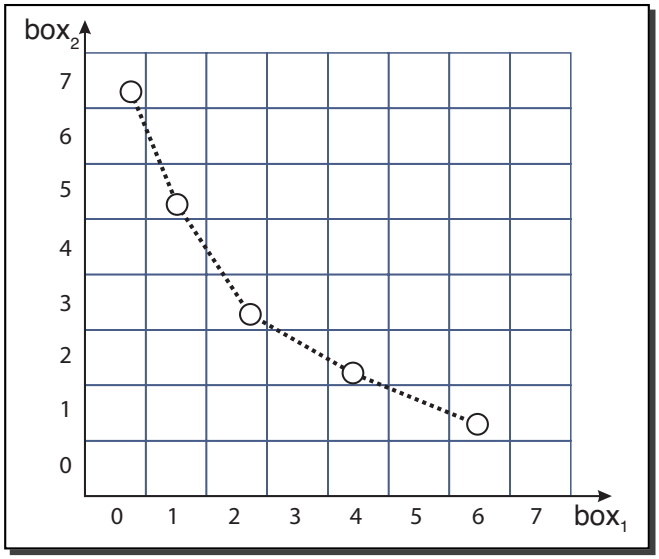


Figure 5: Representation of the objective space with box identification.

by this archive member and thus, the offspring *is not accepted*. This is case (a) in Figure 6.

2. If Box_c of the offspring dominates the Box_a of any archive member a, the archive member is deleted and the offspring *is accepted*. This is case (b) in Figure 6.

   If none of the above cases occur, then it means that the offspring is not ϵ-nondominated with respect to the archive contents. There are two further possibilities in this case:

   (a) If the offspring shares the same box vector with an archive member (meaning that they belong to the same hyper-box), then they are first checked for the usual nondomination. If the offspring dominates the archive member or the offspring is nondominated with respect to the archive member but is closer to the corner of the box vector (in terms of the Euclidian distance) than the archive member, then the offspring *is retained*. This is case (c) in Figure 6.

   (b) In the event of an offspring not sharing the same box vector with any archive member, the offspring *is accepted*. This is case (d) in

Figure 6.

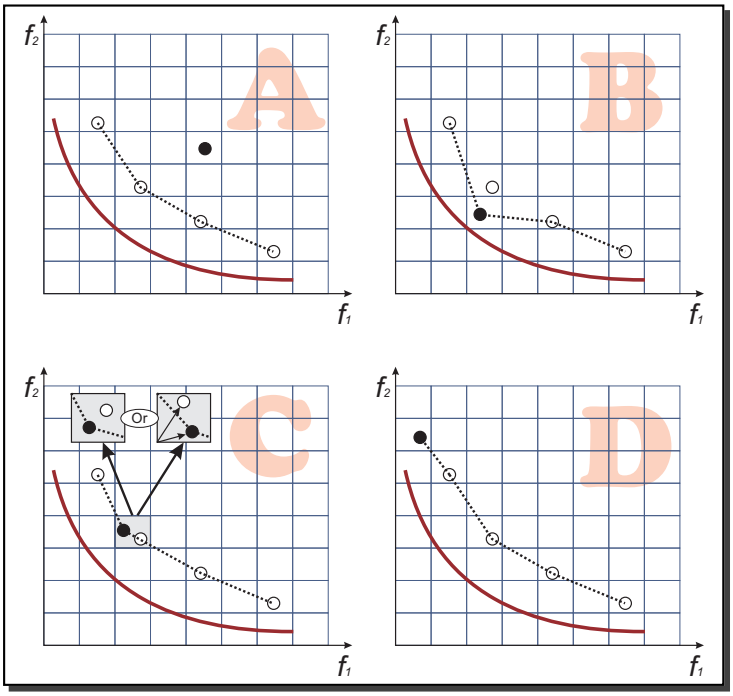


Figure 6: Four cases of accepting an offspring into the external archive

In our proposal, we allow the above selection mechanism to admit one nondominated solution in all the boxes crossed by the Pareto front, as described in the next section.

4 Description of our Proposed Approach

In order to address the problems presented by the two previous approaches based on ϵ-dominance, we propose a new scheme that involves adding an appropriate extra objective to every nondominated solution only when we want to add it to the set of nondominated solutions (or external population) and then apply the conventional Pareto dominance with this extra objective. In order to include a nondominated solution into this external archive, it is compared with respect to each member already contained in the external archive using ϵ-dominance or paϵ-dominance (see Section 3.4). This new objective has to fulfill the following requirements:

1. Adding an extra objective aims to retain some solutions that are in the extreme parts of the Pareto front. If we have two nondominated solutions that are close to each other at the extreme parts of the Pareto front, both ϵ-dominance and paϵ-dominance normally lose one of them. Thus, the new objective should be defined in such a way that these two solutions are incomparable with respect to Pareto dominance, so that both of them can be stored in the external archive.

2. The behavior of the extra objective has to be strictly related to the position of the other objectives. That is, it has to be independent of the shape and specific features of the Pareto front (e.g., convexity, linearity, etc.).

The extra objective which satisfies the above requests is in Table 1, assuming that the objective functions are normalized ($0 \leq f_i \leq 1$, for all i).

2 objectives	3 objectives
$g_1(\vec{x}) = Box_1$ $g_2(\vec{x}) = Box_2$ $g_3(\vec{x}) = 1 - f_1(\vec{x}) - f_2(\vec{x})$	$g_1(\vec{x}) = Box_1$ $g_2(\vec{x}) = Box_2$ $g_3(\vec{x}) = Box_3$ $g_4(\vec{x}) = 1 - f_1(\vec{x}) - f_2(\vec{x}) - f_3(\vec{x})$
k objectives	
$g_1(\vec{x}) = Box_1$ ... $g_k(\vec{x}) = Box_k$ $g_{k+1}(\vec{x}) = 1 - f_1(\vec{x}) - f_2(\vec{x}) - \cdots - f_k(\vec{x})$	

Table 1: Definition of the extra objective function for 2 and 3 objective functions and its generalization to more than 3 objectives.

So, each nondominated solution is assigned to a new extended identification array, denoted by $Box^+ = (g_1, g_2, \cdots, g_{k+1})$.

Thus, the new selection mechanism is as follows: the identification array Box_c^+ of a new nondominated solution c is compared with the identification array of each archive member a, Box_a^+, and one of the following cases could happen (see Figure 7), being the diagonal line crossing a the contour line of $g_{k+1}(\vec{x}) = 1 - f_1(\vec{x}) - f_2(\vec{x}) - \cdots - f_k(\vec{x})$ at the level of a:

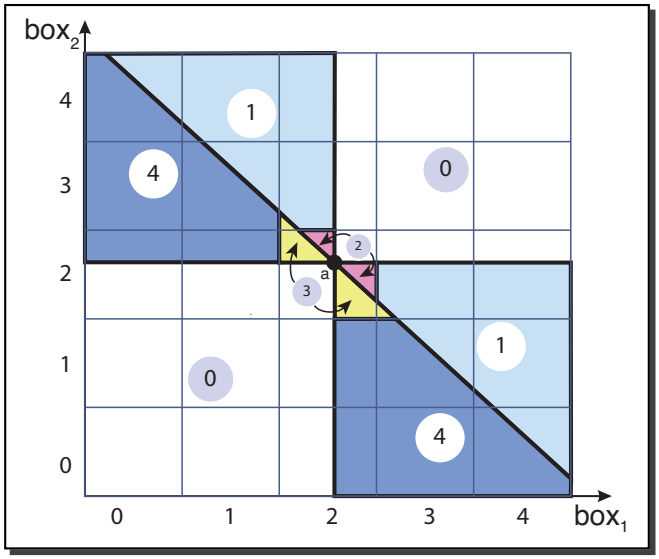


Figure 7: Selection mechanism with the extra objective function.

Region 0: In this case, c is in Region 0, so c and a are incomparable, and they are both marked as incomparable with each other.

Region 1: In this case, c is in Region 1, which means that c is better than a in one of the objectives, but the extra objective of c is lower than a (because c is over the diagonal that crosses a). So, both are marked as incomparable and c will be accepted if the box of c is empty.

Region 2: In this case, c is in Region 2, which means that they share the same box because they have exactly the same identification array values. But c has a lower value than a in the extra objective (because it is over the diagonal that crosses a), so c will replace a as it dominates a using the extra objective.

Region 3: In this case, c is in Region 3, so they are both sharing the same box as they have exactly the same identification array values. But c has a greater value than a in the extra objective (because it is under the diagonal that crosses a), so c will be discarded, as it is dominated by a using the extra objective.

Region 4: In this case, c is in Region 4, which means that c is better than a in one of the objectives, but the extra objective of c is greater than a

(because c is under the diagonal that crosses a). So, both are marked as incomparable and c will be accepted if the box of c is empty.

It is important to mention that when c and a are nondominated with respect to each other, with the proposed approach there is no risk of losing any good solutions and, therefore the convergence properties of the approach remain intact. Moreover, the versatility of this extra objective function is reflected in its behavior when facing Pareto fronts with different geometric characteristics. Next, we describe some of its main features:

Convex Pareto fronts: If the Pareto front is convex and continuous as in Figure 8, $g_3(\vec{x}) \geq 0$ for all $\vec{x}$, and $g_3(\vec{x}) = 0$ only at the extreme points of the Pareto front. Moreover, the third objective increases its value whereas the box vectors are decreasing, and the third objective decreases its value where the box vectors are increasing. But the key factor is that both tendencies change at the same point, which is when the maximum point in g_3 intersects with the Pareto front at the point having a slope of -1. In order to show this, let's assume that the Pareto front and the extra objective function can both be formulated in objective function space as

$$f_2 = F(f_1) \qquad \text{and} \qquad g_3(f_1, f_2) = 1 - f_1 - f_2,$$

being F a proper continuous function. Thus,

$$\frac{\delta F(f_1)}{\delta f_1} = \frac{\delta f_2}{\delta f_1} = -1.$$

On the other hand, the monotonicity of the extra objective, g_3, changes exactly when

$$\frac{\delta g_3(f_1, f_2)}{\delta f_1} = 0,$$

or, equivalently, when

$$\frac{\delta g_3(f_1, f_2)}{\delta f_1} = -1 - \frac{\delta f_2}{\delta f_1} = 0 \Leftrightarrow \frac{\delta f_2}{\delta f_1} = -1.$$

This shows that all the boxes crossed by the Pareto front are nondominated to each other. In case of a disconnected Pareto front, this proof can be locally reproduced for each of the continuous parts of the front with an appropriate function F.

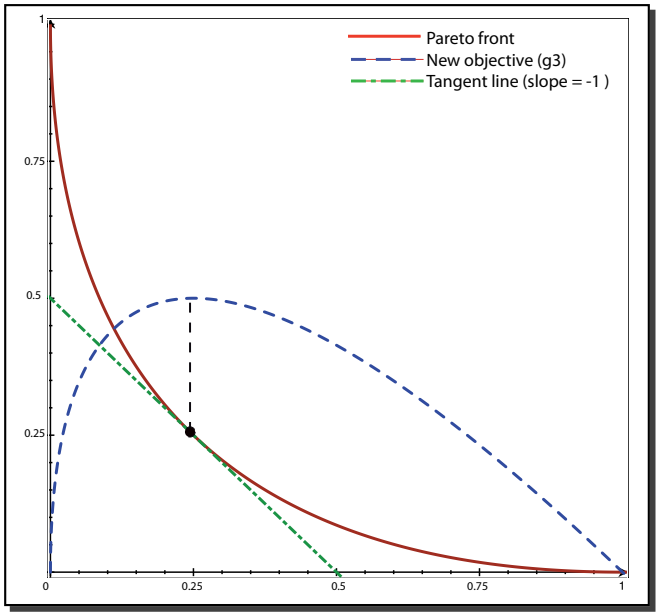


Figure 8: Convex Pareto front and the plot of the third objective with respect to the first and second objectives.

Concave Pareto fronts: If the Pareto front is concave as in Figure 9, $g_3(\vec{x}) \leq 0$ for all $\vec{x}$, and $g_3(\vec{x}) = 0$ only at the extreme points of the Pareto front. Now, the third objective decreases its value when the box indices are increasing, and viceversa. Again, it can be easily seen that the monotonicity of the third objective changes exactly at the same point in the Pareto front at which the slope is equal to -1.

Linear Pareto fronts: If the Pareto front is linear, both ϵ-dominance and paϵ-dominance generate identical uniform grids (see [11]) and $g_3(\vec{x}) = 0$ for all $\vec{x}$. So the third objective does not have any effect.

There is one additional detail that is important to clarify regarding our proposed approach. If we have, for example, a bi-objective optimization problem, both the classical ϵ-dominance and paϵ-dominance retain a maximum of N nondominated solutions in an $N \times N$ grid. But now, since one solution is always stored in each box, that "maximum" capacity can be exceeded. So, as it was commented in the introduction, three different schemes are proposed here depending on the user's preferences:

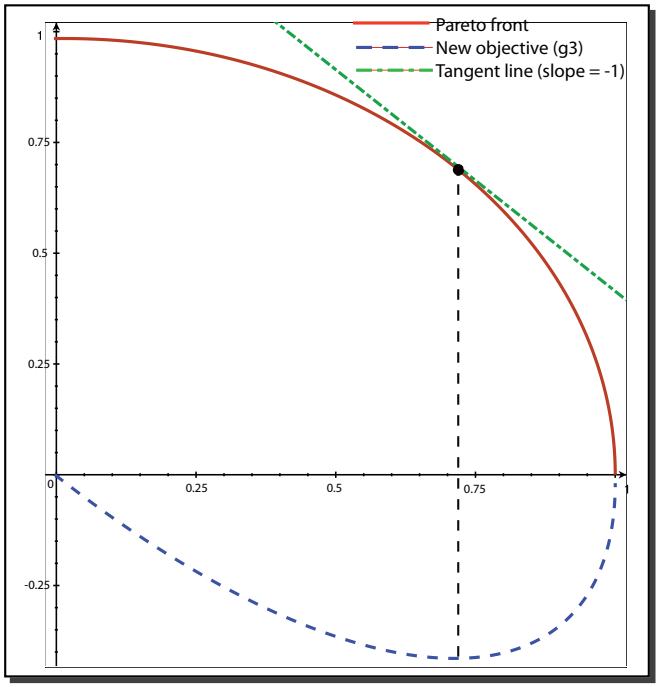


Figure 9: Concave Pareto front and the plot of the third objective with respect to the first and second objectives.

Scheme 1: When the user provides the number of desired nondominated solutions but such limit can be exceeded. In this case, the most accurate grid is ϵ-dominance improved with the new proposed objective function, although an $N \times N$-grid (for bi-objective problems) could store up to $2N + 1$ nondominated solutions (only in extreme cases of concave or convex Pareto fronts. See Scheme 3 for more details). This scheme presents several advantages such as the fact that it does not require any information about the geometric characteristics of the Pareto front and that there is no need of adjusting the ϵ value too accurately.

Scheme 2: When the user provides the number of desired nondominated solutions and is able to incorporate information about the geometric characteristics of the Pareto front. In this case, if we are able to incorporate such type of information, the most accurate grid is the paϵ-dominance improved with the new proposed objective function. Although again an $N \times N$-grid (for bi-objective problems) could store up to $2N + 1$ nondominated solutions, paϵ-dominance takes advantage of that infor-

mation and minimizes the deviation from the maximum "traditional" capacity of N solutions (see again Section 3 or [11] for further details), as we will see in our experimental results. This scheme presents more advantages such as the fact that there is no need of calculating the ϵ value.

Scheme 3: When the user provides the number of desired nondominated solutions and does not want to exceed it, but there is no information about the geometric characteristics of the Pareto front.

Here, we propose the use of ϵ-dominance with a new mechanism for computing the ϵ value. In this case, taking into account the minimum and the maximum capacities of the grid, the ϵ value will be adjusted to match the expected number of solutions stored by the grid with the user's value. Next, we explain this in more detail for problems having two and three objectives.

Bi-objective optimization problems. It has been previously discussed that an $N \times N$ ϵ-dominance grid can store between $N+1$ and $2N+1$ points. The reason is that the extreme solutions belong to boxes $(0, N)$ and $(N, 0)$ and, depending on the geometric characteristics of the Pareto front, the compromise solutions should occupy between $N-1$ and $2N-1$ more boxes. $N-1$ is the number of stored solutions when we have a linear Pareto front, including the extremes, i.e., we would have points in all the diagonal, in boxes with values of: $(1, N-1)$, $(2, N-2)$,..., $(N-1, 1)$. Meanwhile, we would retain $2N-1$ solutions when the Pareto front is extremely convex (or concave) and would go through the boxes with values of: $(1, N)$, $(2, N)$,..., $(N-1, N)$, (N, N), $(N, N-1)$,..., $(N, 1)$. Thus, the average capacity of the grid is

$$\text{Average capacity} = \frac{1+N+1+2N}{2} = \frac{2+3N}{2}.$$

Then, the appropriate N^* values (or, equivalently, the best ϵ value computed as $\epsilon^* = \frac{1}{N^*}$) to obtain a desired capacity of $DCap$ solutions are shown in Table 2.

Three-objective optimization problems. Now, for a $N \times N \times N$ grid, the three extreme solutions belong to boxes $(N, 0, 0)$, $(0, N, 0)$ and $(0, 0, N)$ and it can be seen that the Pareto front may cross through a minimum

$DCap$	N^*	ϵ^*
10	6	0.16
25	16	0.0625
50	32.6	0.0306
100	**66**	**0.015**
150	99.3	0.01
200	132.6	0.00753

Table 2: Appropriate ϵ value for bi-objective functions according to the number of nondominated solutions desired.

of

$$(N+1)+N+(N-1)+\cdots+1=\frac{N^2+3N+2}{2}$$

boxes and a maximum number of boxes of $1+3N^2$. So,

$$\text{Average capacity}=\frac{\frac{N^2+3N+2}{2}+1+3N^2}{2}=\frac{7N^2+3N+4}{4}.$$

Then, the appropriate N^* values (or, equivalently, the best ϵ values computed as $\epsilon^*=\frac{1}{N^*}$) to obtain a desired capacity of $DCap$ solutions are shown in Table 3.

$DCap$	N^*	ϵ^*
10	2.06	0.4854
25	3.4951	0.2861
50	5.08	0.1968
100	**7.31**	**0.1367**
150	9.015	0.1109
200	10.45	0.0957

Table 3: Appropriate ϵ value for three-objective functions according to the number of nondominated solutions desired.

5 Results

In this section we describe the results of the experiments that we conducted to validate our proposed approach. It is important to mention that the efficiency of ϵ-dominance depends only on the geometric properties of the Pareto front.

In the specialized literature, there are well-known test suites to validate MOEAs (see for example [28, 8, 12]). These test problems, however, are designed to test certain search capabilities of MOEAs, and not aspects related to archiving techniques (i.e., many of these test problems have Pareto fronts with the same geometrical shape). This is the reason why we selected a set of test functions that are challenging in terms of the geometrical shapes of their Pareto fronts, since that is what we aim to assess in our case. The test problems adopted are the following five bi-objective problems (three taken from the Zitzler-Deb-Thiele (**ZDT**) set [28], two more from [4]), five problems with three objectives (one taken from the Deb-Thiele-Laumanns-Zitzler (**DTLZ**) set [8], two more from the the Walking-Fish-Group (WFG) set [12], and the last two are new proposals designated specifically to measure the capability of an algorithm to deal with convex problems). The definitions and characteristics of these test problems are provided in Table 13. It is important to mention again that these test problems were chosen because they have different geometric characteristics, so that we can assess the performance of our proposed approach when dealing with convex, concave, connected and disconnected Pareto fronts. For each test problem, we generated an approximation of the real Pareto front with thousands of solutions and kept them in different files. Then, we provided the solutions in a deterministic way to each of the different mechanisms used for the comparison, so they all had the same set of points for their filtering process. The aim of all the approaches under comparison was then to reduce the original (large) file to a much smaller set of solutions maintaining an appropriate distribution along the Pareto front.

In order to allow a fair comparison of the proposed approach with the others, each scheme includes the following experiments:

Scheme 1: Filtered true Pareto fronts obtained with ϵ-dominance are compared with their corresponding filtered Pareto fronts obtained with the improved ϵ-dominance (we call it implicit ϵ-dominance) for a desired capacity of $N = 100$ solutions.

Scheme 2: Filtered true Pareto fronts obtained with paϵ-dominance are

compared with their corresponding filtered Pareto fronts obtained with the improved paϵ-dominance (we call it implicit paϵ-dominance) for a desired capacity of $N = 100$ solutions.

Scheme 3: Filtered true Pareto fronts obtained with the improved ϵ-dominance, using the appropriate ϵ values included in Tables 2 and 3 for a desired capacity of 100 solutions, are compared with their corresponding filtered Pareto fronts obtained with Scheme 1.

Usually, in order to allow a quantitative comparison of results among the different algorithms, there are two distinct goals that we pursue: (1) the solutions should be as close to the Pareto optimal solutions as possible (i.e., closest to the true Pareto front) and (2) the solutions should be as diverse as possible along the Pareto front (i.e., to have a good distribution of solutions along the Pareto front). Apparently, these two goals are independent from each other and there exist different performance measures to deal with each one or both of these goals. Thus, it does not exist a single performance measure that can indicate the superiority of one algorithm over the other in these two aspects. So, in general, there is a clear need of having at least two performance measures for adequately evaluate both goals (convergence and diversity) of a MOEA. Nevertheless, due to the fact that we are filtering the true Pareto fronts, we are only interested in measuring the distribution of solutions (evidently, assessing convergence in this case, makes absolutely no sense). That is why we propose the use of the four following measures specifically designed for assessing diversity:

Number of points (#): $\#(A)$ shows us how far the number of solutions in A is from the desired capacity of the grid. This measure is more relevant for Schemes 2 and 3, where both have been designed in such a way that the maximum capacity of the grid can be exceeded. So, in all our experiments, the grid was defined with an *a priori* desired capacity of 100 points. So, the closer to 100 that an algorithm gets, the better the value of this performance measure. When the user prefers Scheme 1 and it does not care about exceeding it, this measure is not taken into account, although it is, nevertheless, computed.

Spread (Δ): Deb [5] proposed Δ with the idea of measuring both progress towards the Pareto optimal front and the extent of the spread. To this

end, if A is a Pareto front approximation, Δ is defined as follows:

$$\Delta = \frac{\sum_{i=1}^{k} d_i^e + \sum_{i=1}^{\#(PF_{true})} |d_i - \overline{d}|}{\sum_{i=1}^{k} d_i^e + \#(PF_{true})\overline{d}}.$$

where d_i^e denotes the distance between the i-th coordinate for both extreme points in A and the true Pareto front PF_{true}. d_i measures the distance of each point in A to its closer point in PF_{true} meanwhile $\overline{d}$ represents their mean value.

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.

Spacing (Spacing): This measure was proposed by Schott [24] and calculates the distances from each point to its closest neighboor in the approximated Pareto front, A. It can be formally defined as:

$$Spacing = \sqrt{\frac{1}{\#(A)} \sum_{i=1}^{\#(A)} (d_i - \bar{d})^2}$$

where $d_i = min_{j \in A} \sum_{m=1}^{k} |f_m^i - f_m^j|$ and $\bar{d}$ is the mean value of the distance d_i, this is, the minimum value of the sum of the absolute difference in objective function values between the $i - th$ solution and any other solution in the Pareto optimal set. A value of 0 means that all the solutions are equally distributed along the Pareto front.

Standard deviation of the crowding distances (SDC): This performance measure tries to get more information with the crowding distance (see Section 3.2) through the use of the standard deviation from a Pareto set as:

$$SDC = \sqrt{\frac{1}{\#(A)} \sum_{i=1}^{\#(A)} (d_i - \overline{d})^2}$$

where d_i is the crowding distance for the $i-th$ point in A and $\bar{d}$ is the mean value of the distance d_i. Now, $0 \leq SDC \leq \infty$ and the lower the value of SDC, the better the distribution of vectors in A. A perfect distribution, that is $SDC = 0$, means that d_i is constant for all i.

5.1 Discussion of Results

Tables 4, 5 and 6 show a summary of our results for all the bi-objective problems and Tables 7, 8 and 9 show the results for the problems with three objectives, for the three schemes considered, respectively. For each test problem, we report the values obtained for each algorithm with respect to each performance measure. The algorithms only do the reduction of solutions once and the process is deterministic for that set of solutions. The best values in each case are shown in **boldface**.

The graphical results are shown in Figures 10, 11, 12, 13, and 14 for the problems with two objectives and in Figures 15, 16, 17, 18 and 19 for the problems with three objectives. These plots correspond to the unique and single result provided by each algorithm. In all the bi-objective optimization problems, the true Pareto front is shown with a continuous line and the approximation obtained by each algorithm is shown with circles.

5.1.1 Scheme 1: ϵ-dominance vs implicit ϵ-dominance

The main purpose of this experiment is to reduce the number of solutions using both ϵ-dominance and the improved mechanism (implicit ϵ-dominance).

For bi-objetive problems, we fixed the number of divisions per objective to 100 and, once the objective functions are normalized, $\epsilon = 1/100$. So, we expect to have 100 nondominated solutions as a result in each front. In Table 4 we can see the performance measures comparison and the graphical results are shown in Figures 10-a, 10-b, 11-a, 11-b, 12-a, 12-b, 13-a, 13-b, 14-a, and 14-b. It can be clearly seen in Table 4 that the performance measures show that implicit ϵ-dominance produced the best values in all cases for the bi-objective problems. We obtained more solutions in the Pareto front than using the original ϵ-dominance, obtaining the best performance in all the metrics adopted. Graphically, we can see in three problems (Deb24, Deb52 and ZDT3), that the ϵ-dominance method is not able to retain the solutions in the extreme parts of the Pareto fronts, and that our implicit ϵ-dominance can retain the solutions in those extreme parts of the Pareto front without

degrading the performance and the distribution quality of the nondominated solutions.

For three-objetive optimization problems, we set the number of divisions per objective to 10, ideally expecting to get 100 nondominated solutions in the problems with three objectives. We can see in Table 7 the performance measures comparison and the graphical results are shown in Figures 15-a, 15-b, 16-a, 16-b, 17-a, 17-b, 18-a, 18-b, 19-a, and 19-b. It can be seen in Table 7 that the performance measures show that the new implicit ϵ-dominance produced the best results in all cases. It obtained more solutions in the final front, and the distribution of the solutions was better in all cases. Graphically, we can see that the ϵ-dominance mechanism retains just a few solutions from all the Pareto front. More specifically, it cannot retain good solutions in the extreme parts of the Pareto fronts. In contrast, our implicit ϵ-dominance is able to retain more solutions along the Pareto front in all cases.

5.1.2 Scheme 2: paϵ-dominance vs implicit paϵ-dominance

In this experiment, we tried to control the number of solutions using the paϵ-dominance and we added the improved paϵ-dominance to the comparison.

For bi-objective problems, we fixed the number of nondominated solutions to 100 for each problem. In Table 5 we can see the performance measures comparison and the graphical results are shown in Figures 10-c, 10-d, 11-c, 11-d, 12-c, 12-d, 13-c, 13-d, 14-c, and 14-d. It can be clearly seen in Table 5 that the new method, implicit paϵ-dominance, outperformed in all cases to the original paϵ-dominance method with respect to all the performance measures. It obtained more solutions in the final front, and the distribution of the solutions was better in all cases. Graphically, we can see indeed that paϵ-dominance obtained, in general, better results than the original ϵ-dominance, but still has some problems to retain the solutions in the extreme parts of the Pareto front, especially in problem Deb52, in which the left part of the Pareto front is almost completely lost. But, for the new implicit ϵ-dominance, those parts of the Pareto front are filled up with nondominated solutions. In fact, implicit ϵ-dominance is able to retain the solutions in all cases, for convex and nonconvex Pareto fronts.

For three-objetive optimization problems, we also fixed the number of nondominated solutions to 100 for each problem. In Table 8 we can see the performance measures comparison and the graphical results are shown in Figures 15-c, 15-d, 16-c, 16-d, 17-c, 17-d, 18-c, 18-d, 19-c, and 19-d. Al-

though the results shown for the paϵ-dominance mechanism in all the test problems are better than the ϵ-dominance, the performance of the implicit paϵ-dominance is even better than paϵ-dominance. With respect to the performance measures shown in Table 8, we see that the implicit paϵ-dominance gets the best results in most cases. However, paϵ-dominance has the best performance in 3 test functions with respect to Spread and is also the best in 2 other functions with respect to SDC. Graphically, we can notice that the distribution of the points obtained by implicit paϵ-dominance is better because its distribution is very uniform in all cases regardless of the shape of the Pareto front (convex or nonconvex), and it is also able to retain more solutions in the extreme parts of the Pareto fronts.

5.1.3 Scheme 3: Implicit ϵ-dominance adjusting the desired capacity

For bi-objective problems, we fixed the epsilon value according to Table 2 trying to obtain an average number of nondominated solutions of 100. The performance measure results are shown in Table 6 and the graphical results are shown in Figures 10-e, 11-e, 12-e, 13-e, and 14-e. From the performance measures shown in Table 6, we can see that the average number of nondominated solutions obtained by the method is much closer to 100 (with a mean of 95.6) than the first experiment with 100 divisions (with a mean of 143.3, see Table 4). With respect to Spread, Spacing and SDC, the results are very similar to those obtained in the first experiment in spite of the fact that a smaller number of divisions was adopted in this case. Graphically, we can confirm that the use of less divisions per objective does not affect the performance of our proposed implicit ϵ-dominance. In all the test problems we were able to maintain a good performance for the new mechanism when retaining less nondominated solutions.

Finally, for the three-objetive optimization problems, we first tried to fix the number of divisions to get 100 nondominated solutions, but the results in Table 7 show that the improved ϵ-dominance method gets more solutions than we originally wanted in 3 problems. So, for the third experiment we fixed the epsilon value according to Table 3. The performance measure results are shown in Table 9 and the graphical results are shown in Figures 15-e, 16-e, 17-e, 18-e, and 19-e. From Table 9, we can see that the maximum number of nondominated solutions never exceeds 100 and that the values of all the performance measures are slightly poorer to those obtained in Table 7

with more divisions per objective. Regarding the graphical results, we can see that the Pareto fronts obtained for the different problems are uniformly distributed along the Pareto front, and that the proposed approach was able to retain the solutions in all the extreme parts of the Pareto front.

The above results have shown the effectiveness of the relatively simple filtering scheme introduced in this paper. Note however, that if we want to incorporate such an approach into any MOEA, it is important to take into account that the proposed dominance relation has to be used only for nondominated solutions.

In order to couple this approach to any MOEA, the procedure is the following. Once the offspring c is generated:

1. First, it is required to check the Pareto dominance relation with respect to all the nondominated solutions included in the external archive.

2. Once c is classified as a nondominated solution, c has to be sent to the external archive using the proposed dominance relation with the extra objective function.

With regard to the updating of the grid in those situations in which the new nondominated solutions lie outside of the actual dimensions of the grid, our previous experience using ϵ-dominance and paϵ-dominance suggests to avoid as much as possible the use of this option. To this end, two of the most common solutions are:

1. Activate the first grid to filter out the set of nondominated solutions once the size of this set is big enough. In [11], the authors recommend a size of 150 solutions before the initialization of the paϵ-dominance grid.

2. Another solution is the one used by PAES in [15] where the authors propose the use of extra areas in those extreme boxes (see Figure 2).

Finally, in the case in which the objective functions need to be normalized before the extra objective is calculated, we use the next equation to normalize the objectives:

$$f_{i,new} = \frac{f_{i,current} - f_{i,min}}{f_{i,max} - f_{i,min}} \in [0, 1]$$

for each $i = 1, 2, \ldots, k$, where $f_{i,new}, f_{i,current}, f_{i,min}, f_{i,max}$ are the normalized value, the actual value, the minimum and the maximum value, respectively, in the current population, for objective i.

5.2 Application to Real World Problems

Finally, we tested our proposal in two real problems frequently used as benchmarks to validate new evolutionary optimization algorithms: the design of a gear box [13], also known as the Speed Reducer problem, and the design of a welded beam structure [21]. A brief description and the mathematical formulation of both problems are shown in Table 14. Due to the difficulty of these problems, the true Pareto fronts adopted for the filtering process only contained 144 and 117 nondominated solutions, respectively. That is the reason why a grid with a maximum capacity of 50 points was used in this case, instead of 100. Hence, in Tables 10, 11 and 12 we show a summary of the results for the same three schemes that we considered in the previous section. The graphical results are shown in Figures 20 and 21, for the Speed Reducer and the Welded Beam, respectively. It can be clearly seen that the performance measures show that implicit ϵ-dominance and the implicit paϵ-dominance produced the best values in almost all cases. Again, we retained more solutions in the Pareto front than using the original ϵ-dominance or paϵ-dominance, obtaining the best performance with respect to all the metrics adopted.

6 Conclusions and Future Work

In this paper, we have proposed a new scheme to deal with the problem of how to properly distribute nondominated solutions along the Pareto front when using an external archive. Our core idea is to use the ϵ-dominance approach and add an extra objective, which allows us to retain solutions that are normally lost when using the original ϵ-dominance approach. This extra objective has the value: $g_3 = 1 - f_1 - f_2$ for two objectives or $g_4 = 1 - f_1 - f_2 - f_3$ for problems with three objectives. We decided to use another approach that is based on ϵ-dominance called paϵ-dominance which is capable of dynamically adjusting to the geometric characteristics of the Pareto optimal front and that is able to retain more nondominated solutions than the original ϵ-dominance method.

In order to assess the performance of our proposed implicit ϵ-dominance, we solved ten test problems with two and three objectives, and having different geometric characteristics. We also adopted three metrics designed to measure diversity properties and one more measure related to the num-

ber of points found. In all cases, the new mechanism was able to help the existing ones (ϵ-dominance and paϵ-dominance) to retain a higher and well-distributed number of nondominated solutions.

We conducted three experiments: 1) a direct comparison of the new mechanism to help ϵ-dominance to get better results, 2) another comparison with respect to paϵ-dominance to show that our proposed approach could also help in this case and 3) an adjustment of the value of the ϵ-vector used by ϵ-dominance in order to retain an expected value of 100 nondominated solutions in the final Pareto front approximation obtained.

With the new mechanism, we were able to maintain the good convergence properties of the original ϵ-dominance, without requiring any prior information about the actual geometric characteristics of the Pareto front. Our proposed approach was tested in convex and nonconvex problems with two and three objectives, and in all cases it showed a significant improvement regarding the distribution of nondominated solutions, being able to reach regions that the other approaches could not.

The main drawback of the new approach is that it does not give us a well-defined control mechanism that allows us to obtain an exact (pre-defined) number of nondominated solutions. However, with the use of paϵ-dominance, it is possible to have a better control of such solutions and avoid obtaining as many solutions as when using the original ϵ-dominance mechanism. However, it remains as part of our future work to provide a better control mechanism for our proposed approach, such that it can be self-contained when used with any MOEA. Moreover, we are interested in testing the proposed approach coupled to a MOEA and compare its performance with respect to other MOEAs, including those that do not adopt nondominated sorting (e.g., MOEA/D [27] or fast sorting [20]). These alternative approaches have a lower algorithmic complexity than NSGA-II. The computational complexity of each generation in NSGA-II is $O(MN^2)$, where M is the number of objectives and N is the population size. In contrast, the computational complexity of MOEA/D is $O(MNT)$, where T is the result of the descomposition of the multi-objective optimization problem being solve, and is a lower value than N, which makes this approach faster than NSGA-II. Also, the fast sorting mechanism reported in [20] refers to a new rank-sum sorting method to divide every objective into ranks. This has linear complexity $O(N)$, and, therefore, also reduces the complexity of the original nondominated sorting method adopted by NSGA-II.

Finally, it would also be interesting to study the effect of the proposed

	implicit ϵ-dominance				ϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC	#(P)	Spread	Spacing	SDC
ZDT1	**126**	**0.0793**	**0.0026**	**0.0050**	75	0.2614	0.0061	0.0101
Deb24	**148**	**0.0815**	**0.0027**	**0.0054**	52	0.4559	0.0110	0.0181
ZDT2	**126**	**0.0722**	**0.0023**	**0.0051**	77	0.3445	0.0125	0.0181
Deb52	**133**	**0.4965**	**0.0022**	**0.0046**	29	0.8398	0.0793	0.1332
ZDT3	**184**	**0.4723**	**0.0021**	**0.0357**	28	0.6041	0.0383	0.0949
Mean	**143.3**	**0.2403**	**0.0024**	**0.0112**	52.2	0.501	0.029	0.055

Table 4: Performance measure values for the bi-objective problems using Scheme 1 for a grid size of 100×100.

	implicit paϵ-dominance				paϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC	#(P)	Spread	Spacing	SDC
ZDT1	**107**	**0.0731**	**0.0018**	**0.0042**	90	0.1419	0.0023	0.0056
Deb24	**108**	**0.2141**	**0.0043**	**0.0077**	81	0.4022	0.0247	0.0346
ZDT2	**105**	**0.0930**	**0.0025**	**0.0051**	90	0.2563	0.0074	0.0120
Deb52	**140**	**0.5865**	**0.0024**	**0.0050**	61	0.7974	0.0480	0.0833
ZDT3	**108**	**0.4413**	**0.0043**	**0.0447**	31	0.5201	0.0313	0.0897
Mean	**113.6**	**0.282**	**0.003**	**0.013**	70.6	0.424	0.023	0.045

Table 5: Performance measure values for the bi-objective problems using Scheme 2, that is, the number of points stored is controlled by means of paϵ-dominance. The size of the grid is 100×100.

archiving technique when used with search engines that have been specifically designed to exploit the properties of ϵ-dominance, such as ϵ-MOEA [6].

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments which greatly helped them to improve the contents of this paper.

The third author acknowledges support from CONACyT project no. 103570.

	implicit ϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC
ZDT1	83	0.0774	0.0035	0.0087
Deb24	98	0.0726	0.0038	0.0083
ZDT2	84	0.0738	0.0036	0.0079
Deb52	88	0.1861	0.0037	0.0071
ZDT3	125	0.4376	0.0037	0.0406
Mean	95.6	0.169	0.004	0.015

Table 6: Performance measure values for the bi-objective problems using Scheme 3 with a grid size of 66×66 or, equivalently, for an expected capacity of 100.

	implicit ϵ-dominance				ϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC	#(P)	Spread	Spacing	SDC
DTLZ2	**162**	**0.3160**	**0.0378**	**0.0592**	38	0.3572	0.0789	0.1032
WFG4	**142**	**0.3148**	**0.1211**	**0.0595**	43	0.3998	0.4429	0.0762
Convex50	**50**	**0.4410**	**0.0409**	**0.0452**	16	0.7414	0.2029	0.2422
Convex60	**60**	**0.4329**	**0.0437**	**0.0386**	21	0.5013	0.1009	0.1564
WFG2	**107**	**0.3586**	**0.1230**	**0.0523**	21	0.4074	0.3308	0.1054
Mean	**104.2**	**0.3727**	**0.0733**	**0.0509**	27.8	0.4814	0.2313	0.1367

Table 7: Performance measure values for the three-objective problems using Scheme 1 for a grid size of $10 \times 10 \times 10$.

	implicit paϵ-dominance				paϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC	#(P)	Spread	Spacing	SDC
DTLZ2	**71**	**0.1260**	**0.0107**	0.0975	59	0.3682	0.0691	**0.0709**
WFG4	**70**	**0.2380**	**0.1503**	0.0844	57	0.3288	0.3031	**0.0841**
Convex50	**85**	0.5549	**0.0432**	**0.0798**	47	**0.5442**	0.0900	0.1255
Convex60	**79**	0.4914	**0.0428**	**0.0736**	54	**0.4394**	0.0729	0.1006
WFG2	**95**	0.4533	**0.1667**	**0.0590**	34	**0.4504**	0.2978	0.1063
Mean	**80.0**	**0.3727**	**0.0828**	**0.0789**	50.2	0.4262	0.1667	0.0975

Table 8: Performance measure values for the three-objective problems using Scheme 2, that is, the number of points stored is controlled by means of paϵ-dominance. The size of the grid is $10 \times 10 \times 10$.

	implicit ϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC
DTLZ2	94	0.4079	0.0524	0.0752
WFG2	80	0.3787	0.1763	0.0815
WFG4	33	0.5015	0.0655	0.0726
Convex50	37	0.4176	0.0648	0.0637
Convex60	71	0.5392	0.2264	0.0588
Mean	63.0	0.4490	0.1171	0.0704

Table 9: Performance measure values for the three-objective problems using Scheme 3 with a grid size of $7.31 \times 7.31 \times 7.31$ for an expected capacity of 100.

	implicit ϵ-dominance				ϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC	#(P)	Spread	Spacing	SDC
S.R.	**67**	**0.4275**	**0.0177**	**0.0306**	5	0.5244	0.3116	0.3079
W.B.	**71**	**0.2919**	**0.01035**	**0.0176**	19	0.4755	0.05065	0.0898
Mean	**69**	**0.3598**	**0.0141**	**0.0241**	12	0.5000	0.1812	0.1989

Table 10: Performance measure values for the real world problems using Scheme 1 for a grid size of 50×50.

	implicit paϵ-dominance				paϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC	#(P)	Spread	Spacing	SDC
S.R.	**48**	**0.4534**	**0.0220**	**0.0439**	22	0.5361	0.0697	0.1167
W.B.	**54**	0.3360	0.0127	**0.0210**	42	**0.2627**	**0.0108**	0.02293
Mean	**51**	**0.3948**	**0.0174**	**0.0325**	32.0	0.3994	0.0403	0.0699

Table 11: Performance measure values for the real world problems using Scheme 2, that is, the number of points stored is controlled by means of paϵ-dominance. The size of the grid is of 50×50.

	implicit ϵ-dominance			
Fun	#(P)	Spread	Spacing	SDC
S.R.	52	0.2379	0.0138	0.0243
W.B.	51	0.7515	0.2677	0.3075
Mean	51.5	0.495	0.141	0.166

Table 12: Performance measure values for the real world problems using Scheme 3 with a grid size of 32.6×32.6 or, equivalently, for an expected capacity of 50 (see Table 2).

References

[1] C. A. Coello Coello. A Short Tutorial on Evolutionary Multiobjective Optimization. In E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 21–40. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

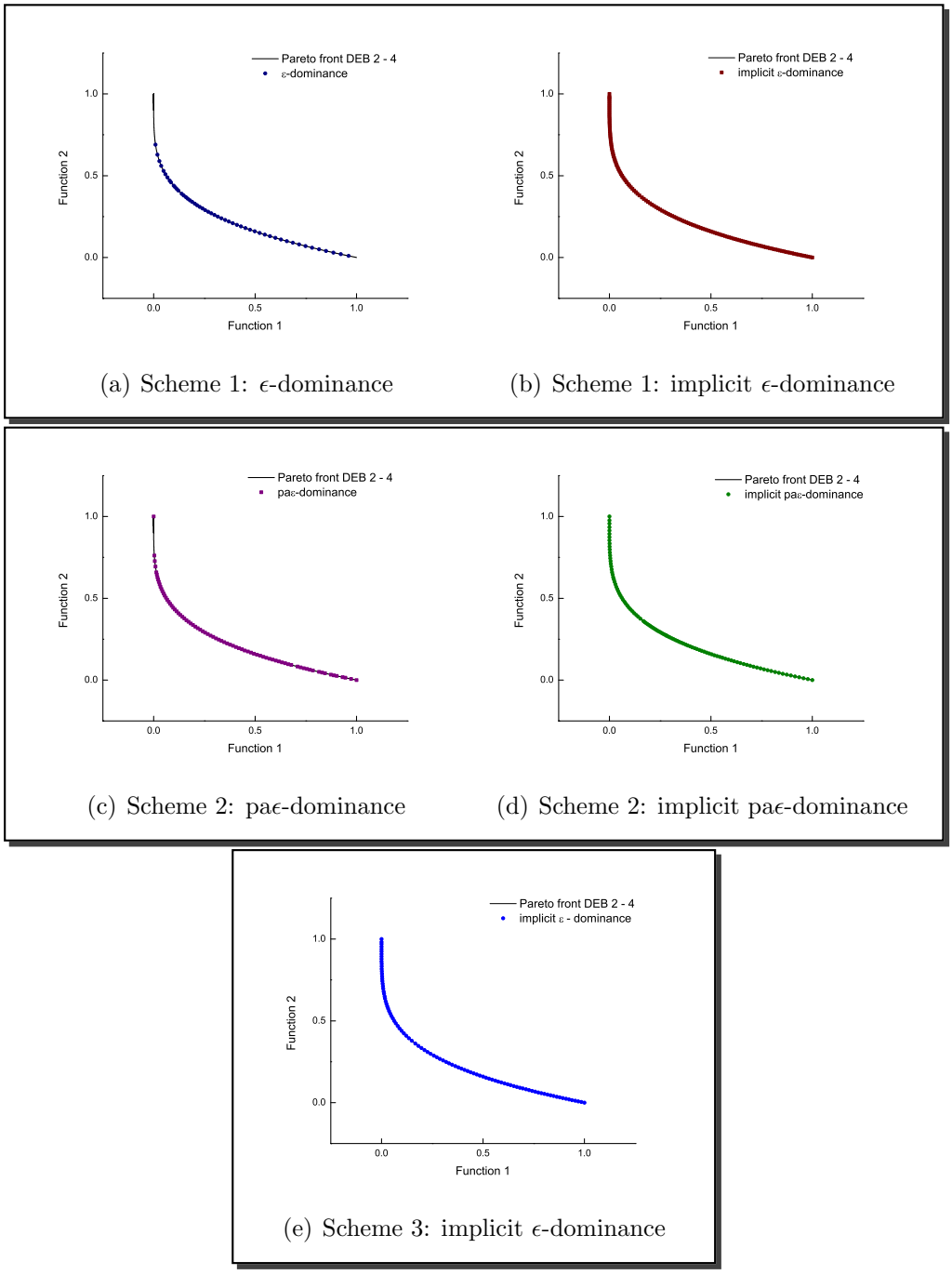
[2] C. A. Coello Coello, G. Toscano Pulido, and M. Salazar Lechuga. Handling Multiple Objectives With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004.

[3] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 283–290, San Francisco, California, 2001. Morgan Kaufmann Publishers.

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

[5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[6] K. Deb, M. Mohan, and S. Mishra. Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of



(a) Scheme 1: ϵ-dominance

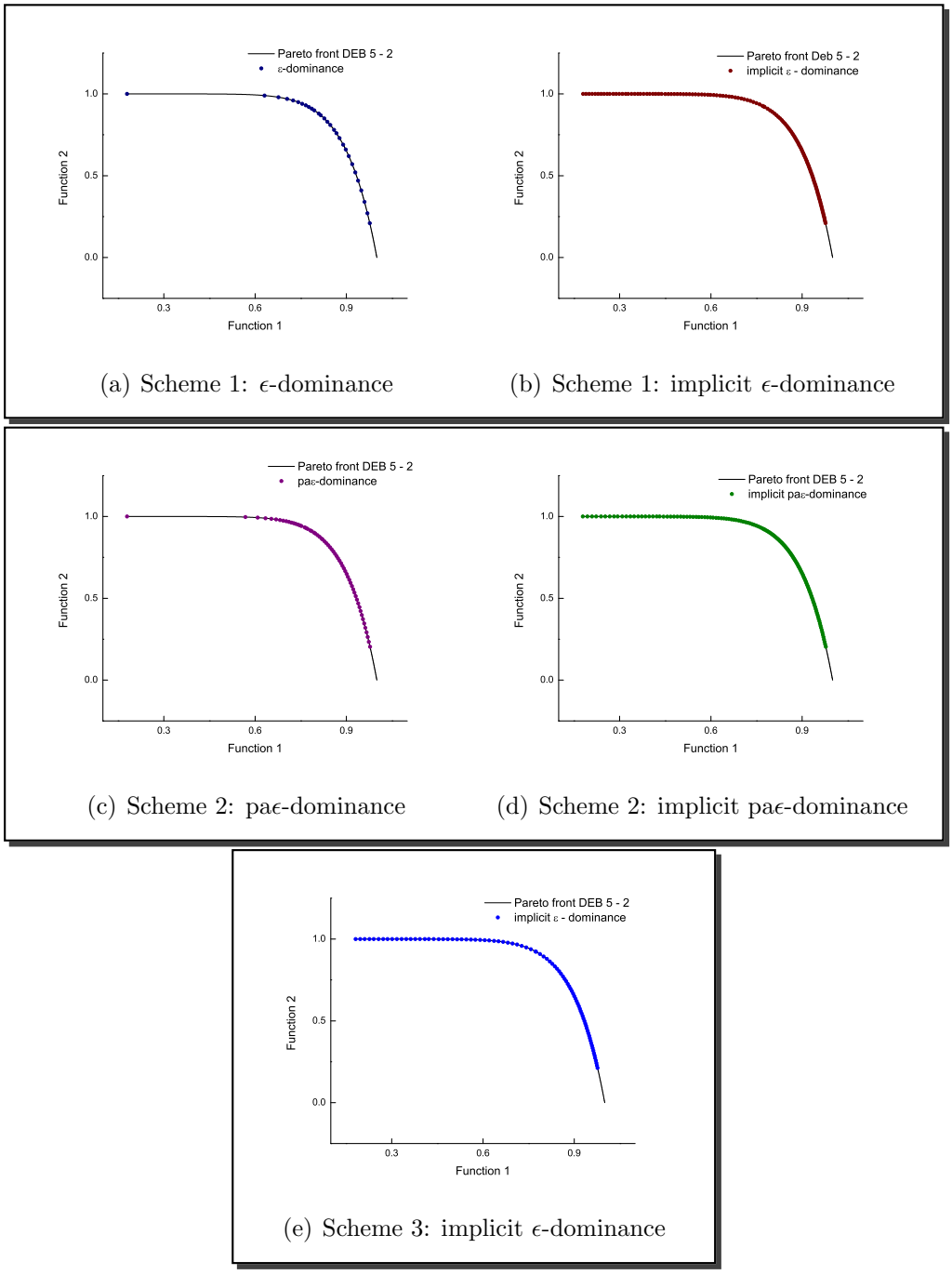
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 10: Pareto fronts obtained for Deb24.



(a) Scheme 1: ϵ-dominance

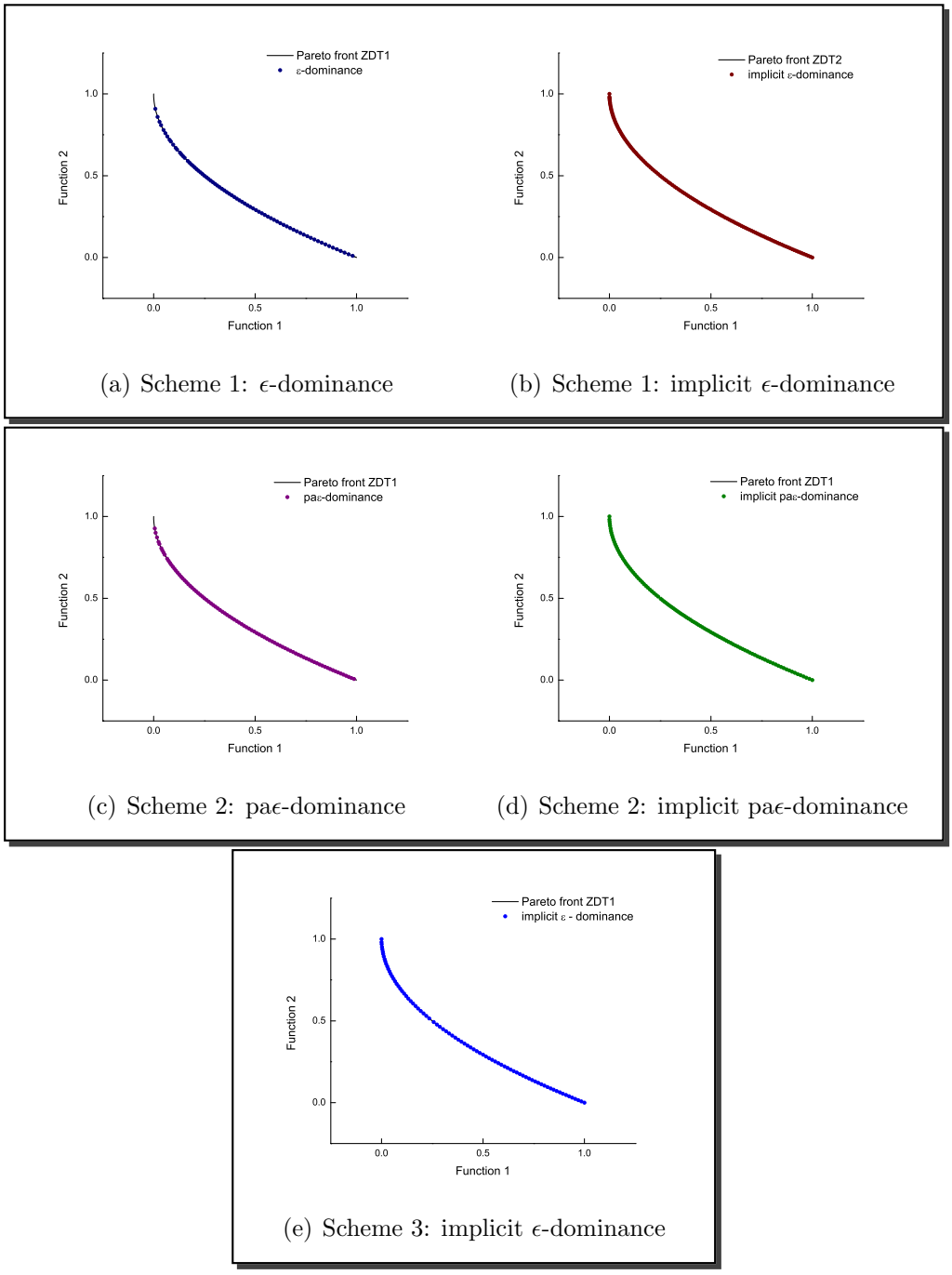
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 11: Pareto fronts obtained for Deb52.



(a) Scheme 1: ϵ-dominance

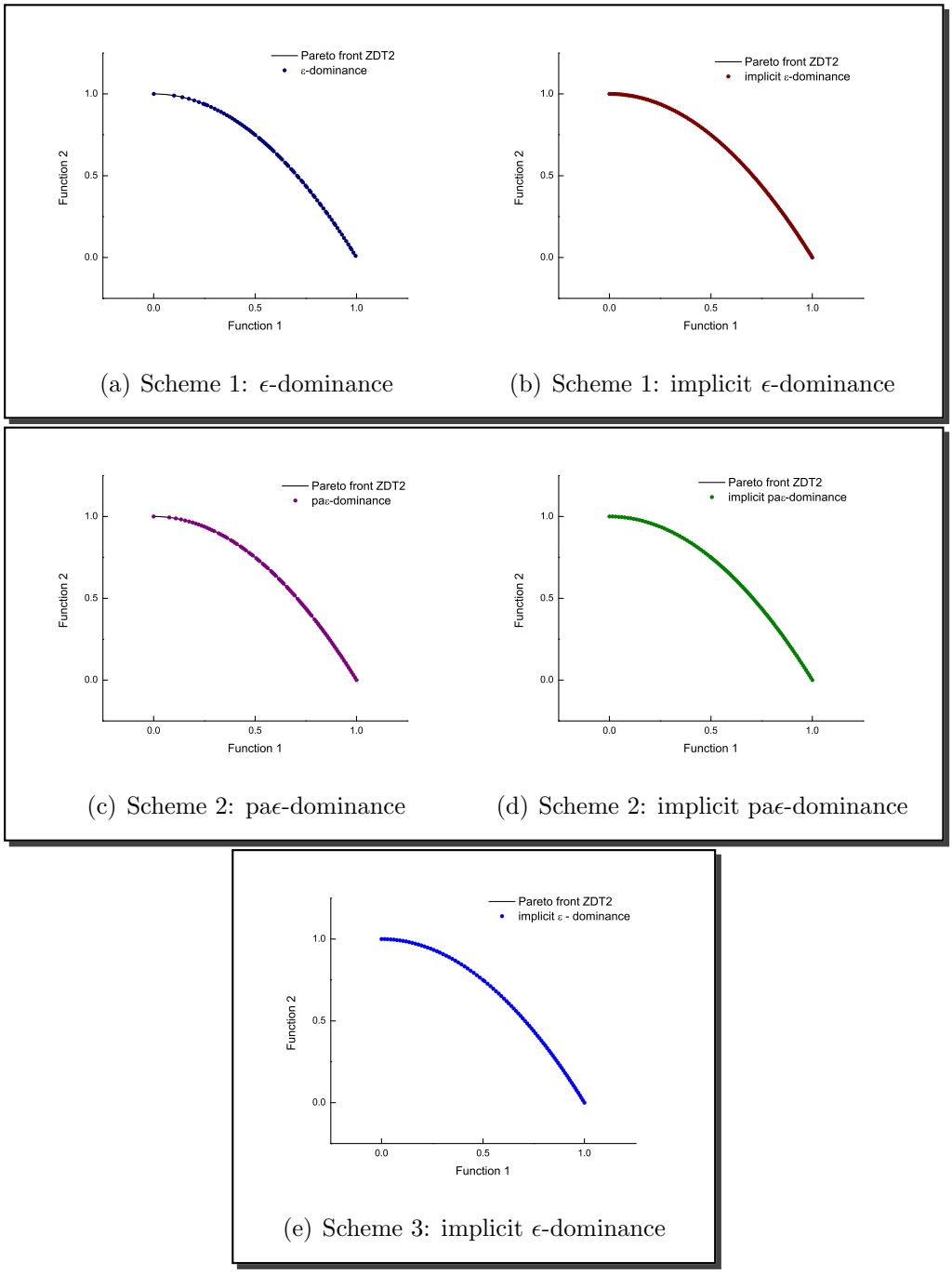
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 12: Pareto fronts obtained for ZDT1.



(a) Scheme 1: ϵ-dominance

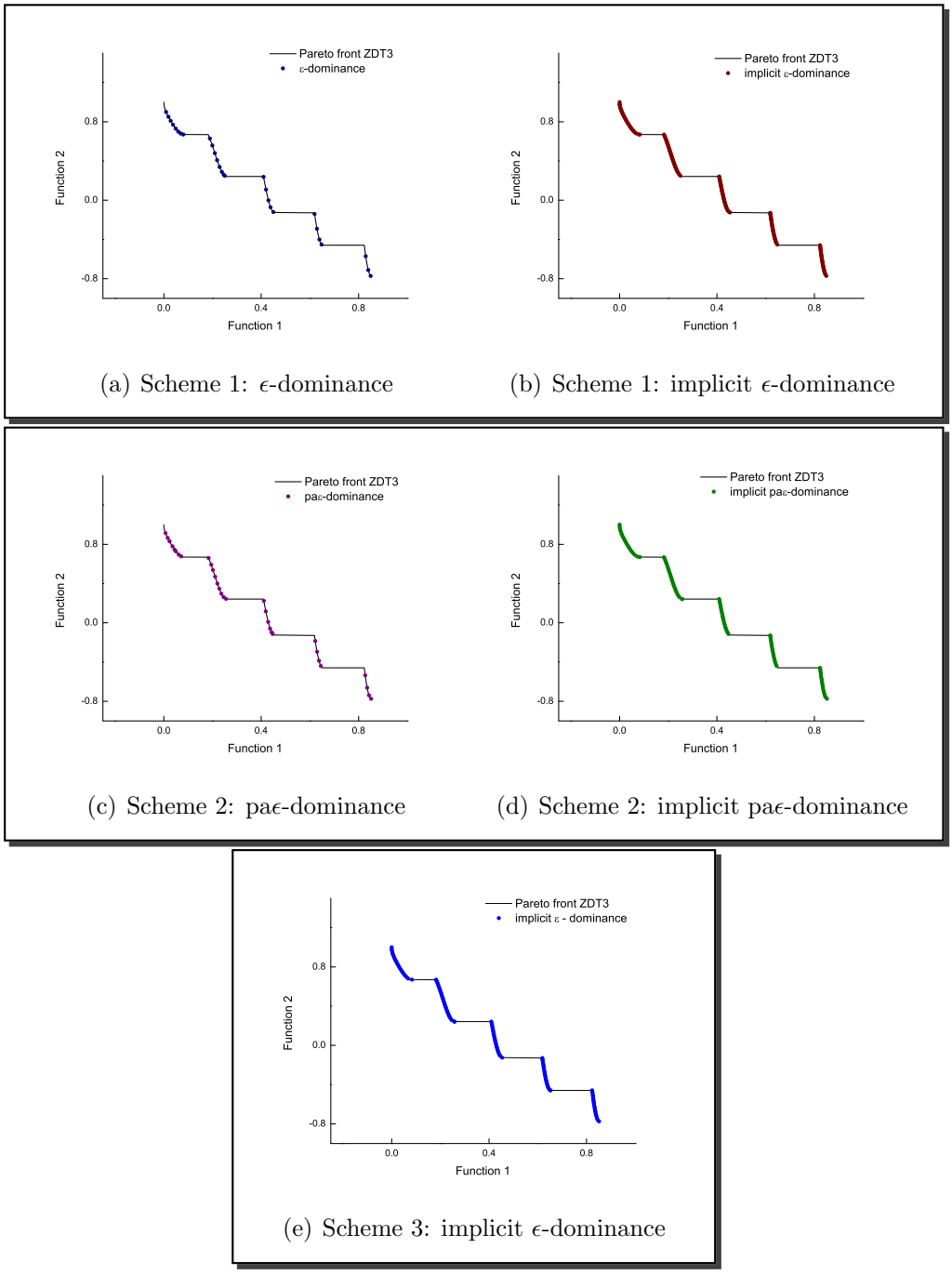
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 13: Pareto fronts obtained for ZDT2.



(a) Scheme 1: ϵ-dominance

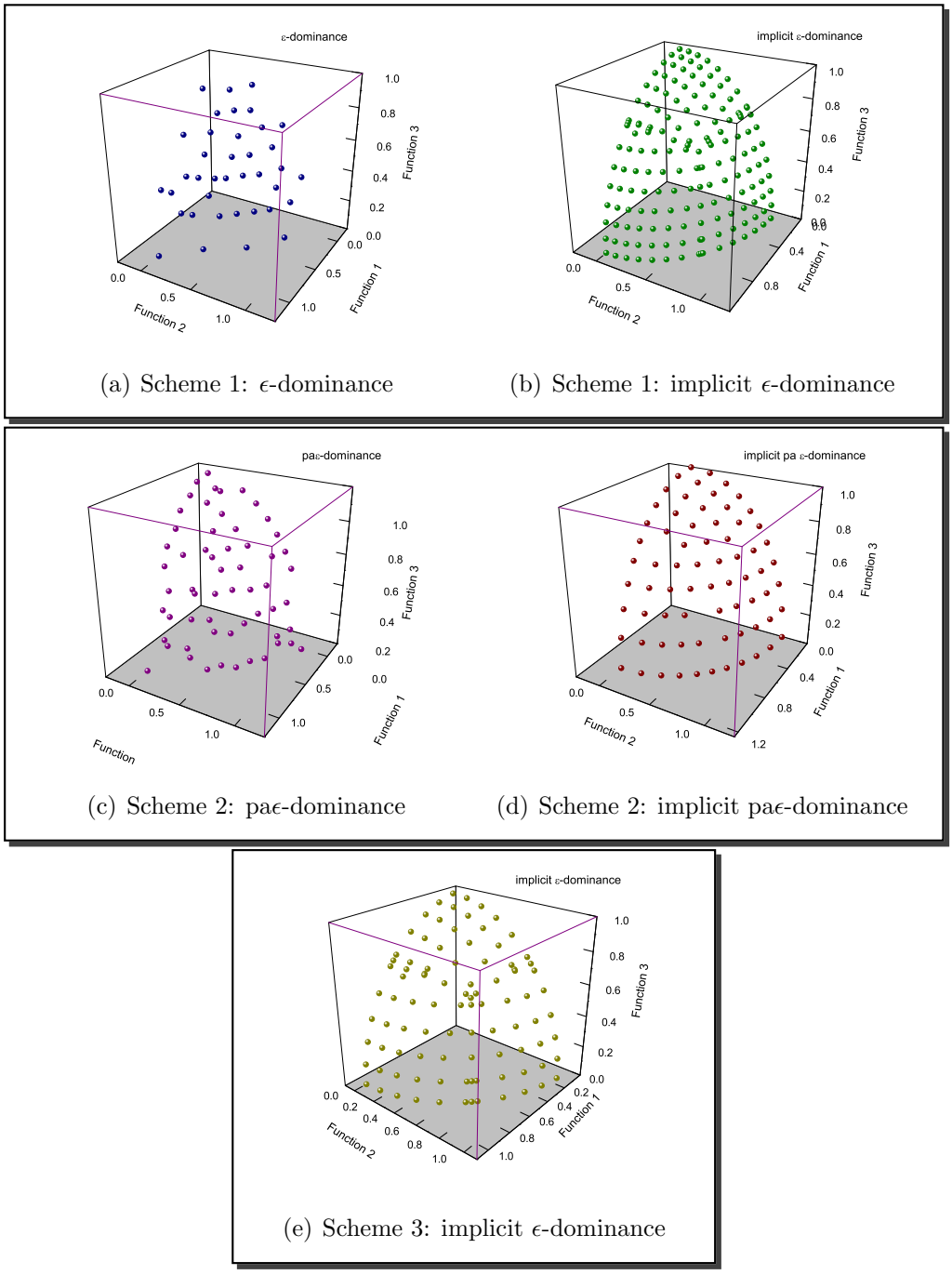
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 14: Pareto fronts obtained for ZDT3.



(a) Scheme 1: ϵ-dominance

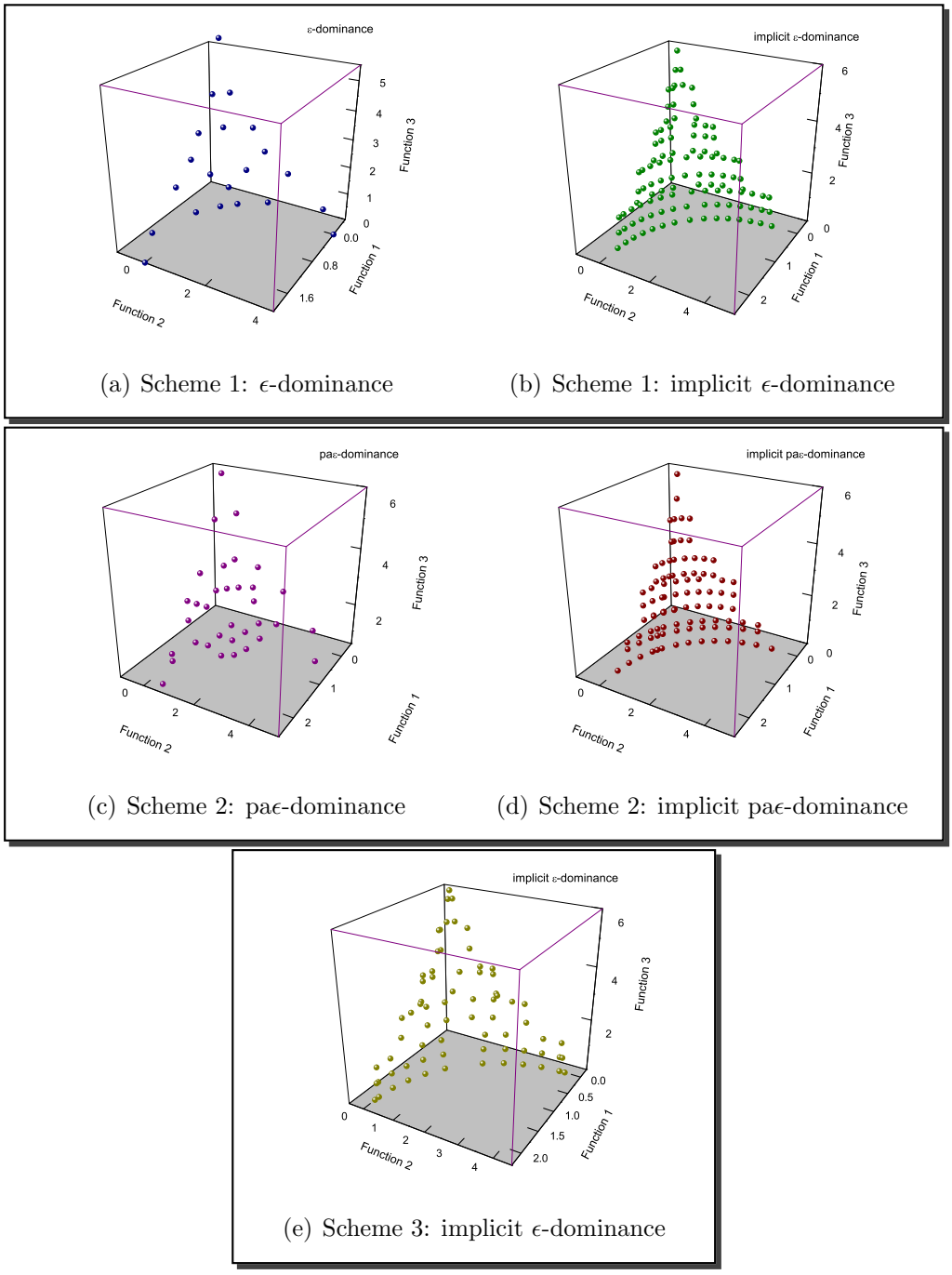
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 15: Pareto fronts obtained for DTLZ2.



(a) Scheme 1: ϵ-dominance

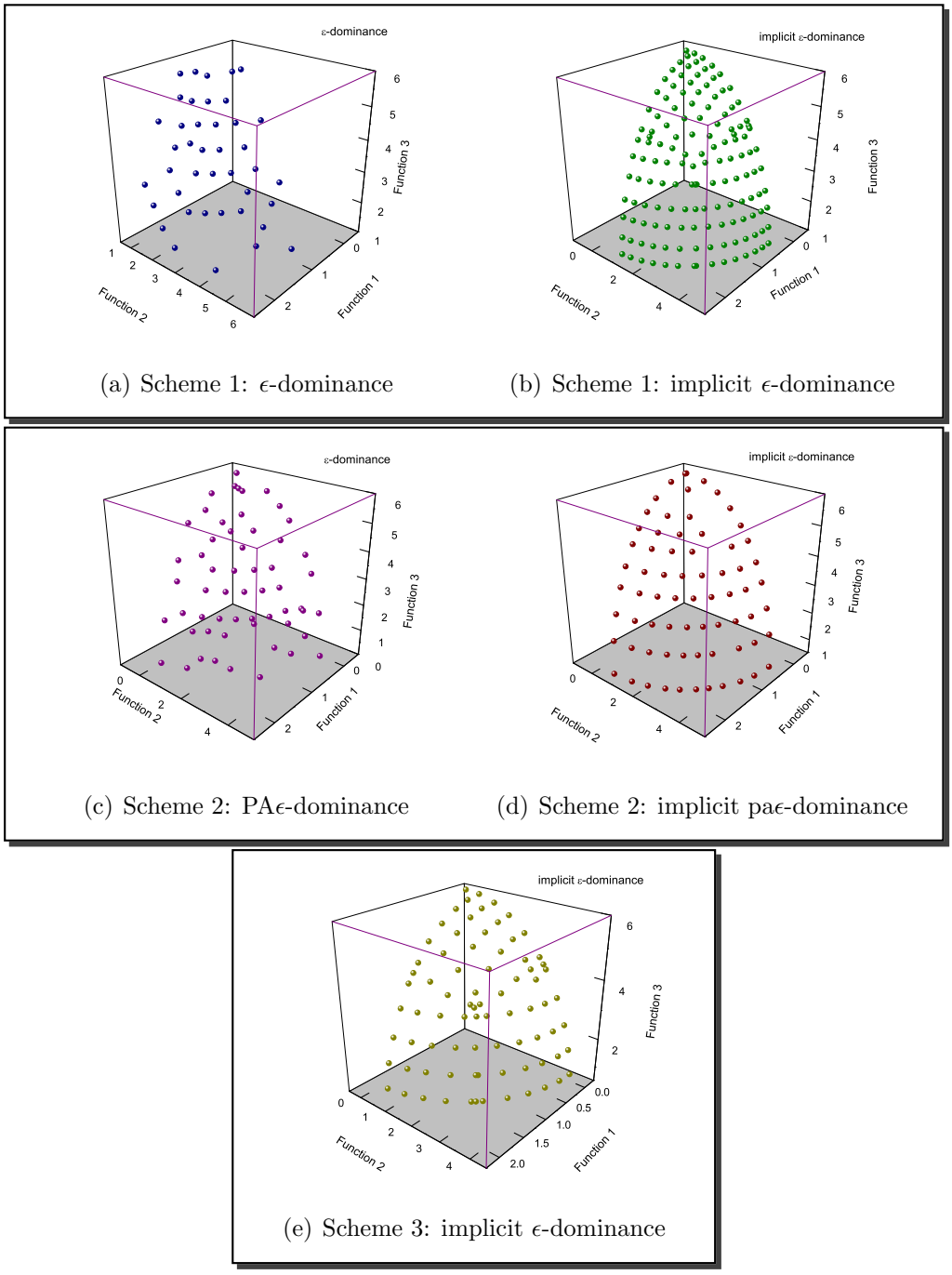
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 16: Pareto fronts obtained for WFG2.



(a) Scheme 1: ϵ-dominance

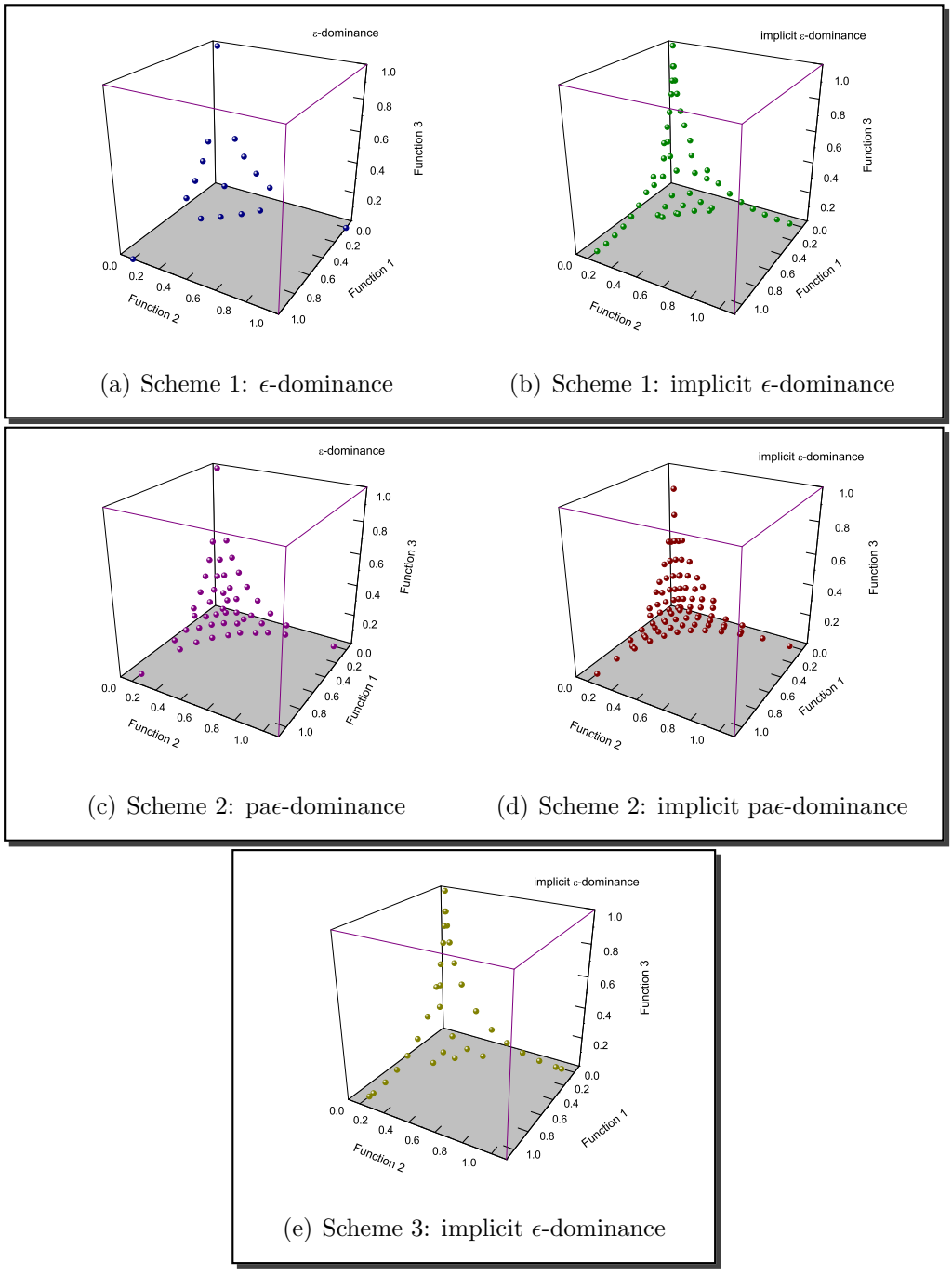
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: PAϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 17: Pareto fronts obtained for WFG4.



(a) Scheme 1: ϵ-dominance

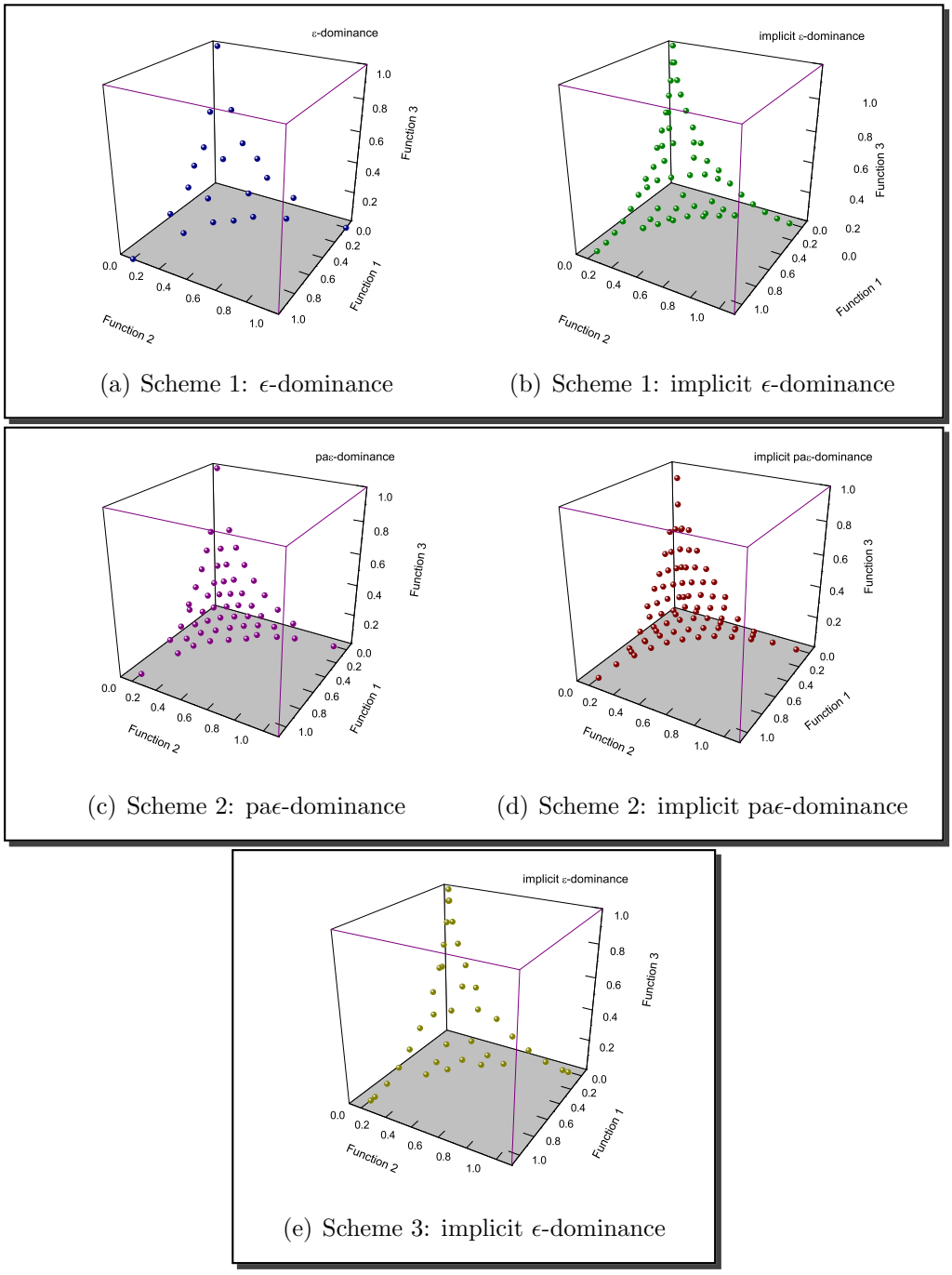
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 18: Pareto fronts obtained for convex50.



(a) Scheme 1: ϵ-dominance

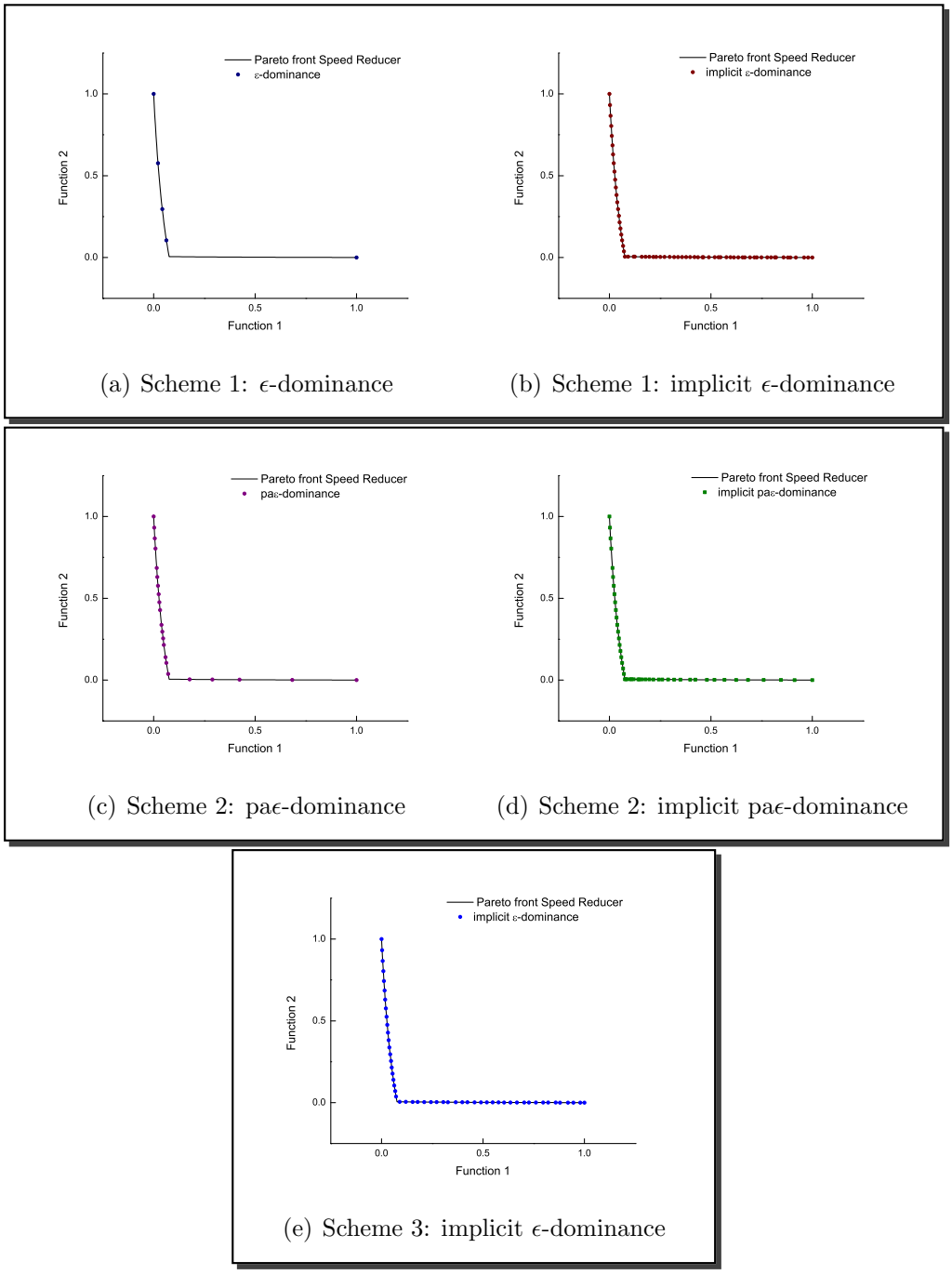
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 19: Pareto fronts obtained for convex60.



(a) Scheme 1: ϵ-dominance

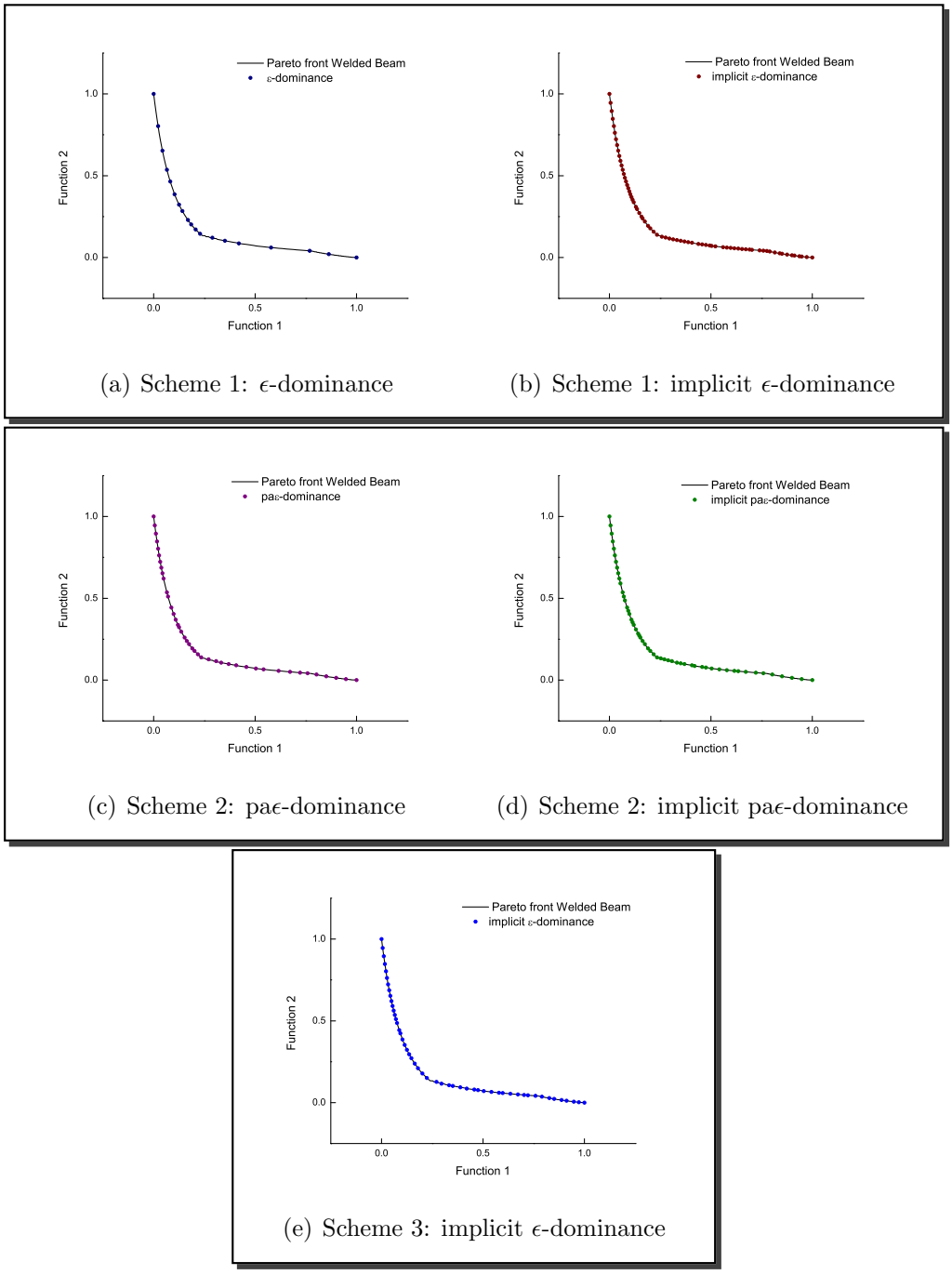
(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

(e) Scheme 3: implicit ϵ-dominance

Figure 20: Pareto fronts obtained for Speed Reducer.



(a) Scheme 1: ϵ-dominance

(b) Scheme 1: implicit ϵ-dominance

(c) Scheme 2: paϵ-dominance

(d) Scheme 2: implicit paϵ-dominance

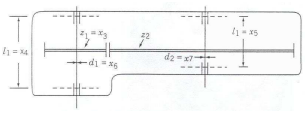
(e) Scheme 3: implicit ϵ-dominance

Figure 21: Pareto fronts obtained for Welded Beam.

Function	Objectives	Bounds	Characteristics
ZDT1	$f_1(\vec{x}) = x_1 \quad \text{and} \quad g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$ $f_2(\vec{x}, g) = 1 - \sqrt{f_1/g(\vec{x})}$	n = 30 $0 \leq x_i \leq 1,$ $i = 1, 2, \ldots, 30$	convex, connected
ZDT2	$f_1(\vec{x}) = x_1 \quad \text{and} \quad g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$ $f_2(\vec{x}, g) = 1 - (f_1/g(\vec{x}))^2$	n = 30 $0 \leq x_i \leq 1,$ $i = 1, 2, \ldots, 30$	nonconvex, connected
ZDT3	$f_1(\vec{x}) = x_1 \quad \text{and} \quad g(\vec{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$ $f_2(\vec{x}, g) = 1 - \sqrt{f_1/g(\vec{x})} - (f_1/g(\vec{x})) \sin(10\pi f_1)$	n = 30 $0 \leq x_i \leq 1,$ $i = 1, 2, \ldots, 30$	disconnected
Deb24	$f_1(x_1) = 4 * x_1$ $f_2(x_1, x_2) = g(x_2) * h(x_1),$ where $g(x_2) = \begin{cases} 4 - 3 * \exp^{-\left(\frac{x_1 - 0.2}{0.02}\right)^2} & \text{if } x_2 \leq 0.4 \quad \text{and} \\ 4 - 2 * \exp^{-\left(\frac{x_1 - 0.7}{0.2}\right)^2} & \text{otherwise.} \end{cases}$ $h(x_1) = \begin{cases} 1 - \left(\frac{f_1(x_1)}{g(x_2)}\right)^{0.25 + (3.75(g(x_2) - 1))} & \text{if } f_1(x_1) \leq g(x_2) \\ 0 & \text{otherwise.} \end{cases}$	$0.1 \leq x_i \leq 1$ $i = 1, 2$	convex, connected
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 $h(x_1) = \begin{cases} 1 - \left(\frac{f_1(x_1)}{g(x_2)}\right)^{10} & if\ f_1(x_1) \leq g(x_2) \\ 0 & \text{otherwise.} \end{cases}$	$0 \leq x_i \leq 1$ $i = 1, 2$	nonconvex, connected
DTLZ2	$f_1(\vec{x}) = \cos(\frac{\pi}{2} x_1) \cos(\frac{\pi}{2} x_2)(1 + g(\vec{x}))$ $f_2(\vec{x}) = \cos(\frac{\pi}{2} x_1) \sin(\frac{\pi}{2} x_2)(1 + g(\vec{x}))$ $f_3(\vec{x}) = \sin(\frac{\pi}{2} x_1)(1 + g(\vec{x}))$ $g(\vec{x}) = \sum_{i=3}^{n} (x_i - 0.5)^2$	n = 12 $0 \leq x_i \leq 1$ $i = 1, \ldots, 12$	nonconvex, connected
WFG2	The full construction of this problem can be found in [12]		nonseparable, unimodal, convex, disconnected
WFG4	The full construction of this problem can be found in [12]		separable, multimodal, nonconvex
Convex50	$f_1(\vec{x}) = x_1,\ f_2(\vec{x}) = x_2,\ f_3(\vec{x}) = (1 - x_1^p - x_2^p)^{1/p}$	$0 \leq x_i \leq 1,$ i = 1,2 $p = 0.5$	convex, connected
Convex60	$f_1(\vec{x}) = x_1,\ f_2(\vec{x}) = x_2,\ f_3(\vec{x}) = (1 - x_1^p - x_2^p)^{1/p}$	$0 \leq x_i \leq 1,$ i = 1,2 $p = 0.6$	convex, connected

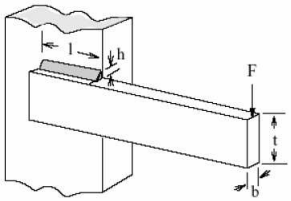
Table 13: Definition and description of each of the 10 unconstrained test problems adopted in this paper.

The objective of Golinski's Speed Reducer problem [13] is to find the minimum volume of a gear box (and, hence, its minimum weight), subject to several constraints. The problem is illustrated in the Figure and there are seven design variables, x1 to x7, which represent:
x1 width of the gear face, in cm
x2 teeth module, in cm
x3 number of pinion teeth
x4 shaft 1 length between bearings, in cm
x5 shaft 2 length between bearings, in cm
x6 diameter of shaft 1, in cm
x7 diameter of shaft 2, in cm



Function	Objectives	Bounds	Characteristics
Speed Reducer	$f_1(\vec{x}) = 0.7854x_1 \cdot x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.5079(x_6^2 + x_7^2)x_1 + 7.477(x_6^3 + x_7^3) + 0.7854(x_4 \cdot x_6^2 + x_5 \cdot x_7^2)$, $f_2(\vec{x}) = \frac{\sqrt{(\frac{745 \cdot x_4}{x_2 x_3})^2 + 1.69e^7}}{0.1x_6^3}$, subject to: $g_1(\vec{x}) \equiv \frac{1}{x_1 \cdot x_2^2 \cdot x_3} - \frac{1}{27} \leq 0$, $g_2(\vec{x}) \equiv \frac{1}{x_1 \cdot x_2^2 \cdot x_3^2} - \frac{1}{397.5} \leq 0$, $g_3(\vec{x}) \equiv \frac{x_4^3}{x_2 \cdot x_3 \cdot x_6^4} - \frac{1}{1.93} \leq 0$, $g_4(\vec{x}) \equiv \frac{x_5^3}{x_2 \cdot x_3 \cdot x_7^4} - \frac{1}{1.93} \leq 0$, $g_5(\vec{x}) \equiv x_2 \cdot x_3 - 40 \leq 0$, $g_6(\vec{x}) \equiv \frac{x_1}{x_2} - 12 \leq 0$, $g_7(\vec{x}) \equiv 5 - \frac{x_1}{x_2} \leq 0$, $g_8(\vec{x}) \equiv 1.9 - x_4 + 1.5x_6 \leq 0$, $g_9(\vec{x}) \equiv 1.9 - x_5 + 1.1x_7 \leq 0$, $g_{10}(\vec{x}) \equiv f_2 - 1300 \leq 0$, $g_{11}(\vec{x}) \equiv \frac{\sqrt{(\frac{745 \cdot x_5}{x_1 \cdot x_2})^2 + 1.575e^8}}{0.1 \cdot x_6^3} - 1100 \leq 0$,	$2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.3 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, $5.0 \leq x_7 \leq 5.5$	convex, connected

The welded beam design problem [21] involves two nonlinear objective functions and seven constraints, in which a uniform beam of rectangular cross section needs to be welded to a base to be able to carry a load of 6000 lb. The objectives are to minimize the cost of fabrication and beam deflection with four constraints and four decision variables, which are represented by:
h weld thickness
l weld length
t beam thickness
b beam width



Function	Objectives	Bounds	Characteristics
Welded Beam	$f_1(\vec{x}) = 1.10471h^2l + 0.04811tb(14.0 + l)$, $f_2(\vec{x}) = \delta(\vec{x})$ subject to: $g_1(\vec{x}) \equiv 13600 - \tau(\vec{x}) \geq 0$, $g_2(\vec{x}) \equiv 30000 - \sigma(\vec{x}) \geq 0$, $g_3(\vec{x}) \equiv b - h \geq 0$, $g_4(\vec{x}) \equiv P_c - 6000 \geq 0$. where: $\tau(\vec{x}) = \sqrt{\frac{(\tau')^2 + (\tau'')^2 + (l\tau'\tau'')}{\sqrt{0.25(l^2 + (h+t)^2)}}}$, $\tau' = \frac{6000}{\sqrt{2} \cdot h \cdot l}$, $\tau'' = \frac{6000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{2\{0.707hl(l^2/12 + 0.25(h+t)^2)\}}$, $\sigma(\vec{x}) = \frac{504000}{t^2 \cdot b}$, $P_c(\vec{x}) = 64746.022(1 - 0.0282346t)tb^3$.	$0.125 \leq h \leq 5.0$, $0.1 \leq l \leq 10.0$. $0.125 \leq b \leq 5.0$, $0.1 \leq t \leq 10.0$.	convex, connected

Table 14: Definition and description of the two engineering optimization problems adopted in this paper.

Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501–525, Winter 2005.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[9] A. Farhang-Mehr and S. Azarm. Diversity Assessment of Pareto Optimal Solution Sets: An Entropy Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 723–728, Piscataway, New Jersey, May 2002. IEEE Service Center.

[10] W. Gong and Z. Cai. An improved multiobjective differential evolution based on pareto-adaptive ϵ-dominance and orthogonal design. *European Journal of Operational Research*, 198(2):576–601, 2009.

[11] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina. Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 15(4):493–517, Winter 2007.

[12] S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.

[13] G. J. Optimal synthesis problems solved by means oj nonmlinear programming and random methods. *Journal of Mechanisms*, 5:287 – 309, 1970.

[14] S. Janson and D. Merkle. A New Multi-objective Particle Swarm Optimization Algorithm Using Clustering Applied to Automated Docking. In M. J. Blesa, C. Blum, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics, Second International Workshop, HM 2005*, pages 128–142, Barcelona, Spain, August 2005. Springer. Lecture Notes in Computer Science Vol. 3636.

[15] J. D. Knowles and D. W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[16] I. Kokolo, K. Hajime, and K. Shigenobu. Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to Optimal? In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 957–962, Piscataway, New Jersey, May 2001. IEEE Service Center.

[17] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.

[18] J. Morse. Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research*, 7(1–2):55–66, 1980.

[19] N. Padhye, J. Branke, and S. Mostaghim. Empirical Comparison of MOPSO Methods - Guide Selection and Diversity Preservation -. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 2516–2523, Trondheim, Norway, May 2009. IEEE Press.

[20] B. Y. Qu and P. N. Suganthan. Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster. In *Congress on Evolutionary Computation (CEC'2009)*, volume 1, pages 2934–2839, Piscataway, New Jersey, May 2009. IEEE Service Center.

[21] K. M. Ragsdell and D. T. Phillips. Optimal design of a class of welded structures using geometric programming. *Journal of Engineering for Industry Series B*, B(98):1021–1025, 1975.

[22] C. R. Raquel and P. C. Naval, Jr. An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. In H.-G. B. et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 257–264, New York, USA, June 2005. ACM Press.

[23] L. V. Santana-Quintero, A. G. Hernández-Díaz, J. Molina, C. A. Coello Coello, and R. Caballero. DEMORS: A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for

constrained problems. *Computers & Operations Research*, 37(3):470–480, 2010.

[24] J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[25] G. Toscano Pulido and C. A. Coello Coello. Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer. In K. D. et al., editor, *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I*, pages 225–237, Seattle, Washington, USA, June 2004. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.

[26] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan. Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Computing*, 14(3):193–209, February 2010.

[27] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712 –731, December 2007.

[28] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[29] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.