AbYSS: Adapting Scatter Search for Multiobjective Optimization

A. J. Nebro [1] , F. Luna [1] , E. Alba [1] , A. Beham [b] , B. Dorronsoro [1]

[a] *Departamento de Lenguajes y Ciencias de la Computación*
University of Málaga
Málaga (Spain)

[b] *Institute for Formal Models and Verification*
Johannes Kepler University
Linz (Austria)

TECH-REPORT: ITI-2006-2

Abstract

In this paper we propose a new algorithm for solving multiobjective optimization problems. Our proposal adapts the well-known scatter search template for single objective optimization to the multiobjective domain. The result is a hybrid metaheuristic algorithm called AbYSS, which follows the scatter search structure but using mutation and crossover operators coming from the field of evolutionary algorithms. AbYSS incorporates typical concepts from the multiobjective field, such as Pareto dominance, density estimation, and an external archive to store the nondominated solutions. We evaluate AbYSS with a standard benchmark including both unconstrained and constrained problems, and it is compared against two state-of-the-art multiobjective optimizers, NSGA-II and SPEA2. The obtained results indicate that AbYSS produces very competitive Pareto fronts according to the applied convergence metric, and it clearly outperforms the other two algorithms concerning the diversity of the solutions and the hypervolume metric.

Key words: Multiobjective optimization, Scatter Search, External Archive, Hybridization, Performance Comparison, Diversity
PACS:

Email addresses: antonio@lcc.uma.es (A. J. Nebro), flv@lcc.uma.es (F. Luna), eat@lcc.uma.es (E. Alba), andreas.beham@students.jku.at (A. Beham), dorronsoro@lcc.uma.es (B. Dorronsoro).

[1] This work has been partially funded by the Ministry of Science and Technology and FEDER under contract TIN2005-08818-C04-01 (the OPLINK project).

1 Introduction

Most real world optimization problems involve the optimization of more than one function, which in turn can require a significant computational time to be evaluated. In this context, deterministic techniques are difficult to apply to obtain the set of Pareto optimal solutions of many multiobjective optimization problems (MOPs), so stochastic methods have been widely used and applied. Among them, the use of evolutionary algorithms for solving MOPs has significantly grown in the last years, giving raise to a wide variety of algorithms, such as NSGA-II [1], SPEA2 [2], PAES [3], and many others [4][5].

Scatter Search [6][7][8] is a metaheuristic algorithm that can be considered as an evolutionary algorithm in the sense that it incorporates the concept of population. However, scatter search tries to avoid using many random components, so it does not use typical evolutionary algorithms operators such as mutation nor crossover. Scatter search is based on using a small population, known as the *reference set*, whose individuals are combined to construct new solutions which are obtained in a systematic way. Furthermore, these new individuals can be improved by applying a local search method. The reference set is initialized from an initial population composed of disperse solutions, and it is updated by taking into account the solutions resulting from the local search improvement. Scatter search has been found to be successful in a wide variety of optimization problems [7], but until recently it had not been extended to deal with MOPs.

Our interest here is to adapt the well-known scatter search template [6] to multiobjective optimization. The goal is to design a competitive algorithm capable of improving the resuls produced by state-of-the-art multiobjective evolutionary algorithms, such as NSGA-II and SPEA2. To achieve this objective we are not concerned about using a pure scatter search approach, so we consider the use of mutation and crossover operators if their use contributes to enhance the search capabilities of the algorithm. As a result, our approach, called AbYSS (Archive-Based hYbrid Scatter Search), cannot be considered strictly as scatter search but an hybrid of this algorithm with an evolutionary algorithm.

AbYSS combines ideas of three state-of-the-art evolutionary algorithms for solving MOPs. On the one hand, an external archive is used to store the nondominated solutions found during the search, following the scheme applied by PAES, but using the crowding distance of NSGA-II as a niching measure instead of the adaptive grid used by PAES [3]; on the other hand, the selection of solutions from the initial set to build the reference set applies the density estimation used by SPEA2.

The contributions of our work are summarized in the following:

- We propose a hybrid algorithm based on scatter search for solving constrained as well as unconstrained MOPs. The algorithm is based on incorporating the concepts of Pareto dominance, external archiving, and two different density estimators.
- Several possible configurations for AbYSS are studied in order to get a better understanding of the behavior of the algorithm.
- We exhaustively analyze the performance of AbYSS by comparing it against the algorithms NSGA-II, and SPEA2, using several test functions and metrics taken from the specialized literature.

The rest of the paper is organized as follows. In Section 2, we discuss related works concerning multiobjective optimization and scatter search. Section 3 is devoted to the description of our proposal. A study of the parameters characterizing AbSS is carried out in Section 4. Experimental results, comparing AbYSS with other evolutionary algorithms for solving MOPs, are presented and analyzed in Section 5. Finally, we conclude the paper and give some lines of future work in Section 6.

2 Related Work

The application of scatter search to multiobjective optimization has received little attention until recently. In this section we analyze first the proposals presented in [9], [10], [11], and [12]; after that, several works focused on solving particular MOPs using scatter search are commented.

MOSS [9] is an algorithm that proposes a tabu/scatter search hybrid method for solving nonlinear multiobjective optimization problems. Tabu search is used in the diversification generation method to obtain a diverse approximation to the Pareto-optimal set of solutions; it is also applied to rebuild the reference set after each iteration of the scatter search algorithm. To measure the quality of the solutions, MOSS uses a weighted sum approach. This algorithm is compared against NSGA-II, SPEA-2, and PESA on a set of unconstrained test functions.

Similarly to MOSS, SSPMO [10] is a scatter search algorithm which includes tabu search, although they differ in the use of different tabu search algorithms. SSPMO obtains a part of the reference set by selecting the best solutions of the initial set for each objective functions. The rest of the reference set is obtained by using the usual approach of selecting the remaining solutions in the initial set that maximize the distance to the solutions already in the reference set. In contrast to MOSS, the initial set is updated with solutions obtained in

the scatter search main loop. SSPMO is evaluated by using a benchmark of unconstrained test functions.

SSMO [11] is scatter search algorithm for solving MOPs. It is characterized by using a nondominating sorting procedure to build the reference set from the initial set, and a local search based on a mutation operator is used to improve the solutions obtained from the reference set. A key feature of SSMO is the use of the initial set as a population where all the non-dominated solutions found in the scatter search loop are stored. This algorithm is evaluated with a set of unconstrained and constrained test functions.

A multiobjective scatter search algorithm, called *M-scatter search*, was presented in [12]. The authors used the non-dominated sorting technique and the niched-type penalty method of NSGA [13] for extending the scatter search algorithm to multiobjective optimization. M-scatter search also uses an *offline set* that stores the non-dominated solutions found during the computation. The NSGA niching method is applied in the updating procedure of the offline set for maintaining non-dominated solutions uniformly distributed along the Pareto front.

Compared to these proposals, AbYSS is also applied to solve MOPs with continous bounded variables. However, it follows the steps of the scatter search algorithm, but using mutation and crossover operators. Another difference is that AbYSS uses two different density estimations in the algorithm.

Scatter search has been applied to solve a number of MOPs. In [14] a scatter search algorithm for solving the bi-criteria multi-dimensional knapsack problems is proposed. This algorithm is tailored to solve a specific problem, so the scatter search methods differ significantly of those used in this work.

In [15], the problem of assigning proctors to exams is formulated as a bi-objective problem. However, the authors combined the objective functions to create a single, weighted function and the problem was solved as a mono-objective problem with the standard scatter search scheme. They did not seek for obtaining a set of non-dominated solutions.

The problem of routing school buses in a rural area was addressed in [16]. This is a bi-objective MOP aimed at minimizing, on the one hand, the number of buses to transport students and, on the other hand, the time a given student spends in route. Although the authors developed a solution procedure that searches for a set of efficient solutions instead of a single optimum, the approach used neither Pareto optimality for comparing the solution quality, nor specialized mechanisms for dealing with the set of efficient solutions: the reference set in scatter search is used as common repository for efficient and non-efficient solutions.

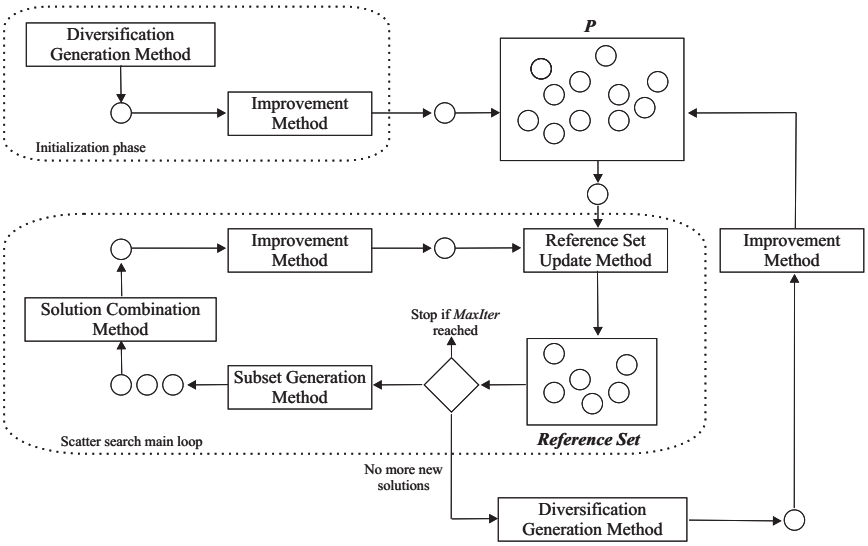

Fig. 1. Outline of the standard scatter search algorithm.

Laminate ply sequence optimization of hybrid composite panels was attempted for simultaneous optimization of both weight of the panel and its cost in [17]. The weighted sum approach was used for solving the multiobjective problem where the two objectives were combined into one overall objective function. The scatter search method used did not incorporate specialized mechanisms for multiobjective functions. Trade-off results were provided by using different values of the weights.

3 Description of the Proposed Algorithm

AbYSS is based on the scatter search template proposed in [6] and its application to solve bounded continuous single objective optimization problems [8]. The template consists of the definition of five methods, as depicted in Fig. 1. The methods are: diversification generation, improvement, reference set update, subset generation, and solution combination.

At this point, we have to note that scatter search is a generic strategy, and many decisions have to be made to design a concrete scatter search algorithm. In particular, the balance between diversification and intensification must be carefully adjusted; otherwise, the algorithm may require a high number of iterations to converge to efficient solutions. In this section, we give a generic description of AbYSS, considering that it is possible to combine different methods and parameters in the algorithm. We make an study of these issues in Section 4.

In the following sections, we first comment the scatter search method according to the template; then we describe the five methods, mainly focusing on the

improvement and reference set update procedures, which constitute the basis of our proposal. After that, we detail how the external archive is managed. Finally, we outline the overall algorithm.

3.1 The Scatter Search Template

The scatter search method starts creating an initial set of diverse individuals in the initialization phase. This phase consists of iteratively generating new solutions by invoking the diversification generation method; each solution is passed to the improvement method, which usually applies a local search, and the resulting individual is added to the initial set. After the initial phase, the scatter search main loop starts.

The main loop begins by building the reference set from the initial set by invoking the reference set update method. Then, solutions in the reference set are systematically grouped in subsets of two or more individuals by means of the subset generation method. In the next step, solutions in each subset are combined to produce a new individual, according to the solution combination method. The improvement method is applied to each new individual, and the final step consists of deciding whether the resulting solution is or not inserted into the reference set. This loop is executed until a stopping condition is fulfilled (for example, a given number of iterations has been performed, or the subset generation method does not produce new subsets).

Optionally, there is a re-start process. The idea is to obtain a new initial set, which includes the individuals in the reference set. The rest of individuals is generated using the diversification generation and improvement methods, as in the initial phase.

3.2 Scatter Search Methods

In order to describe the algorithm, in the following the initial set is named P, and the reference set is known as $RefSet$.

3.2.1 Diversification Generation Method

This method is basically the same one proposed in [8]. The goal is to generate an initial set P of diverse solutions. The method is based on dividing the range of each variable in a number of subranges of equal size; then, each solution is created in two steps. First, a subrange is randomly chosen, with the probability of selecting a subrange being inversely proportional to its frequency count (the

number of times the subrange has been selected); second, a value is randomly generated within the selected range.

3.2.2 Improvement Method

The idea behind this method is to use a local search algorithm to improve the new solutions obtained from the diversification generation and solution combination methods (see Fig. 1). Contrary to [8], where the *simplex* method is used, we have to deal with MOPs which may have constraints, so simplex does not seem adequate. Instead, we propose an improvement method consisting of a simple $(1 + 1)$ Evolution Strategy (ES), which is based on a mutation operator and a Pareto dominance test. This way, we do not follow the strict guidelines of scatter search about avoiding the use of stochastic operators. The justification of this decission has to do with the simplicity of the resulting improvement method and the benefits of using operators that have proven to perform well in evolutionary algorithms. An outline of the method is shown in Fig. 2.

The improvement method is simple. Taking an individual as parameter, it is repeatedly mutated with the aim of obtaining a better individual. The term "better" is defined here in a similar way as the constrained-dominance approach used in NSGA-II [1]. A constraint violation test checks whether two individuals are feasible or not. If one of them is feasible and the other one is not, or both are infeasible but one of them has a smaller overall constraint violation, the test returns the winner. Otherwise, a dominance test is used to decide whether one of the individuals dominates the other one. If the original individual wins, the mutated one is discarded; if the mutated individual wins, it replaces the original one; finally, if they are both non-dominated, the original individual is moved into the external archive and the mutated individual becomes the new original one.

We can point out several features of the proposed improvement method. First, mutated individuals are only evaluated if they are going to replace the original individual. Second, in the case of finding several nondominated solutions in the procedure, they are inserted into the external archive. Finally, we can easily adjust the improvement effort by tuning the parameter *iter*.

3.2.3 Reference Set Update Method

The reference set is a collection of both high quality solutions and diverse solutions that are used to generate new individuals by applying the solution combination method. The set itself is composed of two subsets, $RefSet_1$ and $RefSet_2$, of size p and q, respectively. The first subset contains the best quality solutions in P, while the second subset should be filled with solutions pro-

```
Individual improvement(Individual originalIndividual, int iter) {
  Individual mutatedIndividual
  repeat iter times {
    mutatedIndividual = mutation(originalIndividual)
    if (the problem has constraints) {
      evaluateConstraints(mutatedIndividual)
      best = constraintTest(mutatedIndividual, originalIndividual)
      if (none of them is better than the other one) {
        evaluate(mutatedIndividual)
        best = dominanceTest(mutatedIndividual, originalIndividual)
      } // if
      else if (mutatedIndividual is best)
        evaluate (mutatedIndividual)
    } // if
    else { // the problem has no constraints
      evaluate(mutatedIndividual)
      best = dominanceTest(mutatedIndividual, originalIndividual)
    } // else
    if (mutatedIndividual is the best)
      originalIndividual = mutatedIndividual
    else if (originalIndividual is best)
      delete(mutatedIndividual)
    else { // both individuals are non-dominated
      add originalIndividual to external archive
      originalIndividual = mutatedIndividual
    } // else
  } // repeat
return originalIndividual
} // improvement
```

Fig. 2. Pseudocode describing the improvement method.

moting diversity. In [10] the set $RefSet_2$ is constructed by selecting from P those individuals whose minimum Euclidean distance to $RefSet_1$ is the highest. We keep the same strategy for building $RefSet_2$, but, as it is usual in the multiobjective optimization domain, we have to define the concept of "best individual" to build $RefSet_1$. Additionally, the reference set update method is also used to update the reference set with the new solutions obtained in the scatter search main loop (see Fig. 1). A scheme of this method is included in Fig. 3.

To select the best p individuals of P we use the approach used in SPEA2, i.e., the individuals in P are assigned a fitness value which is the sum of their strength raw fitness and a density estimation [2]. The strength of an individual is the number of solutions it dominates in a population, and its strength raw

```
referenceSetUpdate(bool build) {
  if (build) { // build a new reference set
    select the p best individuals of P
    build the RefSet1 with these p individuals
    compute Euclidean distances in P to obtain q individuals
    build the RefSet2 with these q individuals
  } // if
  else { // update the reference set
    for (each new solution s) {
      test to insert s in RefSet1
      if (test fails)
        test to insert s en RefSet2
        if (test fails)
          delete s
    } // for
  } // else
} // referenceSetUpdate
```

Fig. 3. Pseudocode describing the reference set update method.

fitness is the sum of the strengths of its dominator individuals. The density estimation is based on calculating the distance to the k-th nearest neighbor (see [2] for further details).

Once the reference set is filled, its solutions are combined to obtain new solutions which, after applying the improvement method to them, are checked against those belonging to the reference set. According to the scatter search template, a new solution can become a member of the reference set if either one of the following conditions is satisfied:

- The new individual has better objective function value than the individual with the worst objective value in $RefSet_1$.
- The new individual has a better distance value to the reference set than the individual with the worst distance value in $RefSet_2$.

While the second condition holds in the case of multiobjective optimization, we have again to decide about the concept of best individual concerning the first condition. To determine whether a new solution is better than another one in $RefSet_1$ (i.e., the test to insert a new individual s in $RefSet_1$, as it appears in Fig. 3) we cannot use a ranking procedure because the size of this population usually is small (typically the size of the whole reference set is 20 or less). Our approach is to compare each new solution i to the individuals in $RefSet_1$ using a dominance test. This test is included in Fig. 4. (For the sake of simplicity, we do not consider here constraints in the MOP. The procedure to deal with constraints is as explained in the improvement method in Fig. 2

```
// Test to update the RefSet1 with individual s
bool dominated = false
for (each solution r in RefSet1)
  if (s dominates r)
    remove r from RefSet1
  else if (r dominates s)
    dominated = true
if (not dominated)
  if (RefSet1 not full)
    add s to RefSet1
  else
    add s to the external archive
else // the individual s is dominated
  // test to update the RefSet2 with individual s
  ...
```

Fig. 4. Pseudocode describing the test to add new individuals to $RefSet_1$.

above.)

Let us note that when a new individual is not dominated by the $RefSet_1$, it is inserted into this set only if it is not full. This means that the new individual has to dominate at least one individual in $RefSet_1$. If this condition does not hold, the individual is inserted into the external archive.

3.2.4 Subset Generation Method

According to the scatter search template, this method generates subsets of individuals, which will be used for creating new solutions with the solution combination method. Several kinds of subsets are possible [8]. Usually, this method is used to generate all pairwise combinations of solutions in the reference set. Furthermore, this method should avoid producing repeated subsets of individuals, i.e. subsets previously generated.

In AbYSS this method generates, on one hand, pairwise combinations of individuals in *RefSet1* and, on the other hand, pairwise combinations of individuals in *RefSet2*. Our preliminary experiments (see Section 4) revealed that generating combinations of individuals from the two subsets makes the algorithm to converge poorly. The reason is related to the fact that the combination of individuals from the two RefSets increases the exploration capabilities of the search, thus producing an unbalance between intensification and diversification. As as result, the algorithm requires to perform a large number of iterations to converge to an accurate Pareto front.

3.2.5 Solution Combination Method

The idea of this method in the scatter search strategy is to find linear combinations of reference solutions. After studying this issue in our preliminary tests, we observed that the results were very competitive for many problems, but the algorithm failed when trying to solve some difficult problems. In Section 4 we analyze the use of a simulated binary crossover operator (SBX) [13] instead, concluding that the use of SBX makes AbYSS more robust.

3.3 Managing the External Archive

The main objective of the external archive (or repository) is to store a historical record of the nondominated individuals found along the search process, trying to keep those individuals producing a well-distributed Pareto front. The key issue in the archive management is to decide whether a new solution should be added to it or not.

When a new solution is found in the improvement or the solution combination methods, it is compared with the content of the archive, on a one-per-one basis. If this new solution is dominated by an individual in the archive (i.e., the solution is dominated by the archive), then such solution is discarded; otherwise, the solution is stored in the archive. If there are solutions in the archive that are dominated by the new element, then such solutions are removed. If the archive reaches its maximum allowable capacity after adding the new solution, a decision has to be made for deciding to remove one of its individuals.

The strategy used in other archive-based evolutionary algorithms to decide the individual to be deleted when the archive is full, such as PAES [3] and MOPSO [18], is to divide up the objective function space using an adaptive grid, which is a space formed by hypercubes. Our approach is to use instead the crowding distance used in NSGA-II [1]. The crowding distance is an estimation of the density of solutions surrounding a particular solution in a population (in our case, this population is the external archive), and it is based on calculating the average distance of two points on either side of this point along each of the objectives.

It is worth mentioning here that we could have used the density estimation of SPEA2, as we did in the reference set update method. We decided to use two different density estimations with the aim of hopefully profiting from the combination of them in different parts of our algorithm in the direction of getting a better distributed Pareto front. The rationale of this decision is that the solutions in the archive have passed two filters: first, they are not in the most dense region, according to the crowding distance, and second,

```
construct the initial set P
// outer loop
until (stop condition) {
  referenceSetUpdate(build=true)
  subsetGeneration()
  // scatter search main loop
  while (new subsets are generated) {
    combination()
    for (each combinated individual) {
      improvement()
      referenceSetUpdate(build=false)
    } // for
    subsetGeneration()
  } // while
  // Re-start
  add RefSet1 to P
  move the best n individuals from the archive to P
  fill P with new solutions
} // until
```

Fig. 5. Outline of the AbYSS algorithm.

they are obtained from the best individuals of the initial set according to the density estimation. We have made some experiments comparing the use of only one density estimator in AbYSS, and the combination of both yielded better results.

3.4 Outline of AbYSS

Once the five methods of the scatter search have been proposed and a procedure to manage the external repository has been defined, we are now ready to give an overall view of the technique. The outline of AbYSS is depicted in Fig. 5.

Initially, the diversification generation method is invoked to generate s initial solutions, and each of them is passed to the improvement method. The result is the initial set P. Then, a number of iterations are performed (the outer loop in Fig. 5). At each iteration, the reference set is built, the subset generation method is invoked, and the main loop of the scatter search algorithm is executed until the subset generation method does not produce new subsets of solutions. Then, there is a re-start phase, which consists of three steps. First, the individuals in $RefSet_1$ are inserted into P; second, the best n individuals

in the external archive, according to the crowding distance, are also moved to P; and, third, the diversification generation and improvement methods are used to produce new solutions to filling up the set P.

The idea of moving n individuals from the archive to the initial set is to promote the intensification capabilities of the search. The intensification degree can vary depending on the number of chosen individuals. We use a value of n that is the minimum between the size of the archive and half of the size of P.

The stopping condition of the algorithm can be fixed, or it can depend on other conditions; here, we have used the computation of a predefined number of fitness evaluations (see Section 5).

4 Study of Different Parameter Configurations

As commented before, several issues should be studied to make decisions in order to set a number of parameters defining the behavior of AbYSS. Although an extensive analysis of the parameters of AbYSS is out the scope of this paper, we include here a study of a number of them. In particular, we focus on the use of the SBX crossover operator in the solution combination method, the size of the set P, the generation of all pairwise combinations of individuals in the reference set in the subset generation method, and the number of iterations in the improvement method.

We describe next the metrics we consider in this paper to evaluate the performance of the algorithms. These metrics will be also used in Section 5.

4.1 Performance Metrics

For assessing the performance of algorithms on the tests problems, two different issues are normally taken into account: minimize the distance of the Pareto front generated by the proposed algorithm to the exact Pareto front, and to maximize the spread of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible. To determine the first issue it is usually necessary to know the exact location of the true Pareto front; in this work we have obtained these fronts using an enumerative search strategy (an exception are the ZDTx problem family, whose fronts can be easily obtained because their solutions are known).

The performance metrics can be classified into three categories depending on whether they evaluate the closeness to the Pareto front, the diversity in the obtained solutions, or both [5]. We have adopted one metric of each type.

- **Generational Distance** This metric was introduced by Van Veldhuizen and Lamont [19] for measuring how far the elements are in the set of non-dominated vectors found so far from those in the Pareto optimal set and is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}, \tag{1}$$

where n is is the number of vectors in the set of non-dominated solutions found so far and d_i is the Euclidean distance (measured in objective space) between each of these solutions and the nearest member of the Pareto optimal set. It is clear that a value of $GD = 0$ indicates that all the generated elements are in the Pareto optimal set. In order to get reliable results, non-dominated sets are normalized before calculating this distance measure.

- **Spread** The *Spread* metric [1] is a diversity metric that measures the extent of spread achieved among the obtained solutions. This metric is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}}, \tag{2}$$

where d_i is the Euclidean distance between consecutive solutions, $\bar{d}$ is the mean of these distances, and d_f and d_l are the Euclidean distances to the *extreme* (bounding) solutions of the exact Pareto front in the objective space (see [1] for the details). This metric takes a value zero for an ideal distribution, pointing out a perfect spread out of the solutions in the Pareto front. We apply this metric after a normalization of the objective function values (see Appendix A for a further analysis of this issue).

- **Hypervolume** This metric calculates the volume (in the objective space) covered by members of a non-dominated set of solutions Q (the region enclosed into the discontinuous line in Fig. 6, $Q = \{A, B, C\}$) for problems where all objectives are to be minimized [20]. Mathematically, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its hypervolume (HV) is calculated:

$$HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right). \tag{3}$$

Algorithms with larger values of HV are desirable. Since this metric is not free from arbitrary scaling of objectives, we have evaluated the metric by using normalized objective function values.

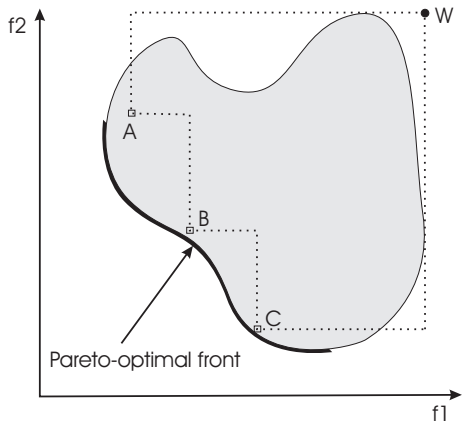

Fig. 6. The hypervolume enclosed by the non-dominated solutions.

4.2 Analysis of the Proposed Configurations

We have performed five experiments, which have been applied to solve five representative problems usually included in similar studies. The problems are: ZDT1 (convex), ZDT2 (nonconvex), ZDT3 (disconnected), ZDT4 (nonconvex, multimodal), and ZDT6 (nonconvex, nonuniformly spaced) [21]. They are fully described in Section 5.1.

The following set of parameters are used by default: we use a polynomial mutation operator with a distribution index equal to 20, the number of iterations in the improvement method is 5, and the sizes of both $RefSet_1$ and $RefSet_2$ is 10.

(1) **Experiment 1**: We configure AbYSS with what can be considered as a typical set of values: we use linear combinations to create new individuals in the solution combination method, the size of P is 100, and the subset generation method generates all pairwise combinations of individuals belonging to both $RefSet_1$ and $RefSet_2$.
(2) **Experiment 2**: We repeat the experiment 1, but using the SBX operator in the solution combination method. The distribution index of this operator has a value of 20.
(3) **Experiment 3**: In this experiment, we keep the same parameter settings as in the previous experiment, but the size of P is reduced down to 20 individuals.
(4) **Experiment 4**: AbYSS is configured as in the third experiment, but the subset generation method produces pairs of individuals belonging only to $RefSet_1$ or $RefSet_2$.
(5) **Experiment 5**: In the last experiment, we repeat the experiment 4, but we turn again to use linear combinations instead of the SBX crossover.

The algorithm stops when 25,000 function evaluations have been computed.

We have made 30 independent runs of each experiment, and the obtained results are shown in Table 1, which includes the mean, $\bar{x}$, and standard deviation, σ_n. The best result is marked in boldface (in the case of comparing values with the same mean, we choose the one with lowest standard deviation).

Table 1
Results of executing different configurations of AbYSS.

	GENERATIONAL DISTANCE				
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Problem	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$
ZDT1	$\mathbf{0.0001}_{\pm 1.8e-05}$	$0.00011_{\pm 2.4e-05}$	$0.00011_{\pm 2.4e-05}$	$0.00012_{\pm 3.1e-05}$	$0.0001_{\pm 3.4e-05}$
ZDT2	$\mathbf{4.6e\text{-}05}_{\pm 1.8e-06}$	$4.7e\text{-}05_{\pm 1.7e-06}$	$4.6e\text{-}05_{\pm 2.4e-06}$	$4.7e\text{-}05_{\pm 1.9e-06}$	$4.6e\text{-}05_{\pm 2.1e-06}$
ZDT3	$0.00019_{\pm 1e-05}$	$0.0002_{\pm 1.5e-05}$	$0.00019_{\pm 1.5e-05}$	$\mathbf{0.00019}_{\pm 8.8e-06}$	$0.00019_{\pm 1.1e-05}$
ZDT4	$13_{\pm 6}$	$5.9_{\pm 7.4}$	$3.9_{\pm 3.4}$	$\mathbf{0.011}_{\pm 0.039}$	$0.9_{\pm 0.48}$
ZDT6	$0.56_{\pm 0.36}$	$0.087_{\pm 0.057}$	$0.086_{\pm 0.05}$	$\mathbf{0.00079}_{\pm 0.00049}$	$0.1_{\pm 0.11}$
	SPREAD				
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Problem	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$
ZDT1	$\mathbf{0.09231}_{\pm 0.0096}$	$0.1025_{\pm 0.012}$	$0.1089_{\pm 0.012}$	$0.1145_{\pm 0.012}$	$0.1192_{\pm 0.019}$
ZDT2	$\mathbf{0.08995}_{\pm 0.012}$	$0.1015_{\pm 0.012}$	$0.1089_{\pm 0.011}$	$0.1121_{\pm 0.011}$	$0.1059_{\pm 0.014}$
ZDT3	$0.7128_{\pm 0.019}$	$0.7077_{\pm 0.021}$	$0.7149_{\pm 0.015}$	$\mathbf{0.6978}_{\pm 0.028}$	$0.7043_{\pm 0.0037}$
ZDT4	$0.6226_{\pm 0.12}$	$0.4418_{\pm 0.23}$	$0.4704_{\pm 0.2}$	$\mathbf{0.2785}_{\pm 0.16}$	$0.3953_{\pm 0.16}$
ZDT6	$0.5131_{\pm 0.22}$	$0.4009_{\pm 0.11}$	$0.4554_{\pm 0.11}$	$\mathbf{0.1546}_{\pm 0.033}$	$0.1546_{\pm 0.2}$
	HYPERVOLUME				
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Problem	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$
ZDT1	$\mathbf{0.662}_{\pm 1.5e-05}$	$0.662_{\pm 2.3e-05}$	$0.662_{\pm 2.1e-05}$	$0.662_{\pm 2.1e-05}$	$0.662_{\pm 3.3e-05}$
ZDT2	$\mathbf{0.3287}_{\pm 2.1e-05}$	$\mathbf{0.3287}_{\pm 2.1e-05}$	$0.3287_{\pm 2.6e-05}$	$0.3287_{\pm 2.6e-05}$	$0.3287_{\pm 3e-05}$
ZDT3	$0.516_{\pm 6e-05}$	$0.5159_{\pm 7.7e-05}$	$0.516_{\pm 3.5e-05}$	$0.5158_{\pm 0.00065}$	$\mathbf{0.516}_{\pm 1.3e-05}$
ZDT4	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$	$\mathbf{0.5796}_{\pm 0.17}$	$0.0007_{\pm 0.004}$
ZDT6	$0.1544_{\pm 0.17}$	$0.09795_{\pm 0.058}$	$0.1066_{\pm 0.082}$	$\mathbf{0.3958}_{\pm 0.0011}$	$0.3651_{\pm 0.1}$

We can observe that there are not important differences when solving the problems ZDT1, ZDT2, and ZDT3 according to the three metrics. The main differences among the experiments arise in the problems ZDT4 and ZDT6. We now analyze each experiment in detail.

Experiment 1 The results show that the configuration used in this experiment produces the best results in the problems ZDT1 and ZDT2. Considering the problem ZDT3, the values of the generational distance and hypervolume metrics are similar to the other experiments, although the spread value is the higuest. With this configuration, AbYSS fails to solve the problem ZDT4, which is indicated by the high Generational Distance value and the fact of obtaining a value of 0 in the hypervolume metric. This indicates that all the

points are out of the limits of the true Pareto front. This experiment produces also the worst results in the problem ZDT6.

Experiment 2 The results of this experiment (see Table 1, third column) show that using the SBX operator in the solution combination method the values of the three metrics are very close to the ones produced by the first experiment considering the problems ZDT1, ZDT2, and ZDT3. Regarding the problems ZDT4 and ZDT6, the convergency and diversity metrics produce better values, although there are not improvements in the hypervolume metric.

Experiment 3 As indicated before, in this experiment we reduce the size of the initial set P from 100 to 20 individuals. We consider this reduction because preliminary experiments seemed to indicate that using large sizes for P had a negative influence in the convergence of the algorithm when solving some problems. The results of this experiment does not allow us to confirm this claim; we observe that although the values of the generational distance in the problems ZDT4 and ZDT6 are slightly better than the previous experiments, the diversity gets worse compared to Experiment 2.

Experiment 4 With the idea in mind of investigating whether the diversification/intensification balance of AbYSS is penalized if the the subset generation method produces pairs of individuals belonging to $RefSet_1$ and $RefSet_2$, in this experiment we allow only combinations of individuals belonging to the same set. The results (see Table 1, fifth column) show that this configuration substantially enhances the solutions of the problems ZDT4 and ZDT6, while keeping practically the same values in the other problems compared to Experiment 3.

Experiment 5 In Experiment 4 we achieved a configuration that successfully solved the considered problems. To analyze the influence of the crossover operator in the solution combination method we use again linear combinations instead of the SBX operator. The results of this last experiment do not improve those obtained in Experiment 4.

A conclusion of these experiments is that the SBX crossover is clearly superior than the linear combinations operator suggested by the orthodox scatter search strategy. A similar result was obtained in [22] in the context of single-objective continuous optimization, where the BLX-α operator also improved the efficacy of stater search with respect to the classical linear combinations method.

To conclude this section, we can state that the parameter settings of the experiment 4 are the most promising of the tested ones for AbYSS. Nevertheless, the configurations used in the rest of experiments are worth of being taken into account to face some problems given the excellent results obtained in the instances ZDT1, ZDT2, and ZDT3. Now that we have decided the set of parameters charaterizing AbYSS, we are ready to make a deeper evaluation of our proposal, including a comparison against two state-of-the-art evolutionary algorithms for solving MOPs.

5 Evaluation

Several test functions have been taken from the specialized literature to compare our approach. In order to know how competitive AbYSS is, we decided to compare it against two algorithms that are representative of the state-of-the-art. These algorithms are NSGA-II and SPEA2. Next, we briefly describe these algorithms, including the parameter settings used in the subsequent experiments.

- **Nondominated Sorting Genetic Algorithm II**: The NSGA-II algorithm was proposed by Deb *et al.* [1]. It is based on obtaining a new population from the original one applying the typical genetic operators (selection, crossover, and mutation); then, the individuals in the two populations are sorted according to their rank, and the best solutions are chosen to create a new population. In the case of having to select some individuals with the same rank, a density estimation based on measuring the crowding distance to the surrounding individuals belonging to the same rank is used to get the most promising solutions.

  We have used Deb's NSGA-II implementation [2]. We have used the real-coded version and the parameter settings used in [1]. A crossover probability of $p_c = 0.9$ and a mutation probability $p_m = 1/n$ (where n is the number of decision variables) are used. The operators for crossover and mutation are SBX and polynomial mutation, with distribution indexes of $\eta_c = 20$ and $\eta_m = 20$, respectively. The population size is 100 individuals.
- **Strength Pareto Evolutionary Algorithm**: SPEA2 was proposed by Zitler *et al.* in [2]. In this algorithm, each individual has assigned a fitness value that is the sum of its strength raw fitness and a density estimation (see Section 3.2.3). The algorithm applies the selection, crossover, and mutation operators to fill an archive of individuals; then, the nondominated individuals of both the original population and the archive are copied into a new population. If the number of nondominated individuals is greater than

[2] The implementation of NSGA-II is available for downloading at: `http://www.iitk.ac.in/kangal/soft.htm`

the population size, a truncator operator based on calculating the distances to the k-th nearest neighbor is used. This way, the individuals having the minimum distance to any other individual are chosen.

We have used the authors' implementation of SPEA2[3]. The algorithm is implemented within the framework PISA [23]. However, the implementation of SPEA2 does not contain a constraint-handling management, so we forced to modify the original implementation to include the same constraint mechanism used in NSGA-II and AbYSS. We have used the following values for the parameters. The population and the archive have a size of 100 individuals, and the crossover and mutation operators are the same used in NSGA-II, using the same values concerning their application probabilities and distribution indexes.

The parameters characterizing AbYSS were discussed in the previous section. AbYSS is written in C++, and its source code is available at the following Web address: `http://neo.lcc.uma.es/Software/deme/html/index.html`.

5.1 Test Problems

In this section we describe first the three different sets of both constrained and unconstrained problems, which have been used in past studies in this area. From them, we have selected first the following bi-objective unconstrained problems: Schaffer [24], Fonseca [25], and Kursawe [26], as well as the problems ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6, which are defined in [21]. The formulation of these problems is provided in Table 8 (see Appendix B for the tables describing the problems), which also shows the number of variables, their bounds, and the nature of the Pareto-optimal front for each problem.

The second set is composed of the following constrained bi-objective problems: Osyczka2 [27], Tanaka [28], Srinivas [13], Constr_Ex [1], and Golinski [29]. Their formulation is presented in Table 9, which includes also the constraints, the number of variables, and the variable bounds.

Finally, we have included four problems of more than two objectives. The problems are Viennet2, Viennet3, and Viennet4 [30], and Water [31]. The two first problems have three objectives and zero contraints, the third one has three objectives and three constrains, and the last one has five objectives and seven constraints. Their formulation is included in Table 10.

[3] The implementation of SPEA2 is available at: `http://www.tik.ee.ethz.ch/pisa/selectors/spea2/spea2.html`

To evaluate each algorithm, we have made two series of experiments. We have run first all the approaches for a number of 25,000 function evaluations, and we have repeated them again using as stopping condition to carry out 50,000 function evaluations. For each problem, we have executed 100 independent runs. The spread metric is not applicable to problems with more than two objectives; for this reason, the problems Viennet2, Viennet3, Viennet4, and Water do not appear in the tables contaning the results of this metric.

In all the tables we include in the results the mean, $\bar{x}$, and standard deviation, σ_n; the best results are marked in boldface. Furthermore, we have applied an Anova test with a 5% of significance level (marked as "+" in tables). This way, a "+" at the end of a row indicates that the values have a statistical confidence in the sense that the differences are unlikely to have occurred by chance with a probability of 95%.

Tables 2, 3, and 4 show the results of the previously described metrics using the algorithms AbYSS, NSGA-II, and SPEA2 when performing 25,000 function evaluations.

Table 2
Mean and standard deviation of the generational distance metric (25,000 function evaluations).

	AbYSS	NSGA-II	SPEA2	
Problem	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	$\bar{x}_{\sigma_n}$	A
Schaffer	$0.00023_{\pm 1.3e-05}$	$\mathbf{0.00023}_{\pm 1.2e-05}$	$0.00024_{\pm 1.1e-05}$	+
Fonseca	$\mathbf{0.00021}_{\pm 2.2e-05}$	$0.00047_{\pm 3.9e-05}$	$0.00023_{\pm 2.4e-05}$	+
Kursawe	$\mathbf{0.00015}_{\pm 1.5e-05}$	$0.00021_{\pm 2.2e-05}$	$0.00016_{\pm 1.5e-05}$	+
Zdt1	$\mathbf{0.00012}_{\pm 2.5e-05}$	$0.00022_{\pm 3.6e-05}$	$0.0002_{\pm 1.3e-05}$	+
Zdt2	$\mathbf{4.7e\text{-}05}_{\pm 2.4e-06}$	$0.00017_{\pm 3.8e-05}$	$0.00011_{\pm 5.4e-05}$	+
Zdt3	$\mathbf{0.00019}_{\pm 1.1e-05}$	$0.00022_{\pm 3.7e-05}$	$0.00023_{\pm 1.3e-05}$	+
Zdt4	$0.021_{\pm 0.089}$	$\mathbf{0.00049}_{\pm 0.00026}$	$0.062_{\pm 0.039}$	+
Zdt6	$\mathbf{0.00072}_{\pm 0.00012}$	$0.001_{\pm 8.7e-05}$	$0.00083_{\pm 5.1e-05}$	+
ConstrEx	$0.00022_{\pm 2.5e-05}$	$0.00029_{\pm 3.2e-05}$	$\mathbf{0.00021}_{\pm 1.8e-05}$	+
Srinivas	$\mathbf{7.6e\text{-}05}_{\pm 4.1e-05}$	$0.00019_{\pm 3e-05}$	$0.00011_{\pm 2e-05}$	+
Osyczka2	$0.0063_{\pm 0.011}$	$\mathbf{0.0011}_{\pm 0.00013}$	$0.0061_{\pm 0.011}$	+
Golinski	$0.0017_{\pm 0.0067}$	$0.00032_{\pm 2.2e-05}$	$\mathbf{0.00028}_{\pm 8.6e-05}$	+
Tanaka	$0.00075_{\pm 7.4e-05}$	$0.0012_{\pm 8e-05}$	$\mathbf{0.00072}_{\pm 7.1e-05}$	+
Viennet2	$\mathbf{0.00077}_{\pm 0.00029}$	$0.00085_{\pm 0.00037}$	$0.00087_{\pm 0.00018}$	+
Viennet3	$\mathbf{0.00011}_{\pm 4.2e-05}$	$0.00023_{\pm 6e-05}$	$0.00032_{\pm 0.00014}$	+
Viennet4	$\mathbf{0.00027}_{\pm 6.8e-05}$	$0.00046_{\pm 0.00013}$	$0.00058_{\pm 0.00016}$	+
Water	$0.0065_{\pm 0.0044}$	$\mathbf{0.0064}_{\pm 0.0011}$	$0.017_{\pm 0.0016}$	+

The generational metric indicates that AbYSS obtains the best results in ten out of the seventeen test problems. It is interesting to note that on the second

Table 3
Mean and standard deviation of the Spread metric (25,000 function evaluations).

Problem	AbYSS $\bar{x}_{\sigma_n}$	NSGA-II $\bar{x}_{\sigma_n}$	SPEA2 $\bar{x}_{\sigma_n}$	A
Schaffer	$0.1545_{\pm 0.014}$	$0.4448_{\pm 0.036}$	$\mathbf{0.1469}_{\pm 0.011}$	+
Fonseca	$\mathbf{0.1174}_{\pm 0.012}$	$0.3596_{\pm 0.028}$	$0.1445_{\pm 0.013}$	+
Kursawe	$\mathbf{0.4137}_{\pm 0.0059}$	$0.546_{\pm 0.024}$	$0.439_{\pm 0.0089}$	+
Zdt1	$\mathbf{0.1107}_{\pm 0.01}$	$0.3645_{\pm 0.029}$	$0.1684_{\pm 0.013}$	+
Zdt2	$\mathbf{0.1125}_{\pm 0.012}$	$0.3644_{\pm 0.03}$	$0.1403_{\pm 0.067}$	+
Zdt3	$\mathbf{0.7007}_{\pm 0.037}$	$0.7416_{\pm 0.023}$	$0.704_{\pm 0.018}$	+
Zdt4	$0.3143_{\pm 0.18}$	$0.3651_{\pm 0.033}$	$\mathbf{0.1049}_{\pm 0.17}$	+
Zdt6	$\mathbf{0.1466}_{\pm 0.017}$	$0.2988_{\pm 0.025}$	$0.1728_{\pm 0.012}$	+
ConstrEx	$\mathbf{0.1728}_{\pm 0.014}$	$0.4212_{\pm 0.035}$	$0.5204_{\pm 0.016}$	+
Srinivas	$\mathbf{0.08339}_{\pm 0.01}$	$0.368_{\pm 0.03}$	$0.1628_{\pm 0.013}$	+
Osyczka2	$0.3269_{\pm 0.15}$	$0.4603_{\pm 0.056}$	$\mathbf{0.3145}_{\pm 0.14}$	+
Golinski	$\mathbf{0.213}_{\pm 0.12}$	$0.414_{\pm 0.035}$	$0.4871_{\pm 0.26}$	+
Tanaka	$0.7313_{\pm 0.033}$	$0.7154_{\pm 0.024}$	$\mathbf{0.6655}_{\pm 0.027}$	+

Table 4
Mean and standard deviation of the hypervolume metric (25,000 function evaluations).

Problem	AbYSS $\bar{x}_{\sigma_n}$	NSGA-II $\bar{x}_{\sigma_n}$	SPEA2 $\bar{x}_{\sigma_n}$	A
Schaffer	$\mathbf{0.8298}_{\pm 4.2e-05}$	$0.8287_{\pm 0.00024}$	$0.8296_{\pm 6e-05}$	+
Fonseca	$\mathbf{0.3106}_{\pm 0.00026}$	$0.3064_{\pm 0.00047}$	$0.3105_{\pm 0.00024}$	+
Kursawe	$0.4009_{\pm 0.00021}$	$0.3997_{\pm 0.00023}$	$\mathbf{0.4009}_{\pm 0.00014}$	+
Zdt1	$\mathbf{0.6619}_{\pm 2.1e-05}$	$0.6594_{\pm 0.00028}$	$0.66_{\pm 0.00025}$	+
Zdt2	$\mathbf{0.3287}_{\pm 2.6e-05}$	$0.3262_{\pm 0.00032}$	$0.268_{\pm 0.13}$	+
Zdt3	$\mathbf{0.5154}_{\pm 0.0052}$	$0.5148_{\pm 0.00037}$	$0.5139_{\pm 0.00041}$	+
Zdt4	$0.5934_{\pm 0.15}$	$\mathbf{0.6546}_{\pm 0.004}$	$0.1483_{\pm 0.15}$	+
Zdt6	$\mathbf{0.3957}_{\pm 0.0017}$	$0.386_{\pm 0.0013}$	$0.3925_{\pm 0.00088}$	+
ConstrEx	$\mathbf{0.776}_{\pm 0.0002}$	$0.7745_{\pm 0.00032}$	$0.7751_{\pm 0.00033}$	+
Srinivas	$\mathbf{0.5406}_{\pm 9.5e-05}$	$0.5382_{\pm 0.00037}$	$0.54_{\pm 0.00015}$	+
Osyczka2	$0.694_{\pm 0.12}$	$\mathbf{0.7464}_{\pm 0.0083}$	$0.6775_{\pm 0.12}$	+
Golinski	$\mathbf{0.9702}_{\pm 0.043}$	$0.969_{\pm 0.00017}$	$0.9659_{\pm 0.0057}$	-
Tanaka	$0.3071_{\pm 0.00036}$	$0.3075_{\pm 0.00037}$	$\mathbf{0.3088}_{\pm 0.00026}$	+
Viennet2	$0.9221_{\pm 0.00095}$	$0.9204_{\pm 0.0014}$	$\mathbf{0.9252}_{\pm 0.00038}$	+
Viennet3	$\mathbf{0.836}_{\pm 0.00035}$	$0.8337_{\pm 0.00054}$	$0.8266_{\pm 0.0013}$	+
Viennet4	$0.8595_{\pm 0.0015}$	$0.8571_{\pm 0.002}$	$\mathbf{0.8637}_{\pm 0.00078}$	+
Water	$\mathbf{0.4235}_{\pm 0.0044}$	$0.4087_{\pm 0.0057}$	$0.4029_{\pm 0.0056}$	+

group of problems (biobjective and constrained) our extended implementation of SPEA2 achieves the best results in three out of the five problems in that group, what points out the accurate implementation we developed.

The results obtained from the Spread metric (see Table 3) indicates that AbYSS also outperforms the other two algorithms concerning the diversity

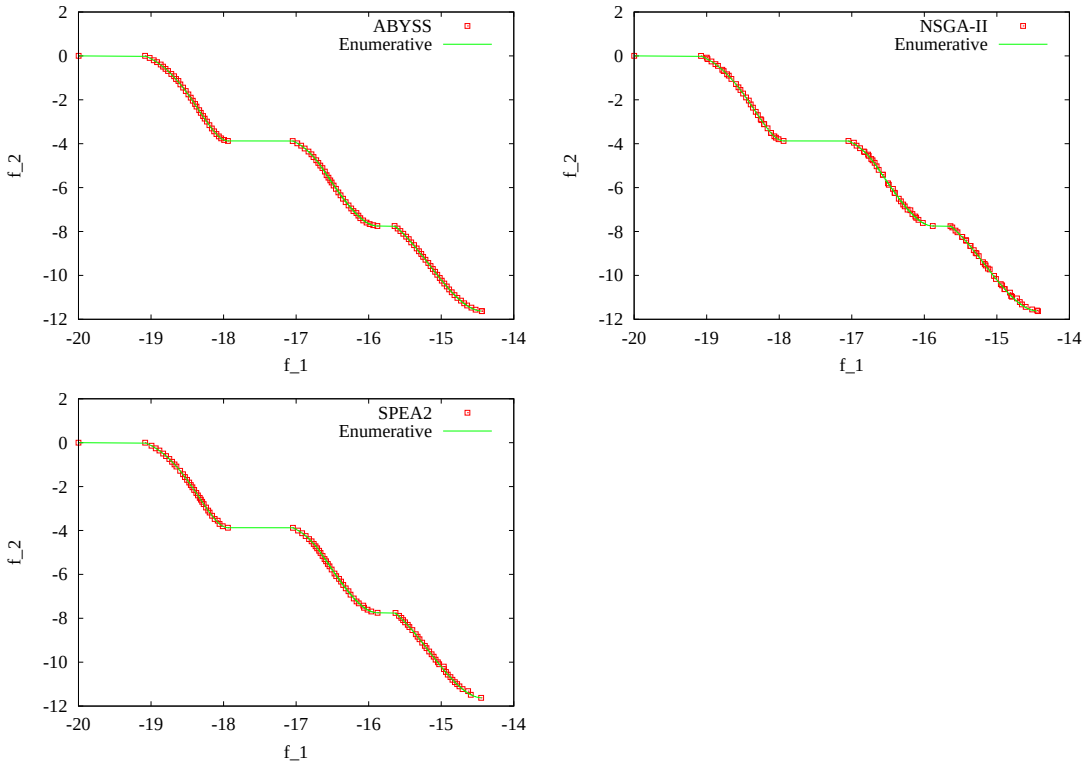

Fig. 7. AbYSS and SPEA2 find better spread of solutions than NSGA-II on problem Kursawe

of the obtained Pareto fronts. It yields the best values in nine out of the thirteen problems. To illustrate this fact, we show typical simulation results of the three algorithms when solving the problems Kursawe and ConstrEx. In Fig. 7 we include one of the 100 fronts produced by each technique when solving the first problem. We can observe that the nondominated set of solutions generated by AbYSS achieves an almost perfect spread out; only SPEA2 is able to produce a similar Pareto front. Fig. 8 shows the fronts obtained by the three algorithms when solving the second problem; again AbYSS produces the best quality front.

The hypervolume metric confirms the results of the two other metrics; thus, we observe that AbYSS obtains the best values in eleven problems. NSGA-II obtains the best results in the problems ZDT4 and Osyczka2, with significant differences when compared to AbYSS and SPEA2.

We now turn to analyze the results obtained when running 50,000 function evaluations, which are included in Tables 5, 6, and 7. The ranking of the algorithms yielding better generational distance is maintained compared with the previous experiment. It is interesting to note that AbYSS significatively enhances the generational distance values when solving the problems ZDT4, Golinski, and Viennet3. This indicates that our algorithm encounters some difficulties to converge to the true Pareto front when trying to solve some

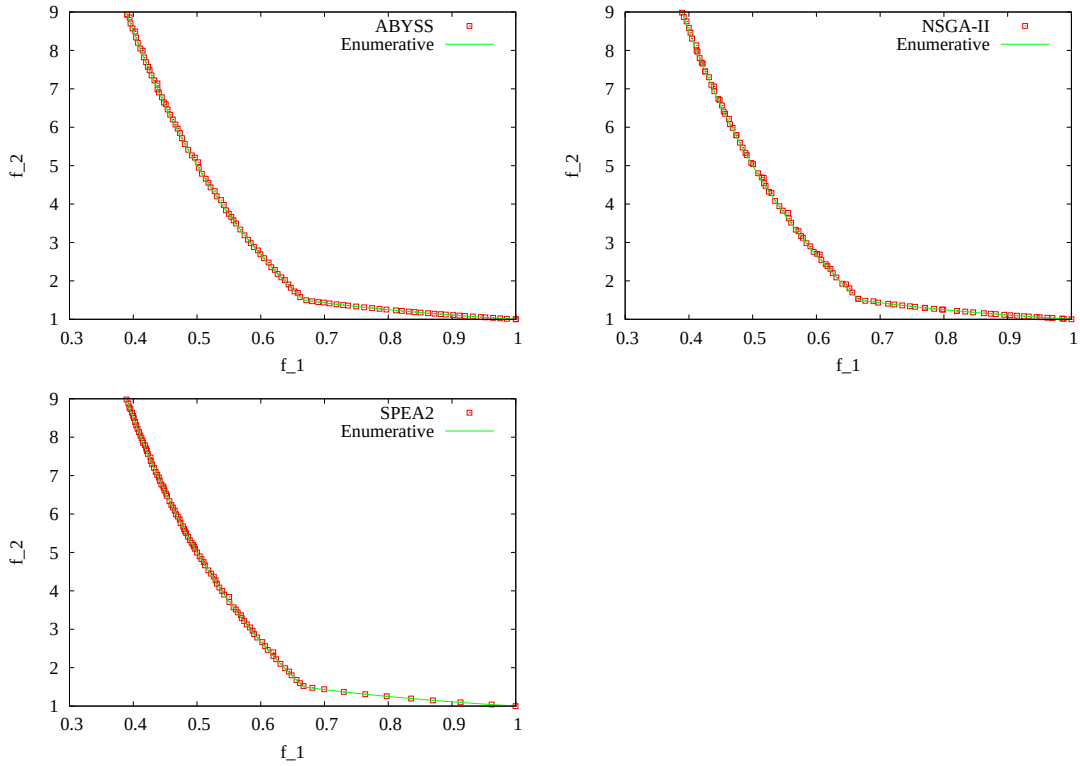

Fig. 8. AbYSS obtains a better spread of solutions than SPEA2 and NSGA-II on problem ConstrEx

kind of problems, and a larger number of steps are required to achieve better results in these situations. The spread (Table 6) and hypervolume (Table 7) confirm this claim, and AbYSS gets the best values in eleven out of the thirteen problems and eleven out of the seventeen problems, respectively. Finally, it is worth mentioning that some algorithms produce worse results when computing more function evaluations (e.g., AbYSS on Srinivas, Osyczka2, and Tanaka; SPEA2 on ZDT4 and Tanaka).

We also want to remark that, concerning diversity, AbYSS is not only the best of the three analyzed algorithms, but the differences in the spread values are in general noticeable compared to the rest of algorithms.

In order to demonstrate the working principles of AbYSS, we include simulation results of AbYSS on the test problems ZDT1, ZDT2, ZDT4, and ZDT6 (computing 50000 function evaluations). The fronts are included in Fig. 9. The performance of AbYSS on these problems confirms the excellent values of the spread metric, while achieving also a noticeable convergence to the true Pareto front.

Table 5
Mean and standard deviation of the generational distance metric (50,000 function evaluations).

Problem	AbYSS $\bar{x}_{\sigma_n}$	NSGA-II $\bar{x}_{\sigma_n}$	SPEA2 $\bar{x}_{\sigma_n}$	A
Schaffer	$0.00023_{\pm 1.4e-05}$	$\mathbf{0.00023}_{\pm 1.2e-05}$	$0.00024_{\pm 1.1e-05}$	+
Fonseca	$\mathbf{0.00018}_{\pm 2.1e-05}$	$0.00046_{\pm 3.9e-05}$	$0.00022_{\pm 2.2e-05}$	+
Kursawe	$\mathbf{0.00014}_{\pm 9.8e-06}$	$0.00021_{\pm 2.4e-05}$	$0.00016_{\pm 1.5e-05}$	+
Zdt1	$\mathbf{0.00012}_{\pm 2.5e-05}$	$0.00019_{\pm 3.8e-05}$	$0.00016_{\pm 1.9e-05}$	+
Zdt2	$4.6e\text{-}05_{\pm 2.4e-06}$	$0.00013_{\pm 4.3e-05}$	$\mathbf{4e\text{-}05}_{\pm 2e-05}$	+
Zdt3	$\mathbf{0.00019}_{\pm 8.2e-06}$	$0.00021_{\pm 1.3e-05}$	$0.00021_{\pm 1.3e-05}$	+
Zdt4	$0.00049_{\pm 0.00023}$	$\mathbf{0.00019}_{\pm 7.2e-05}$	$0.065_{\pm 0.037}$	+
Zdt6	$\mathbf{0.00054}_{\pm 1.4e-05}$	$0.00056_{\pm 3.4e-05}$	$0.00055_{\pm 1.7e-05}$	+
ConstrEx	$\mathbf{0.00017}_{\pm 1.9e-05}$	$0.00029_{\pm 3.4e-05}$	$0.0002_{\pm 2.3e-05}$	+
Srinivas	$\mathbf{5.8e\text{-}05}_{\pm 2.6e-05}$	$0.00019_{\pm 3.5e-05}$	$0.00012_{\pm 2.2e-05}$	+
Osyczka2	$0.0019_{\pm 0.004}$	$\mathbf{0.0016}_{\pm 0.0033}$	$0.0067_{\pm 0.012}$	+
Golinski	$0.00085_{\pm 0.00083}$	$0.00033_{\pm 2.2e-05}$	$\mathbf{0.00027}_{\pm 8.6e-05}$	+
Tanaka	$0.0012_{\pm 0.0039}$	$\mathbf{0.0012}_{\pm 6.9e-05}$	$0.00078_{\pm 5.2e-05}$	-
Viennet2	$\mathbf{0.00083}_{\pm 0.00027}$	$0.00084_{\pm 0.00035}$	$0.00087_{\pm 0.00014}$	-
Viennet3	$\mathbf{8.6e\text{-}05}_{\pm 2.8e-05}$	$0.00022_{\pm 6.3e-05}$	$0.0028_{\pm 0.022}$	-
Viennet4	$\mathbf{0.00026}_{\pm 7.3e-05}$	$0.00049_{\pm 0.00017}$	$0.00051_{\pm 0.00014}$	+
Water	$\mathbf{0.006}_{\pm 0.0023}$	$0.0062_{\pm 0.00088}$	$0.017_{\pm 0.0019}$	+

Table 6
Mean and standard deviation of the Spread metric (50,000 function evaluations).

Problem	AbYSS $\bar{x}_{\sigma_n}$	NSGA-II $\bar{x}_{\sigma_n}$	SPEA2 $\bar{x}_{\sigma_n}$	A
Schaffer	$\mathbf{0.1328}_{\pm 0.017}$	$0.4466_{\pm 0.037}$	$0.1472_{\pm 0.012}$	+
Fonseca	$\mathbf{0.105}_{\pm 0.011}$	$0.3631_{\pm 0.029}$	$0.1467_{\pm 0.016}$	+
Kursawe	$\mathbf{0.4109}_{\pm 0.0035}$	$0.5499_{\pm 0.022}$	$0.4368_{\pm 0.0091}$	+
Zdt1	$\mathbf{0.1028}_{\pm 0.013}$	$0.3612_{\pm 0.033}$	$0.1676_{\pm 0.012}$	+
Zdt2	$\mathbf{0.1005}_{\pm 0.011}$	$0.3758_{\pm 0.032}$	$0.1378_{\pm 0.068}$	+
Zdt3	$\mathbf{0.6975}_{\pm 0.03}$	$0.743_{\pm 0.014}$	$0.706_{\pm 0.0043}$	+
Zdt4	$0.1296_{\pm 0.019}$	$0.3885_{\pm 0.033}$	$\mathbf{0.1186}_{\pm 0.22}$	+
Zdt6	$\mathbf{0.09334}_{\pm 0.0096}$	$0.4324_{\pm 0.033}$	$0.1714_{\pm 0.012}$	+
ConstrEx	$\mathbf{0.1516}_{\pm 0.01}$	$0.4274_{\pm 0.029}$	$0.5187_{\pm 0.016}$	+
Srinivas	$\mathbf{0.07583}_{\pm 0.0094}$	$0.3615_{\pm 0.035}$	$0.1628_{\pm 0.012}$	+
Osyczka2	$\mathbf{0.2402}_{\pm 0.065}$	$0.4501_{\pm 0.054}$	$0.3098_{\pm 0.13}$	+
Golinski	$\mathbf{0.1515}_{\pm 0.069}$	$0.4216_{\pm 0.036}$	$0.5231_{\pm 0.25}$	+
Tanaka	$0.6474_{\pm 0.029}$	$0.6998_{\pm 0.023}$	$\mathbf{0.6225}_{\pm 0.02}$	+

6 Conclusions and Future Work

We have presented a proposal to adapt the scatter search method to handle multiobjective optimization problems. The proposed algorithm, AbYSS, is an hybridized scatter search to the multiobjective field, and it uses an external

Table 7
Mean and standard deviation of the hypervolume metric (50,000 function evaluations).

Problem	AbYSS $\bar{x}_{\sigma_n}$	NSGA-II $\bar{x}_{\sigma_n}$	SPEA2 $\bar{x}_{\sigma_n}$	A
Schaffer	**0.8298**$_{\pm 4e-05}$	0.8287$_{\pm 0.00027}$	0.8296$_{\pm 6.3e-05}$	+
Fonseca	**0.311**$_{\pm 0.00023}$	0.3064$_{\pm 0.0005}$	0.3106$_{\pm 0.00024}$	+
Kursawe	**0.4013**$_{\pm 0.00013}$	0.3998$_{\pm 0.00023}$	0.4009$_{\pm 0.00016}$	+
Zdt1	**0.662**$_{\pm 2.2e-05}$	0.6604$_{\pm 0.00025}$	0.6615$_{\pm 5.5e-05}$	+
Zdt2	**0.3287**$_{\pm 2.1e-05}$	0.3273$_{\pm 0.00023}$	0.2659$_{\pm 0.13}$	+
Zdt3	**0.5159**$_{\pm 0.0006}$	0.5154$_{\pm 9.8e-05}$	0.5156$_{\pm 7.2e-05}$	+
Zdt4	0.6561$_{\pm 0.0032}$	**0.6592**$_{\pm 0.0013}$	0.1366$_{\pm 0.15}$	+
Zdt6	**0.4003**$_{\pm 0.00012}$	0.3943$_{\pm 0.00025}$	0.3994$_{\pm 0.0002}$	+
ConstrEx	**0.7764**$_{\pm 0.00017}$	0.7745$_{\pm 0.00029}$	0.7752$_{\pm 0.00034}$	+
Srinivas	**0.5407**$_{\pm 7.2e-05}$	0.5383$_{\pm 0.00032}$	0.54$_{\pm 0.00015}$	+
Osyczka2	**0.7452**$_{\pm 0.039}$	0.7387$_{\pm 0.06}$	0.6774$_{\pm 0.13}$	+
Golinski	0.9687$_{\pm 0.088}$	**0.9691**$_{\pm 0.00018}$	0.9671$_{\pm 0.00045}$	-
Tanaka	0.3061$_{\pm 0.02}$	0.3078$_{\pm 0.00024}$	**0.3092**$_{\pm 0.00015}$	-
Viennet2	0.922$_{\pm 0.0008}$	0.9203$_{\pm 0.0015}$	**0.9252**$_{\pm 0.00046}$	+
Viennet3	**0.8361**$_{\pm 0.00028}$	0.8336$_{\pm 0.00061}$	0.8121$_{\pm 0.1}$	+
Viennet4	0.8595$_{\pm 0.0016}$	0.8575$_{\pm 0.0021}$	**0.8638**$_{\pm 0.00094}$	+
Water	**0.4251**$_{\pm 0.0029}$	0.4103$_{\pm 0.0057}$	0.4033$_{\pm 0.0055}$	+

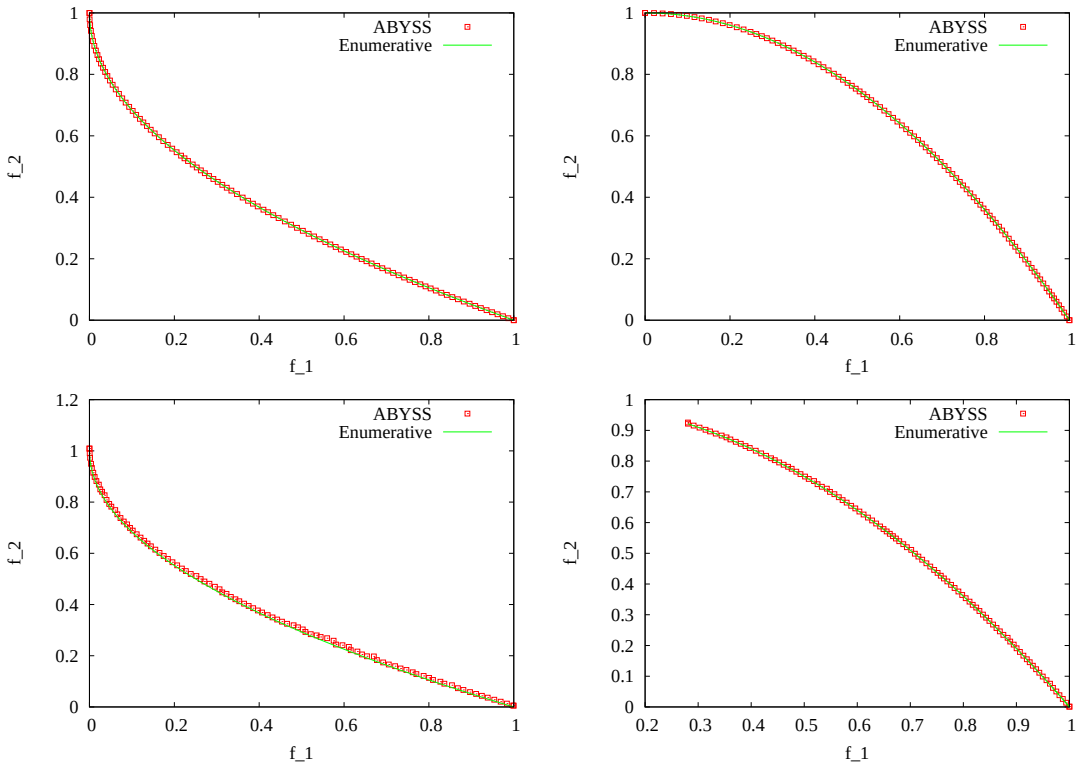

Fig. 9. Nondominated solutions with AbYSS on problems ZDT1 (top left), ZDT2 (top right), ZDT4 (bottom left), and ZDT6 (bottom right)

archive to store the nondominated individuals found during the search. Salient features of AbYSS are the feedback of individuals from the archive to the initial set in the re-start phase of the scatter search, as well as the combination of two different density estimators in different parts of the search. On the one hand, the crowding distance, taken from NSGA-II, is applied to remove individuals from the archive when it becomes full and to choose the best individuals which are removed from the archive to feed the initial set in the re-start; on the other hand, the density estimator used in SPEA2 is considered to obtain the best individuals from the initial set to create the reference set.

AbYSS was validated using a standard methodology which is currently used in the evolutionary multiobjective optimization community. The algorithm was compared against two state-of-the-art multiobjective optimization algorithms; for that purpose, seventeen test problems, including unconstrained and constrained ones, were chosen and three metrics were used to assess the performance of the algorithms. The results of the three metrics reveal that AbYSS outperforms all the proposals on the considered test problems: running the algorithms for 50000 function evaluations, AbYSS achieves the best results in eleven and twelve out of the seventeen test problems according to the generational and hypervolume metrics, respectively, and in eleven out of the thirteen problems considered according to the spread metric.

The evaluation of AbYSS with other benchmarks and its application to solve real-world problems are matter of future work. It is also desirable to make a deep study of the parameters defining the behavior of the algorithm.

References

[1] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[2] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001).

[3] J. Knowles, D. Corne, The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, 1999, pp. 9–105.

[4] C. Coello, D. Van Veldhuizen, G. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Genetic Algorithms and Evolutionary Computation, Kluwer Academic Publishers, 2002.

[5] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, 2001.

[6] F. Glover, A template for scatter search and path relinking, Lecture Notes in Computer Science, Springer Verlag, 1997.

[7] F. Glover, M. Laguna, R. Martí, Fundamentals of scatter search and path relinking, Control and Cybernetics 29 (3) (2000) 653–684.

[8] F. Glover, M. Laguna, R. Martí, Scatter search, in: A. Ghosh, S. Tsutsui (Eds.), Advances in Evolutionary Computing: Theory and Applications, Springer, 2003.

[9] R. P. Beausoleil, MOSS: Multiobjective scatter search applied to nonlinear multiple criteria optimization, to appear in the European Journal of Operational Research (2005).

[10] J. Molina, M. Laguna, R. Martí, R. Caballero, SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization, to be published in INFORMS Journal on Computing (2005).

[11] A. J. Nebro, F. Luna, E. Alba, New ideas in applying scatter search to multiobjective optimization, in: C. Coello, A. Hernández, E. Zitler (Eds.), Third International Conference on Evolutionary MultiCriterion Optimization, EMO 2005, Vol. 3410 of Lecture Notes in Computer Science, Springer, 2005, pp. 443–458.

[12] J. A. Vasconcelos, J. H. R. D. Maciel, R. O. Parreiras, Scatter Search Techniques Applied to Electromagnetic Problems, IEEE Transactions on Magnetics 4 (5) (2005) 1804–1807.

[13] N. Srinivas, K. Deb, Multiobjective function optimization using nondominated sorting genetic algorithms, Evolutionary Computation 2 (3) (1995) 221–248.

[14] C. G. da Silva, J. Clímaco, J. Figueira, A scatter search method for the bi-criteria multi-dimensional {0,1}-knapsack problem using surrogate relaxation, Journal of Mathematical Modelling and Algorithms 3 (3) (2004) 183–208.

[15] R. Martí, H. Loureno, M. Laguna, Computing tools for modeling, optimization and simulation: Interfaces in computer science and operations research, Kluwer Academic Publishers, 2000, Ch. Assigning Proctors to Exams with Scatter Search, pp. 215–227.

[16] A. Corberán, E. Fernández, M. Laguna, R. Martí, Heuristic Solutions to the Problem of Routing School Buses with Multiple Objectives, Journal of the Operational Research Society 53 (4) (2002) 427–435.

[17] A. Rama Mohan Rao, N. Arvind, A Scatter Search Algorithm for Stacking Sequence Optimisation of Laminate Composites, Composite Structures (2005) To appear.

[18] C. Coello, G. Toscano, M. Salazar, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 256–278.

[19] D. A. Van Veldhuizen, G. B. Lamont, Multiobjective Evolutionary Algorithm Research: A History and Analysis, Tech. Rep. TR-98-03, Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., Wright-Patterson, AFB, OH (1998).

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

[21] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation 8 (2) (2000) 173–195.

[22] F. Herrera, M. Lozano, D. Molina, Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies, European Journal of Operational Research 169 (2006) 450–476.

[23] S. Bleuler, M. Laumanns, L. Thiele, E. Zitzler, PISA - A Platform and Programming Language Independent Interface for Search Algorithms, in: Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), 2003, pp. 494–508.

[24] J. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: J. Grefensttete (Ed.), First International Conference on Genetic Algorithms, Hillsdale, NJ, 1987, pp. 93–100.

[25] C. Fonseca, P. Flemming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms - part II: Application example, IEEE Transactions on System, Man, and Cybernetics 28 (1998) 38–47.

[26] F. Kursawe, A variant of evolution strategies for vector optimization, in: H. Schwefel, R. Männer (Eds.), Parallel Problem Solving for Nature, Springer-Verlag, Berlin, Germany, 1990, pp. 193–197.

[27] A. Osyczka, S. Kundo, A new method to solve generalized multicriteria optimization problems using a simple genetic algorithm, Structural Optimization 10 (1995) 94–99.

[28] M. Tanaka, H. Watanabe, Y. Furukawa, T. Tanino, Ga-based decision support system for multicriteria optimization, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2, 1995, pp. 1556–1561.

[29] A. Kurpati, S. Azarm, J. Wu, Constraint handling improvements for multi-objective genetic algorithms, Structural and Multidisciplinary Optimization 23 (3) (2002) 204–213.

[30] R. Viennet, C. Fontiex, I. Marc, Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set, Journal of Systems Science 27 (2) (1996) 255–260.

[31] T. Ray, K. Tai, K. Seow, An Evolutionary Algorithm for Multiobjective Optimization, Engineering Optimization 33 (3) (2001) 399–424.

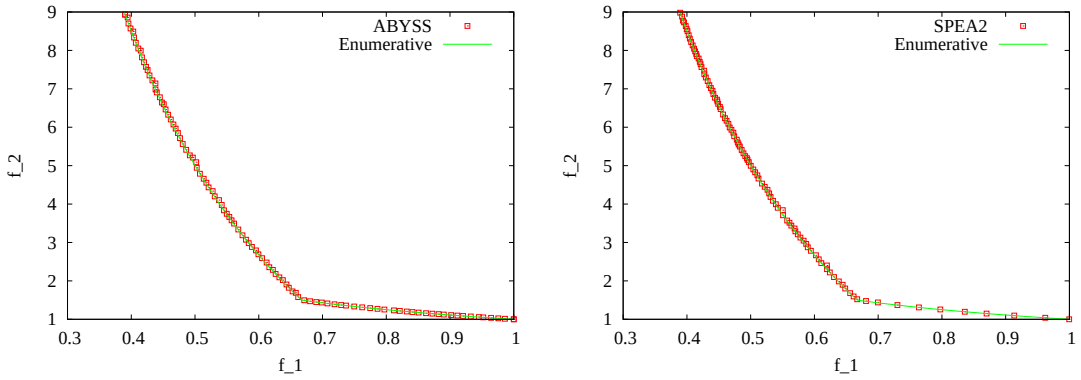

Fig. 10. Pareto fronts of the problem ConstrEx with AbYSS (left) and SPEA2 (right)

Appendix A: About the Spread Metric

The Spread metric measures the extent of spread achieved among the solutions obtained by a multiobjective optimization algorithm. However, as defined in [1], the metric can give confusing results if we compare two fronts and the two objective functions range between values of different order of magnitude. We observe this behaviour when comparing the fronts of problem ConstrEx produced by the algorithms AbYSS and SPEA2.

As can be seen in Fig 10, the Pareto front produced by AbYSS achieves a better spread that the one obtained by SPEA2; after applying the Spread metric, the values reported are 0.5085 and 0.2024, respectively. Thus, according to the metric, SPEA2 is more than twice times better than AbYSS on this problem.

If we observe the Pareto front of problem ConstrEx, we can see that it is composed of two parts. The left part ranges roughly between 0.4 and 0.65 in the x-axis and 1.5 and 9 in the y-axis, while the right part ranges between 0.65 and 1 (x-axis) and 1 and 1.5 (y-axis). A closer look to the Pareto fronts reveals that SPEA2 produces more solutions in the left part, while the solutions obtained by AbYSS are uniformly spreaded among the two parts. As both set of solutions are composed of the same number of points (100 solutions), the Spread metric favours SPEA2 because the distances measured in the right front are neglibible compared to those of the left front.

To solve this issue, we take the approach of normalizing the values of the two objective functions between 0 and 1. This way, the shape of the Pareto fronts are kept identical, and the results of applying the Spread metric yield 0.15 to AbYSS and 0.51 to SPEA2.

Appendix B: Formulation of the Benchmark Problems

Table 8
Unconstrained bi-objective problems.

Problem	Objective functions	n	Variable bounds	Comments
Schaffer	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	1	$-10^5 \leq x \leq 10^5$	*convex*
Fonseca	$f_1(\vec{x}) = 1 - e^{-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2}$ $f_2(\vec{x}) = 1 - e^{-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2}$	3	$-4 \leq x_i \leq 4$	*nonconvex*
Kursawe	$f_1(\vec{x}) = \sum_{i=1}^{n-1}\left(-10e^{\left(-0.2*\sqrt{x_i^2+x_{i+1}^2}\right)}\right)$ $f_2(\vec{x}) = \sum_{i=1}^{n}(\lvert x_i\rvert^a + 5\sin(x_i)^b)$	3	$-5 \leq x_i \leq 5$	*nonconvex*
ZDT1	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})[1 - \sqrt{x_1/g(\vec{x})}]$ $g(\vec{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	30	$0 \leq x_i \leq 1$	*convex*
ZDT2	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})\left[1 - (x_1/g(\vec{x}))^2\right]$ $g(\vec{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	30	$0 \leq x_i \leq 1$	*nonconvex*
ZDT3	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})\left[1 - \sqrt{\frac{x_1}{g(\vec{x})}} - \frac{x_1}{g(\vec{x})}\sin(10\pi x_1)\right]$ $g(\vec{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	30	$0 \leq x_i \leq 1$	*convex* *disconnected*
ZDT4	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})[1 - (x_1/g(\vec{x}))^2]$ $g(\vec{x}) = 1 + 10(n-1) +$ $\sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)])$	10	$0 \leq x_1 \leq 1$ $-5 \leq x_i \leq 5$ $i = 2, ..., n$	*nonconvex*
ZDT6	$f_1(\vec{x}) = 1 - e^{-4x_1}\sin^6(6\pi x_1)$ $f_2(\vec{x}) = g(\vec{x})[1 - (f_1(\vec{x})/g(\vec{x}))^2]$ $g(\vec{x}) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$	10	$0 \leq x_i \leq 1$	*nonconvex* *nonunformly* *spaced*

Table 9
Constrained test bi-objective problems.

Problem	Objective functions	Constraints	n	Variable bounds
Osyczka2	$f_1(\vec{x}) = -(25(x_1-2)^2+$ $(x_2-2)^2+$ $(x_3-1)^2(x_4-4)^2+$ $(x_5-1)^2)$ $f_2(\vec{x}) = x_1^2+x_2^2+$ $x_3^2+x_4^2+x_5^2+x_6^2$	$g_1(\vec{x}) = 0 \leq x_1+x_2-2$ $g_2(\vec{x}) = 0 \leq 6-x_1-x_2$ $g_3(\vec{x}) = 0 \leq 2-x_2+x_1$ $g_4(\vec{x}) = 0 \leq 2-x_1+3x_2$ $g_5(\vec{x}) = 0 \leq 4-(x_3-3)^2-x_4$ $g_6(\vec{x}) = 0 \leq (x_5-3)^3+x_6-4$	6	$0 \leq x_1, x_2 \leq 10$ $1 \leq x_3, x_5 \leq 5$ $0 \leq x_4 \leq 6$ $0 \leq x_6 \leq 10$
Tanaka	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = x_2$	$g_1(\vec{x}) = -x_1^2-x_2^2+1+$ $0.1\cos(16\arctan(x_1/x_2)) \leq 0$ $g_2(\vec{x}) = (x_1-0.5)^2+$ $(x_2-0.5)^2 \leq 0.5$	2	$-\pi \leq x_i \leq \pi$
Constr_Ex	$f_1(x) = x_1$ $f_2(x) = (1+x_2)/x_1$	$g_1(\vec{x}) = x_2+9x_1 \geq 6$ $g_2(\vec{x}) = -x_2+9x_1 \geq 1$	2	$0.1 \leq x_1 \leq 1.0$ $0 \leq x_2 \leq 5$
Srinivas	$f_1(\vec{x}) = (x_1-2)^2+$ $(x_2-1)^2+2$ $f_2(\vec{x}) = 9x_1-(x_2-1)^2$	$g_1(\vec{x}) = x_1^2+x_2^2 \leq 225$ $g_2(\vec{x}) = x_1-3x_2 \leq -10$	2	$-20 \leq x_i \leq 20$
Golinski	$f_1(\vec{x}) = 0.7854x_1x_2^2(10x_3^2/3+$ $14.933x_3-43.0934)$ $-1.508x_1(x_6^2+x_7^2)+$ $7.477(x_6^3+x_7^3)$ $+0.7854(x_4x_6^2+x_5x_7^2)$ $f_2(\vec{x}) = \frac{\sqrt{(\frac{745.0x_4}{x_2x_3})^2+1.69*10^7}}{0.1x_6^3}$	$g_1(\vec{x}) = \frac{1.0}{x_1x_2^2x_3}-\frac{1.0}{27.0} \leq 0$ $g_2(\vec{x}) = \frac{1.0}{x_1x_2^2x_3}-\frac{1.0}{27.0} \leq 0$ $g_3(\vec{x}) = \frac{x_4^3}{x_2x_3^2x_6^4}-\frac{1.0}{1.93} \leq 0$ $g_4(\vec{x}) = \frac{x_5^3}{x_2x_3x_7^4}-\frac{1.0}{1.93} \leq 0$ $g_5(\vec{x}) = x_2x_3-40 \leq 0$ $g_6(\vec{x}) = x_1/x_2-12 \leq 0$ $g_7(\vec{x}) = 5-x_1/x_2 \leq 0$ $g_8(\vec{x}) = 1.9-x_4+1.5x_6 \leq 0$ $g_9(\vec{x}) = 1.9-x_5+1.1x_7 \leq 0$ $g_{10}(\vec{x}) = f_2(\vec{x}) \leq 1300$ $a = 745.0x_5/x_2x_3$ $b = 1.575*10^8$ $g_{11}(\vec{x}) = \frac{\sqrt{a^2+b}}{0.1x_7^3} \leq 1100$	7	$2.6 \leq x_1 \leq 3.6$ $0.7 \leq x_2 \leq 0.8$ $17.0 \leq x_3 \leq 28.0$ $7.3 \leq x_4 \leq 8.3$ $7.3 \leq x_5 \leq 8.3$ $2.9 \leq x_6 \leq 3.9$ $5.0 \leq x_7 \leq 5.5$

Table 10
Problems with more than two objectives.

Problem	Objective functions	Constraints	n	Variable bounds
Viennet2	$f_1(\vec{x}) = \frac{(x_1-2)^2}{2}+ \frac{(x_1+1)^2}{13} + 3.0$ $f_2(\vec{x}) = \frac{(x_1+x_2-3)^2}{36}+ \frac{(-x_1+x_2+2)^2}{8} - 17$ $f_3(\vec{x}) = \frac{(x_1+2x_2-1)^2}{175}+ \frac{(2x_2+x_1)^2}{17} - 13$		2	$-4.0 \leq x_i \leq 4.0$
Viennet3	$f_1(\vec{x}) = 0.5x_1^2 + x_2^2 + sin(x_1^2 + x_2^2)$ $f_2(\vec{x}) = \frac{(3x_1-2x_2+4)^2}{8}+ \frac{(x_1-x_2+1)^2}{27} + 15$ $f_3(\vec{x}) = \frac{1}{x_1^2+x_2^2+1} - 1.1exp(-x_1^2 - x_2^2)$		2	$-3.0 \leq x_i \leq 3.0$
Viennet4	$f_1(\vec{x}) = \frac{(x_1-2)^2}{2}+ \frac{(x_2+1)^2}{13} + 3$ $f_2(\vec{x}) = \frac{(x_1+x_2-3)^2}{175}+ \frac{(2x_2-x_1)^2}{17} - 13$ $f_3(\vec{x}) = \frac{(3x_1-2x_2+4)^2}{8}+ \frac{(x_1-x_2+1)^2}{27} + 15$	$g_1(\vec{x}) = -x_2 - 4x_1 + 4 \geq 0$ $g_2(\vec{x}) = x_1 + 1 \geq 0$ $g_3(\vec{x}) = x_2 - x_1 + 2 \geq 0$	2	$-4.0 \leq x_i \leq 4.0$
Water	$f_1(\vec{x}) = 106780.37(x_2 + x_3) + 61704.67$ $f_2(\vec{x}) = 3000x_1$ $f_3(\vec{x}) = \frac{305700*2289x_2}{(0.06*2289)^{0.65}}$ $f_4(\vec{x}) = 250 * 2289x_2 exp(-39.75x_2 + 9.9x_3 + 2.74)$ $f_5(\vec{x}) = 25\frac{1.39}{(x_1x_2)+4940*x_3-80}$	$g_1(\vec{x}) = 1 - \frac{0.00139}{(x_1x_2)} + 4.94x_3 - 0.08 \geq 0$ $g_2(\vec{x}) = 1 - \frac{0.000306}{(x_1x_2)} + 1.082x_3 - 0.0986 \geq 0$ $g_3(\vec{x}) = 5000 - \frac{12.307}{(x_1x_2)} + 4.9408x_3 + 4051.02 \geq 0$ $g_4(\vec{x}) = 16000 - \frac{2.09}{(x_1x_2)} + 8046.33x_3 - 696.71 \geq 0$ $g_5(\vec{x}) = 10000 - \frac{2.138}{(x_1x_2)} + 7883.39x_3 - 705.04 \geq 0$ $g_6(\vec{x}) = 2000 - \frac{0.417}{(x_1x_2)} + 1721.26x_3 - 136.54 \geq 0$ $g_7(\vec{x}) = 550 - \frac{0.164}{(x_1x_2)} + 631.13x_3 - 54.48 \geq 0$	3	$0.01 \leq x_1 \leq 0.45$ $0.01 \leq x_2 \leq 0.10$ $0.01 \leq x_3 \leq 0.10$