

A hybrid method for solving multi-objective global optimization problems

C. Gil · A. Márquez · R. Baños · M. G. Montoya ·
J. Gómez

Received: 28 December 2005 / Accepted: 11 October 2006 / Published online: 22 November 2006
© Springer Science+Business Media B.V. 2006

Abstract Real optimization problems often involve not one, but multiple objectives, usually in conflict. In single-objective optimization there exists a global optimum, while in the multi-objective case no optimal solution is clearly defined but rather a set of optimums, which constitute the so called Pareto-optimal front. Thus, the goal of multi-objective strategies is to generate a set of non-dominated solutions as an approximation to this front. However, most problems of this kind cannot be solved exactly because they have very large and highly complex search spaces. The objective of this work is to compare the performance of a new hybrid method here proposed, with several well-known multi-objective evolutionary algorithms (MOEA). The main attraction of these methods is the integration of selection and diversity maintenance. Since it is very difficult to describe exactly what a good approximation is in terms of a number of criteria, the performance is quantified with adequate metrics that evaluate the proximity to the global Pareto-front. In addition, this work is also one of the few empirical studies that solves three-objective optimization problems using the concept of global Pareto-optimality.

C. Gil (✉) · A. Márquez · R. Baños · M. G. Montoya
Dpt. de Arquitectura de Computadores y Electrónica, Universidad de Almería, Cañada de San Urbano, 04120 Almería, Spain
e-mail: cgil@ace.ual.es

J. Gómez
Dpt. Lenguajes y Computación, Universidad de Almería, Cañada de San Urbano, 04120 Almería, Spain
e-mail: jgomez@ual.es

A. Márquez
e-mail: amarquez@ace.ual.es

R. Baños
e-mail: rbanos@ace.ual.es

M. G. Montoya
e-mail: mari@ace.ual.es

Keywords Multi-objective optimization · Global pareto-optimal front · Evolutionary algorithms

1 Introduction

The aim of global optimization (GO) is to find the best solution of decision models, in presence of multiple local solutions [1]. In recent years, researchers have begun to apply different GO techniques to problems that occur in the analysis and solution of multi-objective linear and non linear programming problems [2,3]. However, when the problem has several objective functions, the notion of optimum changes. Classical optimization methods (including multi-criteria decision-making methods) suggest converting the multi-objective optimization problem (MOP) into a single-objective optimization problem by combining the objectives in a single mathematical function [4]. However, in last two decades most papers dealing with MOPs used the concept of Pareto-dominance [5], where the goal is to find trade-off solutions (widely known as Pareto-optimal solutions) rather than a single solution. In the absence of any further information, none of these Pareto-optimal solutions is considered to be better than the others. Generating the Pareto-optimal set in complex problems is computationally expensive and often infeasible, but using heuristic methods can allow to obtain a representative sample of it.

Conceptually, evolutionary algorithms (EA) [5] base their operation on maintaining a set of solutions (individuals) which evolve in successive iterations (generations) trying to optimize an objective function. In the multi-objective context, it is necessary to design strategies that obtain good solutions in all the objectives. The main advantage of the multi-objective EAs (MOEAs) [6,7] is their ability to find multiple Pareto-optimal solutions in one single run. Over the past decade, a number of MOEAs have been suggested [8–16]. The interest in the design and implementation of hybrid meta-heuristics has increased remarkably in recent years [17]. However, to date there has been little research effort dedicated to the hybridization of meta-heuristics for multi-objective optimization. Some studies [15,16] have shown that combining ideas of different methods allows the quality of the solutions to be improved. Although many MOEAs have adequately demonstrated their ability to solve two-objective problems, there are few studies in problems with three or more objectives. For instance, Deb et al. [18] compared the performance of two Pareto-based MOEAs in some three-objective optimization problems. This paper presents a new hybrid method, msPESA, whose performance is evaluated versus other well-known MOEAs in several test functions of two and three objectives.

The remainder of the paper is organized as follows. Section 2 gives some general Multi-objective optimization concepts, and a brief overview of some well known MOEAs found in the literature. Section 3 describes the new hybrid method here proposed. Section 4 details the test problems and performance metrics used to evaluate the performance of the MOEAs described in Sects. 2 and 3. Experimental results are provided and analyzed in Sect. 5. Finally, we outline the conclusions of this paper in Sect. 6.

2 Multi-objective optimization: concepts and techniques

This section introduces some multi-objective concepts, and briefly describe some well-known methods we have implemented to compare their performance with the new method presented in Sect. 3.

Definition 1 Multi-Objective Optimization is the process of searching for one or more decision variables that simultaneously satisfy all constraints, and optimize an objective function vector that maps the decision variables to two or more objectives.

$$\text{minimize/maximize}(f_k(s)), \forall k \in [1, K]$$

Each decision vector or solution $s = \{s_1, s_2, \dots, s_m\}$ represents accurate numerical qualities for a MOP. The set of all decision vectors constitutes the *decision space*. The set of decision vectors that simultaneously satisfies all the constraints is called *feasible set* (F). The objective function vector (f) maps the decision vectors from the decision space into a K -dimensional objective space $Z \in \mathbb{R}^K, z = f(s), f(s) = \{f_1(s), f_2(s), \dots, f_K(s)\}, z \in Z, s \in F$.

Given a MOP with $K \geq 2$ objectives, instead of giving a scalar value to each solution, a partial order is defined according to Pareto-dominance relations, as we detail below.

Definition 2 Order relation between decision vectors Let s and s' be two decision vectors. The dominance relations in a minimization problem are:

$$s \text{ dominates } s' (s \prec s') \quad \text{iff} \quad f_k(s) < f_k(s') \wedge f_{k'}(s) \not> f_{k'}(s') \quad , \forall k' \neq k \in [1, K]$$

$$s, s' \text{ are incomparable } (s \sim s') \quad \text{iff} \quad f_k(s) < f_k(s') \wedge f_{k'}(s) > f_{k'}(s') \quad , k' \neq k \in [1, K]$$

Definition 3 Pareto-optimal solution A solution s is called *Pareto-optimal* if there is no other $s' \in F$, such that $f(s') < f(s)$. All the Pareto-optimal solutions define the *Pareto-optimal set*.

Definition 4 Non-dominated solution A solution $s \in F$ is *non-dominated* with respect to a set $S' \in F$ if and only if $\nexists s' \in S'$, verifying that $s' \prec s$.

Definition 5 Non-dominated set Given a set of solutions S' , such that $S' \in F$ and $Z' = f(S')$, the function $ND(S')$ returns the set of non-dominated solutions from S' :

$$ND(S') = \{\forall s' \in S' \mid s' \text{ is non-dominated by any other } s'', s'' \in S'\}$$

Figure 1 graphically describes the Pareto-dominance concept for a minimization problem with two objectives (f_1 and f_2). Figure 1a shows the location of several solutions. The filled circles represent non-dominated solutions, while the non-filled ones symbolize dominated solutions. Figure 1b shows the relative distribution of the solutions in reference to s . There exist solutions that are *worse* (in both objectives) than s , *better* (in both objectives) than s , and *incomparable* (better in one objective and worse in the other).

Over the past decade, a number of MOEAs have been proposed [8–16]. In the following we offer a brief description of three methods implemented to be compared with the new MOEA presented in Sect. 3. The first method we have adapted is the strength pareto evolutionary algorithm 2 (SPEA2) [12], which is an improved version

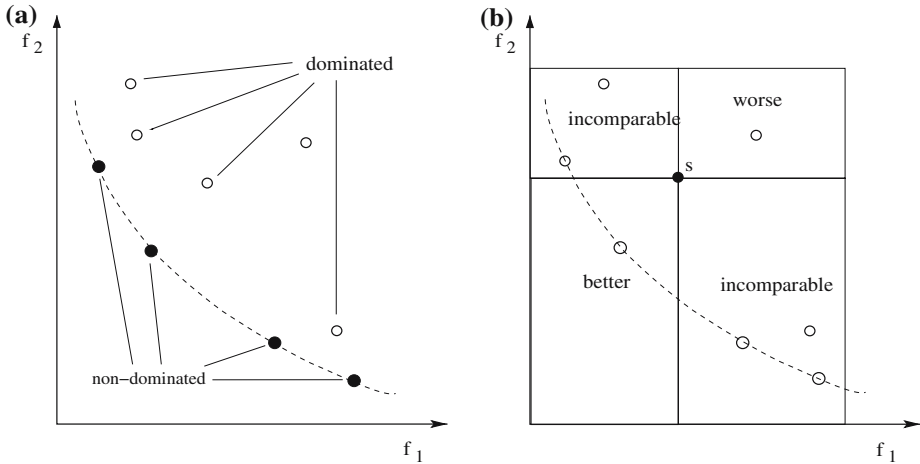


Fig. 1 Pareto-dominance relations in the two-objective case

of the SPEA method [12]. SPEA2 incorporates a fitness assignment strategy that takes into account the number of individuals dominated by each solution and the number of individuals that dominate it, so that best individuals have higher probability to be chosen in the selection process. It uses a nearest neighbor density estimation technique which guides the search more efficiently. In addition, SPEA2 uses a secondary (external) population of solutions to store promising solutions found in the main population, and applies an enhanced archive truncation method that guarantees the preservation of boundary solutions. The second MOEA implemented is the non-dominated sorting genetic algorithm-II (NSGA-II) [11], which is an extended version of NSGA [10]. NSGA-II hierarchizes the population in several categories according to Pareto-dominance relations, and uses elitism and a crowded comparison operator that maintains diversity. Third MOEA analyzed is the pareto envelope-based selection algorithm (PESA) [15], which combines some aspects of SPEA [12] and PAES [14]. It uses a small internal population and a larger external population (archive), where non-dominated solutions found in the main population are stored using a hyper-grid based scheme.

3 msPESA (Mixed spreading between PESA and NSGA-II)

In this section we present a new hybrid MOEA, msPESA, which combines some aspects of PESA, NSGA-II, and local search (LS). Results obtained demonstrate that the synergy of these methods allows an improvement in the quality of the solutions obtained, in comparison with PESA, NSGA-II, and SPEA2. Common aspects of msPESA with other MOEAs are: it uses a population of solutions (internal population, IP), which are improved by applying evolutionary operators (crossover and mutation); it stores promising solutions found in IP in an external population (EP) of solutions; it uses the concept of Pareto-dominance in the selection process; it also uses strategies of diversification and elitism.

The pseudo-code of msPESA is shown in Procedure 1 and operates as follows. Input parameters are the number of objectives ($K \geq 2$), IP size, EP size, and the stop condition (e.g. number of iterations or evaluations). The first step of this procedure consists of initializing the internal population by random solutions. After that, a secondary population (external population, EP) is created to store promising solutions obtained in the IP, which could be lost due to the application of evolutionary operators. Like PESA, msPESA manages EP using a hypergrid based scheme which is dynamically resized according to the extreme values of the non-dominated solutions found in each objective. However, while PESA divides the search space in a K -dimensional grid of boxes, msPESA divides it in a $K-1$ dimensional one. The reason for this variation is that when the search process advances, and therefore the closeness of the solutions to the Pareto-optimal front advances, the use of a K -dimensional space implies that most boxes in the hypergrid are empty. The use of one less dimension less in msPESA is therefore compensated by an increase in the precision of the remaining ones, maintaining the number of boxes constant. Figure 2 describes the difference between using a K -dimensional (PESA) or a $K-1$ dimensional hypergrid (msPESA). As it can be observed in Fig. 2a, in the hypergrid used in PESA 11/16 boxes are empty and, although the other 5/11 have approximately the same number of solutions each, they are not very well distributed. However, a $K-1$ dimensional hypergrid Fig. 2b tends to obtain a better distribution of the solutions if the number of solutions located in each box is approximately the same.

In the selection process, msPESA uses a variation of NSGA-II where only the best front of IP is considered for inclusion in EP. The direct consequence of this variation is that the selection pressure is lower than in NSGA-II. Thus, once the hypergrid has been created, only non-dominated solutions located in IP (those with higher hierarchy) are stored in the hypergrid (EP). While the stop condition is not fulfilled, the algorithm works as follows. While in PESA the child population is generated from EP, msPESA uses IP for this purpose. In particular, children are generated by applying evolutionary operators to each solution (s_1) belonging to IP. Considering that the representation of solutions is not binary, but integer, the crossover operator used consists of calculating the arithmetic average between two parents solutions, while the

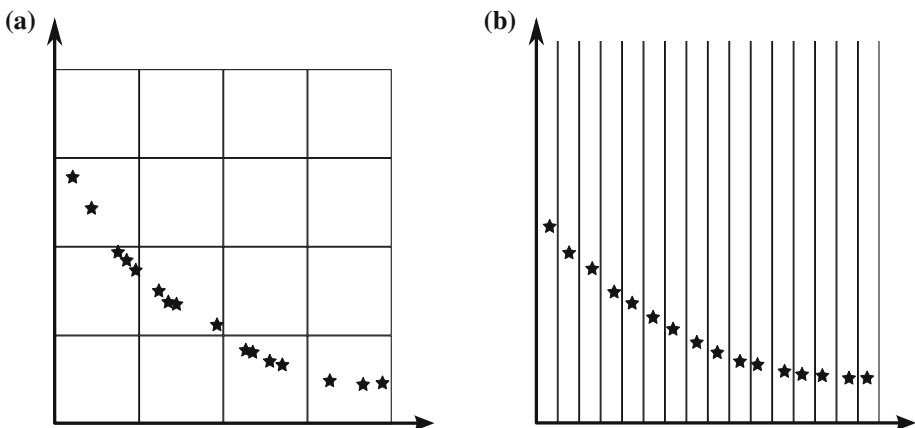


Fig. 2 (a) Hypergrid in PESA, (b) Hypergrid in msPESA

mutation operator is an arbitrary variation of the current solution. Crossover and mutation operators in each iteration according to two probability values $P_{\text{crossover}}$ and P_{mutation} , respectively. If this new solution is not dominated by any solution s' located in IP it is considered as “primary candidate”. A local search procedure is then applied with the aim of increasing the diversity of the solutions. In particular, this local search procedure works by generating a subset of solutions by means of the mutation operator. Solutions located in that subset, whose size $LSsize$ is arbitrary, are considered as “firm candidates”. Each one of these firm candidate solutions (s_2) is included in the EP using Pareto-dominance relations. Thus, s_2 is stored in the EP if it dominates any solution located in EP. If that condition is not fulfilled there is a remaining possibility to include s_2 in EP: if that “firm candidate” (s_2) is incomparable to any solution s' in EP, and the squeeze factor of s_2 is lower than the squeeze factor of s' , s_2 substitutes s' in EP. This approach aims to minimize the maximum number of chromosomes per box, i.e. to improve the distribution of solutions in the search space. As in PAES, the squeeze factor is calculated dynamically, since previous movements may have modified the structure of the hypergrid. All the solutions of EP are located in their corresponding box. The cardinal of that box is then obtained, and therefore the squeeze factor of each solution. When the stop condition is fulfilled, the EP is returned as the non-dominated set obtained by msPESA.

Procedure 1 : Pseudocode of msPESA

```

input:  $K, IPsize, EPsize, LSsize, P_{\text{crossover}}, P_{\text{mutation}}, stop\_condition$ ;
Initialize_IP( $IPsize$ );
Create HyperGrid of  $K-1$  dimensions divided into  $EPsize$  boxes per dimension;
 $EP \leftarrow ND(IP)$ ;
While  $stop\_condition$  is not fulfilled
  For each solution  $s_1 : s_1 \in IP$ 
    Select two random parents  $s_f, s_m : s_f, s_m \in IP$ ;
    If  $P_{\text{crossover}} > \text{random}(0,1)$  Then
       $s_1 \leftarrow \text{crossover}(s_f, s_m)$ ;
    If  $P_{\text{mutation}} > \text{random}(0,1)$  Then
       $s_1 \leftarrow \text{mutation}(s_1)$ ;
    If  $s_1$  is not dominated by any solution  $s' : s' \in EP$  Then
      subset  $\leftarrow \text{LocalSearch}(s_1, LSsize)$ ;
      For each solution  $s_2 : s_2 \in \text{subset}$ 
        If  $s_2$  dominates any solution  $s' : s' \in EP$  Then
           $s_1' \leftarrow s_2$ ;
          Cardinal[ $b$ ]  $\leftarrow$  Count the number of solutions located in the box  $b, \forall b \in \text{Hypergrid}$ ;
          For each solution  $s : s \in EP$ 
            squeeze_factor[ $s$ ]  $\leftarrow$  Cardinal[ $b$ ] :  $b$  is the box where  $s$  is located;
          Else
            squeeze_max  $\leftarrow$  max(squeeze_factor[ $s'$ ] :  $\forall s' \in EP$ );
            If squeeze_factor[ $s_2$ ] < squeeze_max Then
               $s_3 \leftarrow$  Random solution of box with maximum squeeze factor :  $s_3 \in EP$ ;
               $s_2 \leftarrow s_3$ ;
              Cardinal[ $b$ ]  $\leftarrow$  Count the number of solutions located in the box  $b, \forall b \in \text{Hypergrid}$ ;
              For each solution  $s : s \in EP$ 
                squeeze_factor[ $s$ ]  $\leftarrow$  Cardinal[ $b$ ] :  $b$  is the box where  $s$  is located;
  Return (EP);

```

4 Test problem suite

With the aim of evaluating the performance of the MOEAs described in Sect. 3, we propose to apply several test problems. In particular NSGA-II, SPEA2, PESA, and msPESA have been evaluated on eight test problems (five two-objective problems, and three three-objective problems).

4.1 Two-objective test problems

Here we describe five bi-objective test problems [19] we have used in our comparison. Each is structured in the same manner and consists of three functions f_1 , g , h :

$$\begin{aligned} &\text{Minimize } F(x) = (f_1(x_1), f_2(x)) \\ &\text{subject to : } f_2(x) = g(x_2, \dots, x_m) h(f_1(x_1), g(x_2, \dots, x_m)) \\ &\text{where } x = (x_1, \dots, x_m) \end{aligned}$$

Function f_1 is a function of x_1 only, g is a function of the remaining $m-1$ variables, and the parameters of h are the function values of f_1 and g . The test functions differ in these three functions. Each of the functions is a two-objective minimization problem on m parameters.

Test Problem ZDT1 (convex Pareto-optimal front)

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + \frac{9}{m-1} \sum_{i=2}^m x_i \\ h(f_1(x_1), g(x_2, \dots, x_m)) &= 1 - \sqrt{\frac{f_1}{g}} \\ \text{where } m &= 30, x_i \in [0, 1]. \end{aligned}$$

Test Problem ZDT2 (non-convex Pareto-optimal front)

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + \frac{9}{(m-1)} \sum_{i=2}^m x_i \\ h(f_1(x_1), g(x_2, \dots, x_m)) &= 1 - \left(\frac{f_1}{g}\right)^2 \\ \text{where } m &= 30, x_i \in [0, 1]. \end{aligned}$$

Test Problem ZDT3 (discontinuous Pareto-optimal fronts)

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + \frac{9}{(m-1)} \sum_{i=2}^m x_i \\ h(f_1(x_1), g(x_2, \dots, x_m)) &= 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \sin(10\pi f_1) \\ \text{where } m &= 30, x_i \in [0, 1]. \end{aligned}$$

Test Problem ZDT4 (multi-modal problem)

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 10(m-1) + \sum_{i=2}^m (x_i^2 - 10\cos(4\pi x_i)) \\ h(f_1(x_1), g(x_2, \dots, x_m)) &= 1 - \sqrt{\frac{f_1}{g}} \\ \text{where } m &= 10, x_1 \in [0, 1], x_2, \dots, x_m \in [-5, 5] \end{aligned}$$

Test Problem ZDT6 (non-uniformity problem)

$$\begin{aligned} f_1(x_1) &= 1 - \exp^{-4x_1} \sin^6(6\pi x_1) \\ g(x_2, \dots, x_m) &= 1 + \frac{9}{(m-1)} \cdot \sum_{i=2}^m x_i^{25} \\ h(f_1(x_1), g(x_2, \dots, x_m)) &= 1 - (f_1/g)^2 \\ \text{where } m &= 10, x_i \in [0, 1] \end{aligned}$$

4.2 Three-objective test problems

Most papers dealing with multi-objective optimization try to solve two-objective test problems. However, the effect of increasing the number of objectives to optimize becomes an interesting aspect to analyze. With this aim, we evaluate the MOEAs described above in some three-objective optimization problems proposed by Deb et al. [18].

Test Problem DTLZ1 This test function is K -objective problem with a linear Pareto-optimal front:

$$\begin{aligned} \text{Minimize } f_1(x) &= \frac{1}{2}x_1x_2\dots x_{K-1}(1 + g(x_K)), \\ \text{Minimize } f_2(x) &= \frac{1}{2}x_1x_2\dots(1 - x_{K-1})(1 + g(x_K)), \\ &\vdots \\ \text{Minimize } f_{K-1}(x) &= \frac{1}{2}x_1(1 - x_2)(1 + g(x_K)), \\ \text{Minimize } f_K(x) &= \frac{1}{2}(1 - x_1)(1 + g(x_K)) \\ &\text{subject to } 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n \end{aligned}$$

The function $g(x_K)$ requires $|x_K| = w$ variables and must take any function with $g \geq 0$, e.g. the following one:

$$g(x_K) = 100 \left[|x_K| + \sum_{x_i} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right], \quad \forall x_i \in x_K$$

The Pareto-optimal solution corresponds to $x_K = 0$ and the objective function values lie on the linear hyper-plane: $\sum_{k=1}^K f_k = 0.5$. A value of $w = 5$ is suggested here. In the above problem, the total number of variables is $n = K + w - 1$. The difficulty in this problem is to converge to the hyper-plane. The search space contains $(11^w - 1)$ local Pareto-optimal fronts.

Test Problem DTLZ2 This test function is very suitable to evaluate the ability to scale up its performance in a large number of objectives:

$$\begin{aligned} \text{Minimize } f_1(x) &= (1 + g(x_K)) \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \cos(x_{K-2}\pi/2) \cos(x_{K-1}\pi/2), \\ \text{Minimize } f_2(x) &= (1 + g(x_K)) \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \cos(x_{K-2}\pi/2) \sin(x_{K-1}\pi/2), \\ \text{Minimize } f_3(x) &= (1 + g(x_K)) \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \sin(x_{K-2}\pi/2), \\ &\vdots \\ \text{Minimize } f_{K-1}(x) &= (1 + g(x_K)) \cos(x_1\pi/2) \sin(x_2\pi/2), \\ \text{Minimize } f_K(x) &= (1 + g(x_K)) \sin(x_1\pi/2) \\ &\text{subject to } 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n \\ \text{where: } g(x_K) &= \sum_{x_i} (x_i - 0.5)^2, \quad \forall x_i \in x_K \end{aligned}$$

Here, a value of $w = |x_K| = 10$ is used. The total number of decision variables is $n = K + w - 1$. The function g introduces $(3^w - 1)$ local Pareto-optimal fronts,

and one global Pareto-optimal front. All the local Pareto-optimal fronts are parallel to the global Pareto-optimal front. The global Pareto-optimal front corresponds to $x_K = (0.5, \dots, 0.5)^T$.

Test Problem DTLZ7 The last test function we use in our comparison is DTLZ7, which is formed by several disconnected sets of Pareto-optimal regions:

$$\begin{aligned}
 &\text{Minimize } f_1(x_1) = x_1, \\
 &\text{Minimize } f_2(x_2) = x_2, \\
 &\quad \vdots \\
 &\quad \vdots \\
 &\text{Minimize } f_{K-1}(x_{K-1}) = x_{K-1}, \\
 &\text{Minimize } f_K(x) = (1 + g(x_K))h(f_1, f_2, \dots, f_{K-1}, g) \\
 &\text{subject to } 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n \\
 &\text{where : } g(x_K) = 1 + \frac{9}{x_K} \sum_{x_i} x_i, \\
 &h(f_1, f_2, \dots, f_{K-1}, g) = K - \sum_{i=1}^{K-1} \frac{f_i}{(1 + g)} (1 + \sin(3\pi f_i)), \quad \forall x_i \in x_K
 \end{aligned}$$

The test problem has 2^{K-1} disconnected Pareto-optimal regions in the search space. The functional g requires $w = |x_K|$ decision variables and the total number of variables is $n = K + w - 1$. Authors suggest $w = 20$. The Pareto-optimal solutions corresponds to $x_K = 0$. This test problem will evaluate the ability of the MOEAs to maintain the solutions in different Pareto-optimal regions.

4.3 Performance measures

The elaboration of a merit ranking between several methods is not a trivial procedure. In general, an exact order of merit in MOEAs is almost impossible, although it is possible to extract some results from the behavior of each algorithm using accurate metrics. Some authors [20,21] have analyzed a large variety of performance metrics for MOPs. In these studies, metrics are classified in two main categories, according to whether they evaluate the closeness to the Pareto-optimal front, or the diversity in the obtained front.

In the first category, almost all metrics suggested for bi-objective MOPs can be applied to problems with more objectives. In this category we can find the error ratio metric, set coverage metric, generational distance metric, etc.. The second category, which includes the spacing metric, has the drawback that measurements of the spread in bi-objective MOPs cannot be directly applied to problems with three or more objectives. This is because a set of non-dominated solutions with a good spacing metric does not necessarily mean a good distribution of solutions in the entire Pareto-optimal front. Other alternatives, like the maximum spread metric, which measures the Euclidean distance between extreme solutions, do not reveal the true distribution of intermediate solutions. Other authors have proposed applying the hyper-volume metric, in combination with the relative set coverage metric, in the 2, 3, and 4-objective knapsack problem [12]. Our empirical study is focused to evaluate the proximity

of the solutions to the Pareto-optimal front, reason why we have adapted the metrics proposed in SPEA [12], as we describe now.

Definition 6 Set coverage metric (C) Let X, X' be two subsets of solutions. The function C (see Formula (1)) maps the ordered pair (X, X') to the interval $[0,1]$. The value $C(X, X') = 1$ means that all points in X' are dominated by or incomparable to the points of X .

$$C(X, X') \leftarrow \frac{|a' \in X'; \exists a \in X : a \leq a'|}{|X'|} \tag{1}$$

Definition 7 Average size of the space covered (S) Given a set of solutions, $X = \{x_1, x_2, \dots, x_n\}$, the function $S(X)$ returns the average volume enclosed by the union of the polytopes p_1, p_2, \dots, p_K , where each p_i is formed by the intersections of the following hyperplanes arising out of x_i , along with the axes: for each axis in the objective space, there is a perpendicular hyperplane passing through the point $(f_1(x_i), \dots, f_K(x_i))$. In the bi-dimensional case, each p_i represents a rectangle defined by the points $(0,0)$ and (f_1, f_2) . Figure 1b displays how this rectangle is calculated for solution s . Thus, the smaller this average volume is, the better the approximation to the (unknown) Pareto-optimal front is supposed. In order to compare sets of different sizes, the sum of space covered is normalized according to the number of solutions of each non-dominated set, as it is shown in Formula (2).

$$S(X) \leftarrow \frac{\sum_{i=1}^{|X|} \left(\prod_{k=1}^{|K|} \frac{f_k(x_i)}{\max(f_k(X))} \right)}{|X|} \tag{2}$$

5 Experimental results

In the search for an impartial, accurate comparison that excludes the effects of chance, it is necessary to consider many aspects, and for this reason we have made our own implementation of each algorithm, and all of them have been integrated in the same platform.

Since EAs are highly configurable procedures [5], the first step is to assign values to the parameters. Some values are fixed and equal for all the methods during the whole comparison. Although each algorithm could be undoubtedly improved by assigning particular values to the parameters, authors do not suggest this, leaving the users freedom of choice. The genetic operators are the same for all the algorithms, as are the data structures, so the running conditions are equivalent, and the differences between results appear as a consequence of the implementation of the different algorithms. Due to the recommendation in [15], the size of the IP in PESA should be smaller than the EP. For this reason, all the methods use the following IP and EP sizes: {IPsize = 10; EP = 100}. On the other hand, all the algorithms are run for the same number of evaluations of the fitness function. In particular, in the bi-objective problems this number of evaluations is 10000, while in the three-objective problems this value has been increased to 25000, due to the higher complexity of these problems. Crossover and mutation probabilities are the same in all the executions. These probabilities, $\{P_{\text{crossover}} = 0.9, P_{\text{mutation}} = 0.1\}$, are often used in the design of evolutionary algorithms. Finally, in msPESA the subset size is $LSsize = 10$.

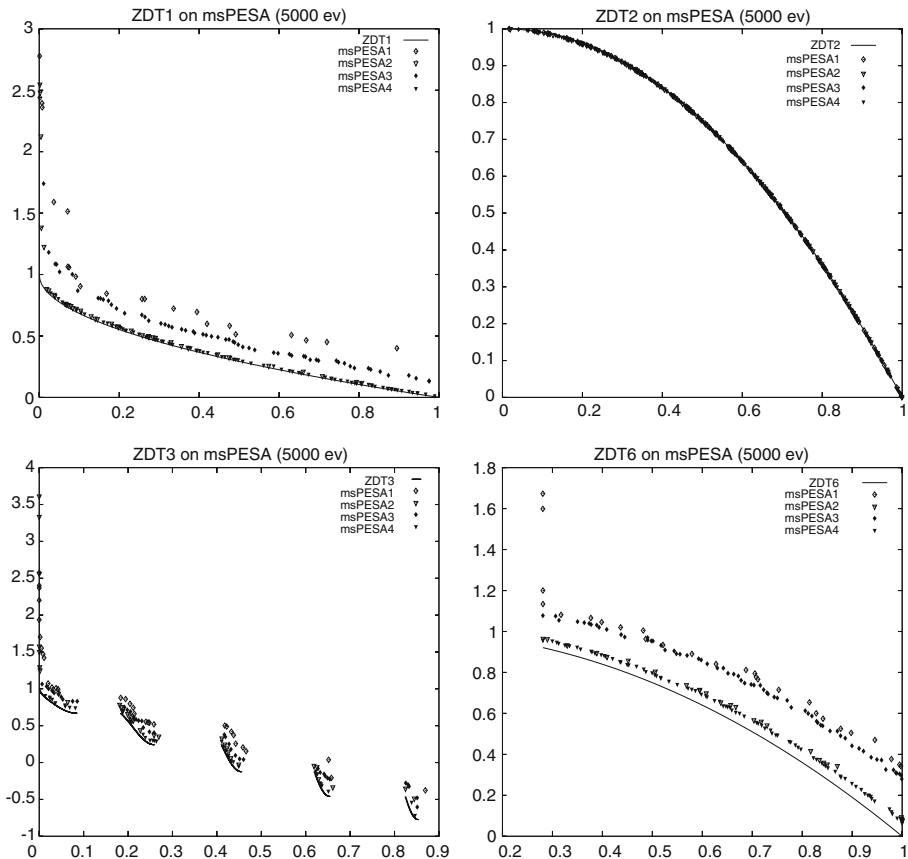


Fig. 3 Non-dominated sets of msPESA variants on four bi-objective functions

Table 1 Average size of the space covered (S) obtained by the variants in the two-objective test problems

	msPESA1	msPESA2	msPESA3	msPESA4
ZDT1	0.16	0.09	0.17	0.10
ZDT2	0.27	0.27	0.27	0.27
ZDT3	0.06	0.01	0.01	0.01
ZDT4	0.05	0.10	0.19	0.10
ZDT6	0.32	0.31	0.44	0.32

The first aspect we analyze is the improvement provided by msPESA. To this end, Table 1 summarizes the average size of the space covered obtained in the two-objective test functions by four variants: msPESA1 (without LS, PESA hypergrid), msPESA2 (with LS, PESA hypergrid), msPESA3 (without LS, msPESA hypergrid), and msPESA4 (with LS, msPESA hypergrid). This table shows that none of these alternatives stands out from the others in all cases, although msPESA4 is well suited.

Table 2 compares in all test functions these four variants using the set coverage metric. Each numerical value in this table indicates that the method located in its row dominates R% solutions of the method located in its column, where R% is that

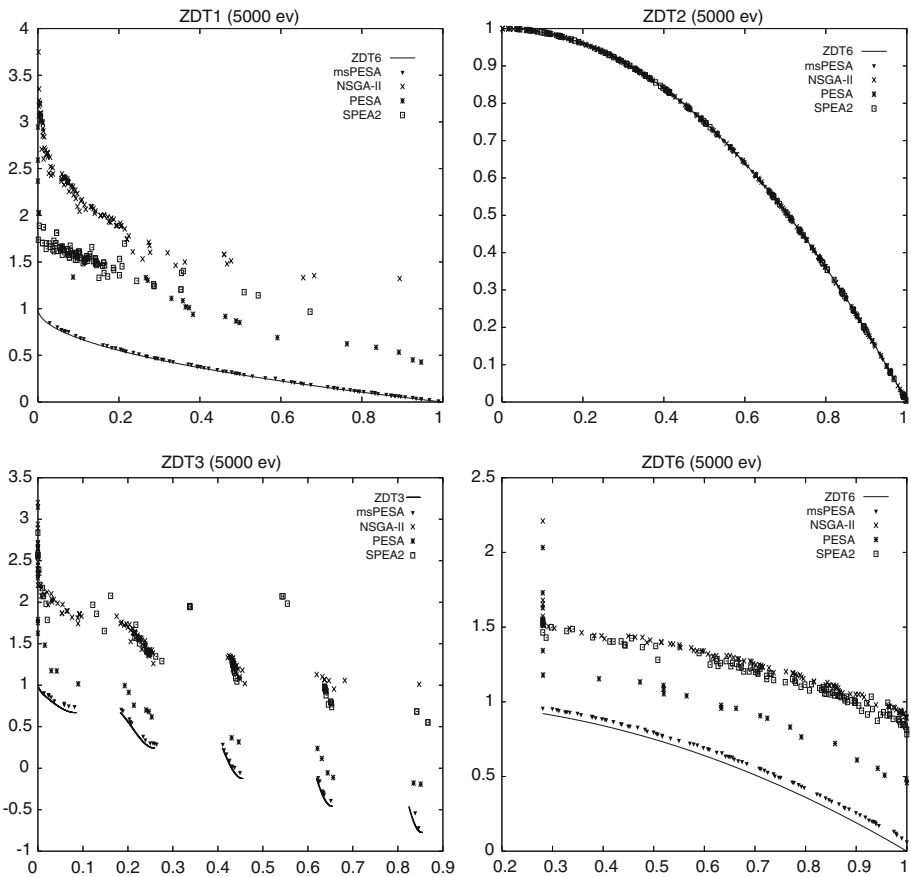


Fig. 4 Non-dominated sets of SPEA2, NSGA-II, PESA, and msPESA on bi-objective functions

value multiplied by 100. On average, solutions obtained by msPESA4 covers 73.0%, 92.7%, and 80.5% of non-dominated solutions obtained by msPESA1, msPESA2, and msPESA3, respectively. On the contrary, only 65.5%, 75.1% and 53.2% of the non-dominated solutions obtained by msPESA4 are covered by msPESA1, msPESA2, and msPESA3, respectively. These results show the advantages obtained by using local search and the new grid strategy (msPESA4).

Figure 3 shows the non-dominated solutions obtained by the four variants described above in ZDT1, ZDT2, ZDT3, and ZDT6. It is clear that msPESA4, obtains non-dominated solutions closer to the Pareto-optimal front. Furthermore, msPESA also obtains the best distribution of solutions along the front. Only 5000 evaluations have been used in these executions in order to notice the fast convergence of msPESA4.

Once we have shown the advantages provided by the simultaneous use of local search and the new grid archiving strategy, we go on to evaluate the performance of msPESA (msPESA4) in comparison with NSGA-II, SPEA2, and PESA. Table 3 shows the results obtained using the metric *S*. As we can observe, msPESA obtains the best performance in some test functions, but not in all cases. In the two-objective

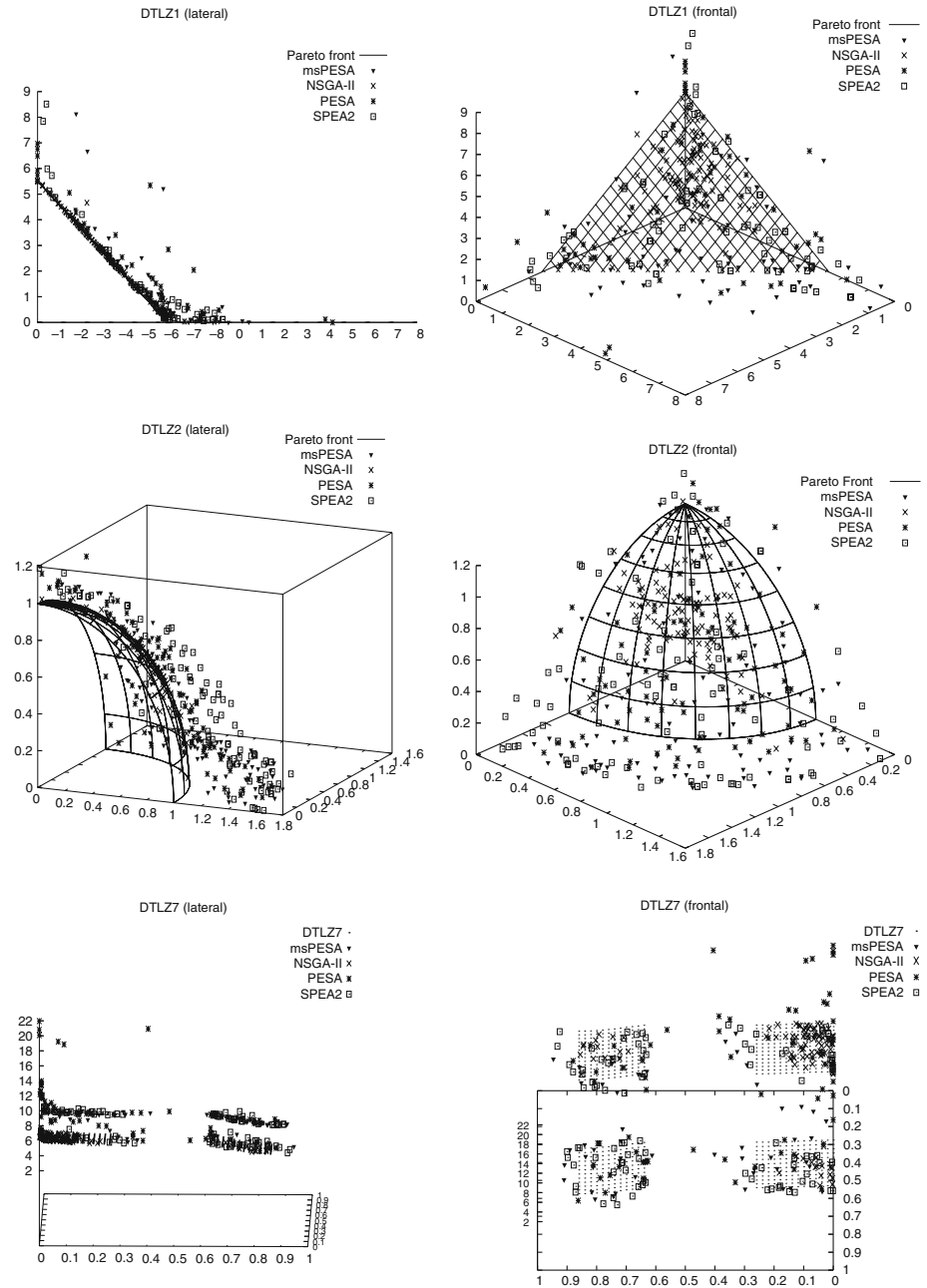


Fig. 5 Non-dominated sets of SPEA2, NSGA-II, PESA, and msPESA on three-objective functions

Table 2 Comparison of msPESA alternatives using set coverage metric (C) in all test problems

	On each test problem				On average			
	msPES A1	msPES A2	msPES A3	msPES A4	msPES A1	msPES A2	msPES A3	msPES A4
msPESA1	<i>ZDT1</i>	0.091	0.227	0.182	<i>ZDT1</i>			
	<i>ZDT2</i>	1.000	1.000	1.000	<i>ZDT2</i>			
	<i>ZDT3</i>	0.303	0.212	0.272	<i>ZDT3</i>			
	<i>ZDT4</i>	0.571	1.000	0.762	<i>ZDT4</i>			
	<i>ZDT6</i>	0.185	0.147	0.037	<i>ZDT6</i>	0.635	0.616	0.655
	<i>DTLZ1</i>	0.980	0.770	1.000	<i>DTLZ1</i>			
	<i>DTLZ2</i>	0.960	0.580	1.000	<i>DTLZ2</i>			
	<i>DTLZ7</i>	0.989	0.989	0.989	<i>DTLZ7</i>			
msPESA2	0.947	<i>ZDT1</i>	1.000	0.789		<i>ZDT1</i>		
	1.000	<i>ZDT2</i>	1.000	1.000		<i>ZDT2</i>		
	0.897	<i>ZDT3</i>	1.000	0.448		<i>ZDT3</i>		
	1.000	<i>ZDT4</i>	1.000	1.000		<i>ZDT4</i>		
	1.000	<i>ZDT6</i>	1.000	0.321	0.664	<i>DTLZ6</i>	0.825	0.751
	0.020	<i>DTLZ1</i>	0.250	0.470		<i>DTLZ1</i>		
	0.450	<i>DTLZ2</i>	0.350	0.980		<i>DTLZ2</i>		
	0.000	<i>DTLZ7</i>	1.000	1.000		<i>DTLZ7</i>		
msPESA3	1.000	0.000	<i>ZDT1</i>	0.034			<i>ZDT1</i>	
	1.000	1.000	<i>ZDT2</i>	1.000			<i>ZDT2</i>	
	1.000	0.375	<i>ZDT3</i>	0.025			<i>ZDT3</i>	
	0.136	0.182	<i>ZDT4</i>	0.227			<i>ZDT4</i>	
	0.985	0.106	<i>ZDT6</i>	0.000	0.718	0.578	<i>ZDT6</i>	0.532
	0.680	1.000	<i>DTLZ1</i>	0.970			<i>DTLZ1</i>	
	0.940	0.960	<i>DTLZ2</i>	1.000			<i>DTLZ2</i>	
	0.000	1.000	<i>DTLZ7</i>	1.000			<i>DTLZ7</i>	
msPESA4	1.000	0.963	1.000	<i>ZDT1</i>				<i>ZDT1</i>
	1.000	1.000	1.000	<i>ZDT2</i>				<i>ZDT2</i>
	1.000	1.000	1.000	<i>ZDT3</i>				<i>ZDT3</i>
	1.000	0.871	1.000	<i>ZDT4</i>				<i>ZDT4</i>
	1.000	1.000	1.000	<i>ZDT6</i>	0.730	0.927	0.805	<i>ZDT6</i>
	0.340	0.940	0.250	<i>DTLZ1</i>				<i>DTLZ1</i>
	0.490	0.640	0.190	<i>DTLZ2</i>				<i>DTLZ2</i>
	0.091	1.000	1.000	<i>DTLZ7</i>				<i>DTLZ7</i>

Table 3 Average size of the space covered (S) obtained by the MOEAs in all test problems

	msPESA	NSGA-II	PESA	SPEA2
ZDT1	0.10	0.24	0.29	0.19
ZDT2	0.27	0.23	0.28	0.23
ZDT3	0.01	0.30	0.05	0.46
ZDT4	0.10	0.05	0.13	0.02
ZDT6	0.32	0.80	0.53	0.81
DTLZ1	5.24	1.61	3.49	3.47
DTLZ2	0.47	0.25	0.36	0.48
DTLZ7	0.28	0.27	0.16	0.21

test problems msPESA clearly obtains the best performance, while in three-objective problems it is slightly worse than SPEA2, and NSGA-II.

Table 4 Comparison of MOEAs using set coverage metric (C) in all test problems

	On each test problem				On average			
	msPESA	NSGA-II	PESA	SPEA2	msPESA	NSGA-II	PESA	SPEA2
msPESA	<i>ZDT1</i>	1.000	1.000	1.000	<i>ZDT1</i>			
	<i>ZDT2</i>	1.000	1.000	1.000	<i>ZDT2</i>			
	<i>ZDT3</i>	1.000	1.000	1.000	<i>ZDT3</i>			
	<i>ZDT4</i>	1.000	1.000	1.000	<i>ZDT4</i>			
	<i>ZDT6</i>	1.000	1.000	1.000	<i>ZDT6</i>	0.935	0.946	0.945
	<i>DTLZ1</i>	0.890	0.790	0.810	<i>DTLZ1</i>			
	<i>DTLZ2</i>	0.820	0.870	0.940	<i>DTLZ2</i>			
	<i>DTLZ7</i>	0.870	0.910	0.810	<i>DTLZ7</i>			
NSGA-II	0.300	<i>ZDT1</i>	0.000	0.000		<i>ZDT1</i>		
	1.000	<i>ZDT2</i>	1.000	1.000		<i>ZDT2</i>		
	0.180	<i>ZDT3</i>	0.010	0.200		<i>ZDT3</i>		
	0.810	<i>ZDT4</i>	0.580	0.510		<i>ZDT4</i>		
	0.050	<i>ZDT6</i>	0.060	0.020	0.666	<i>DTLZ6</i>	0.578	0.585
	1.000	<i>DTLZ1</i>	0.990	1.000		<i>DTLZ1</i>		
	0.990	<i>DTLZ2</i>	0.990	1.000		<i>DTLZ2</i>		
	1.000	<i>DTLZ7</i>	0.990	0.950		<i>DTLZ7</i>		
PESA	0.208	1.000	<i>ZDT1</i>	0.750			<i>ZDT1</i>	
	0.700	0.890	<i>ZDT2</i>	0.800			<i>ZDT2</i>	
	0.308	1.000	<i>ZDT3</i>	1.000			<i>ZDT3</i>	
	0.476	1.000	<i>ZDT4</i>	1.000			<i>ZDT4</i>	
	0.077	0.885	<i>ZDT6</i>	0.808	0.503	0.893	<i>ZDT6</i>	0.872
	0.680	1.000	<i>DTLZ1</i>	0.970			<i>DTLZ1</i>	
	0.750	0.610	<i>DTLZ2</i>	0.940			<i>DTLZ2</i>	
	0.830	0.760	<i>DTLZ7</i>	0.710			<i>DTLZ7</i>	
SPEA2	0.090	1.000	0.360	<i>ZDT1</i>				<i>ZDT1</i>
	1.000	1.000	1.000	<i>ZDT2</i>				<i>ZDT2</i>
	0.100	0.560	0.000	<i>ZDT3</i>				<i>ZDT3</i>
	0.890	0.600	0.520	<i>ZDT4</i>				<i>ZDT4</i>
	0.010	0.960	0.300	<i>ZDT6</i>	0.546	0.815	0.595	<i>ZDT6</i>
	0.770	0.900	0.990	<i>DTLZ1</i>				<i>DTLZ1</i>
	0.540	0.520	0.610	<i>DTLZ2</i>				<i>DTLZ2</i>
	0.970	0.980	0.980	<i>DTLZ7</i>				<i>DTLZ7</i>

Table 4 shows the results obtained by all the MOEAs using the set coverage metric. It can be seen that msPESA obtains the best average result of the four methods. On average, msPESA covers 93.5%, 94.6%, and 94.5% of non-dominated solutions obtained by NSGA-II, PESA, and SPEA2, respectively. On the contrary, only 66.6%, 50.3% and 54.6% of the non-dominated solutions obtained by msPESA are covered by NSGA-II, PESA, and SPEA2, respectively. As this table shows, in the two-objective test functions msPESA covers all the non-dominated solutions obtained by the other methods. In the three-objective test functions, only NSGA-II is able to obtain results slightly better than msPESA.

Figure 4 shows the non-dominated solutions obtained by all the MOEAs in ZDT1, ZDT2, ZDT3 and ZDT6. These graphs are very useful to understand the statistical data provided in previous tables. It is shown that msPESA is able to reach the Pareto-optimal in all cases, while the other methods are further away. Fig. 5 shows the same comparison but in the three-objective test functions, using frontal and lateral views,

respectively. Lateral views provide better information on the proximity to the Pareto-optimal front, while frontal views are suitable to have an idea about the distribution of the solutions. For instance, in the lateral view of DTLZ2 it is clear that SPEA2 is far from the Pareto-optimal front, while the frontal view shows that msPESA obtains a good distribution.

6 Conclusions

In this paper, we have proposed a new hybrid multi-objective evolutionary algorithm based on combining some aspects of NSGA-II, and PESA. Furthermore, this new approach also includes a local search procedure in an attempt to improve the quality of the solutions. The performance of this new multi-objective meta-heuristic has been compared with three well-known MOEAs on a suite of five two-objective and three three-objective test problems. Although establishing an order of merit between different algorithms is a difficult task, the use of accurate metrics can evaluate their performance. Metrics used in our experimental analysis, set coverage metric, and average size of the space covered, evaluate the quality of the non-dominated solutions in terms of closeness to the global Pareto-optimal fronts. First conclusion obtained from the results is that the use of the $K-1$ dimensional hypergrid scheme in combination with local search outperforms the quality of the solutions. On the other hand, results demonstrate that msPESA is able to reach the Pareto-optimal fronts in the two-objective test functions, and be close in the three-objective one. In comparison with the other methods, msPESA obtains the best results in the two-objective test functions, while in the three-objective problems is very close to SPEA2 and NSGA-II.

Acknowledgements This work was supported by the Spanish MCyT under contract TIN2005-00447. Authors appreciate the support of the "Structuring the European Research Area" program, RII3-CT-2003-506079, funded by the European Commission.

References

1. Horst, R., Pardalos, P.M. (eds.): Handbook of global optimization. Kluwer Academic Publishers, Dordrecht (1995)
2. Benson, H.P.: An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple-objective linear programming problem. *J. Global Optim.* **13**, 1–24 (1998)
3. Horst, R., Thoai, N.V.: Maximizing a concave function over the efficient or weakly-efficient set. *Eur. J. Oper. Res.* **117**, 239–252 (1999)
4. Hajela, P., Y-Lin, C.: Genetic search strategies in multi-criterion optimal design. *Struct. Optim.* **4**, 99–107 (1992)
5. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison Wesley, New York (1989)
6. Deb K.: Multi-objective optimization using evolutionary algorithms. John Wiley & Sons (2002)
7. Coello, C. A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers (2002)
8. Fonseca, C. M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, 416–423 (1993)
9. Horn, J., Nafpliotis N., Goldberg D.: A niched pareto genetic algorithm for multiobjective optimization. Proceedings of the First IEEE Conference on Evolutionary Computation, Piscataway, New Jersey, USA, 82–87 (1994)

10. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **2**(3), 221–248 (1994)
11. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: nusa-ii. *Springer Lecture Notes in Computer Science*. **1917**, 849–858 (2000)
12. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* **3**(4), 257–271 (1999)
13. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland (2001)
14. Knowles, J.D., Corne D.W.: The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. *Proceedings of the Congress on Evolutionary Computation*, Piscataway, NJ, 98–105 (1999)
15. Corne, D.W., Knowles, J.D., Oates, H.J.: The pareto-envelope based selection algorithm for multiobjective optimisation. *Springer Lecture Notes in Computer Science*. **1917**, 869–878 (2000)
16. Baños, R., Gil, C., Paechter, B., Ortega, J.: A hybrid meta-heuristic for multi-objective optimization: MOSATS. *J. Math. Model. Algorithm*, DOI: 10.1007/s10852-006-9041-6: (2006)
17. Talbi, E.: A taxonomy of hybrid metaheuristics. *J. Heuristics*. **8**(5), 541–564 (2002)
18. Deb, K., Thiele, L., Laumanns, M., Zitzler, E. Scalable test problems for evolutionary multi-objective optimization. *Evolutionary computation based multi-criteria optimization: theoretical advances and applications*. Springer-Verlag, USA 105–145: (2005)
19. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. *Evolu. Comput.* **8**(2), 173–195 (2000)
20. Veldhuizen, D.V.: Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1999)
21. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Constrained test problems for multi-objective evolutionary optimization. *Proceedings of First International Conference on Evolutionary Multi-Criterion Optimization*, 284–298 (2001)