AbYSS: Adapting Scatter Search to Multiobjective Optimization

Antonio J. Nebro, Francisco Luna, *Student Member, IEEE*, Enrique Alba, Bernabé Dorronsoro, *Student Member, IEEE*, Juan J. Durillo, *Student Member, IEEE*, and Andreas Beham, *Student Member, IEEE*

***Abstract*—We propose the use of a new algorithm to solve 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 Archive-Based hYbrid Scatter Search (AbYSS), which follows the scatter search structure but uses mutation and crossover operators from 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 with two state-of-the-art multiobjective optimizers, NSGA-II and SPEA2. The results obtained indicate that, according to the benchmark and parameter settings used, AbYSS outperforms the other two algorithms as regards the diversity of the solutions, and it obtains very competitive results according to the convergence to the true Pareto fronts and the hypervolume metric.**

***Index Terms*—Hybrid metaheuristics, multiobjective optimization, scatter search.**

I. Introduction

In the last few years, much attention has been paid to the optimization of problems whose formulation involves optimizing more than one objective function. This interest is mainly motivated by the multiobjective nature of most real-world problems [1], [2]. Generally speaking, multiobjective optimization is not restricted to finding a single solution to a given multiobjective optimization problem (MOP), but rather it finds a set of solutions called *nondominated solutions*. Each one in this set is said to be a *Pareto optimum*, and when they are plotted in the objective space they are collectively known as the *Pareto front*. Obtaining the Pareto front of a given MOP is the main goal of multiobjective optimization. This means that multiobjective optimizers need to explore larger portions of the search space because they search for the Pareto front, i.e., not a single optimum but a set of Pareto optima. Additionally, many real-world MOPs typically need computationally expensive methods for computing the objective functions and constraints.

In this context, deterministic techniques are generally not applicable, which leads us therefore to using stochastic methods [3]. Among these, metaheuristics appear as a family of approximate techniques that are widely used in many fields to solve optimization problems; in particular, the use of evolutionary algorithms (EAs) for solving MOPs has significantly grown in recent years, giving rise to a wide variety of algorithms, such as NSGA-II [4], SPEA2 [5], PAES [6], and many others [1], [2].

Scatter search [7]–[9] is a metaheuristic algorithm that can be considered an EA in the sense that it incorporates the concept of population. However, scatter search usually avoids using many random components, and typical evolutionary operators such as mutation or crossover operators do not fit, theoretically, with the philosophy of this algorithm. The method is based on using a small population, known as the *reference set*, whose individuals are combined to construct new solutions which are generated systematically. Furthermore, these new individuals can be improved by applying a local search method. The reference set is initialized from an initial population composed of diverse solutions, and it is updated with the solutions resulting from the local search improvement. Scatter search has been found to be successful in a wide variety of optimization problems [8], but until recently it had not been extended to deal with MOPs.

Our interest here is to adapt the well-known scatter search template [7] to multiobjective optimization. This template characterizes scatter search algorithms according to five methods that must be defined: diversification generation, improvement, reference set update, subset generation, and solution combination. Compared with EAs, scatter search is attractive because it is a structured strategy, in which is clearly stated where local searches can be applied (in the improvement method), and the diversification/intensification balance can be tuned in several ways: adjusting the size of the reference set, defining how the reference set is updated (reference set update method), and determining how to create new solutions (in the subset generation method).

Our goal is to design a competitive algorithm capable of beating the results produced by state-of-the-art multiobjective optimizers, such as NSGA-II and SPEA2. With this objective in mind, we have explored not only using a pure scatter search algorithm but also the possibility of using mutation and crossover operators if they 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 a hybridization of this algorithm with randomized operators typically used in EAs.

Manuscript received September 18, 2006; revised March 29, 2007, and June 20, 2007. First published February 22, 2008; last published July 30, 2008 (projected). This work was supported in part by the Spanish Ministry of Education and Science and FEDER under Contract TIN2005-08818-C04-01 (the OPLINK project). The work of J. J. Durillo is supported by Grant AP-2006-03349.

A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, and J. J. Durillo are with the Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, Málaga 29071, Spain (e-mail: antonio@lcc.uma.es; flv@lcc.uma.es; eat@lcc.uma.es; bernabe@lcc.uma.es; durillo@lcc.uma.es).

A. Beham is with the Research Center of the Upper Austrian University of Applied Sciences, A-4232 Hagenberg, Austria (e-mail: Andreas.Beham@fh-hagenberg.at).

Digital Object Identifier 10.1109/TEVC.2007.913109

AbYSS combines ideas from three state-of-the-art EAs 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 [6]; on the other hand, the selection of solutions from the initial set used to build the reference set applies the SPEA2 density estimation.

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 incorporates 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 analyze the performance of AbYSS by comparing it to NSGA-II and SPEA2, using several test functions and metrics taken from the specialized literature.

The rest of this paper is organized as follows. In Section II, we discuss related works concerning multiobjective optimization and scatter search. Section III is devoted to the description of our proposal. Experimental results, comparing AbYSS with other EAs for solving MOPs, are presented and analyzed in Section IV. Finally, we conclude the paper and give some lines of future work in Section V.

II. Related Work

The application of scatter search to multiobjective optimization has received little attention until recently. In fact, straightforward approaches and early studies were based on solving MOPs with the standard, single-objective scatter search algorithm.

Martí *et al.* [10] study the problem of assigning proctors to exams, which is formulated as a biobjective problem. However, the authors combine the objective functions to create a single, weighted function and the problem is solved as a mono-objective problem with the standard scatter search scheme. They do not seek to obtain a set of nondominated solutions.

The problem of routing school buses in a rural area is addressed by Corberán *et al.* [11]. This is a biobjective MOP aimed at minimizing, on the one hand, the number of buses required to transport students and, on the other hand, the time a given student spends on route. Although the authors develop a solution procedure that searches for a set of efficient solutions instead of a single optimum one, the approach uses 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 a common repository for efficient and nonefficient solutions.

Rao and Arvind [12] attempt laminate ply sequence optimization of hybrid composite panels in order to simultaneously optimize both the weight of the panel and its cost. The weighted sum approach is used to solve the multiobjective problem where the two objectives are combined into one overall objective function. The scatter search method used does not incorporate specialized mechanisms for supporting multiobjective functions. Tradeoff results are provided by using different values of the weights.

Krishna and Rao [13] address optimization of the grinding parameters of wheel speed, work piece speed, depth of dressing, and lead of dressing for the surface grinding process with a scatter search approach. This is a minimization problem in which objectives are normalized before the weighted function is computed.

Next, we analyze related works in which multiobjective versions of the scatter search technique are proposed as new algorithmic approaches.

A scatter search algorithm for solving the bicriteria multidimensional knapsack problem is presented by da Silva *et al.* in [14]. Even though the algorithm aims at finding a set of nondominated solutions, it is tightly tailored to solve a specific problem, and the scatter search methods differ significantly from those used in this work.

MOSS [15] is an algorithm that proposes a tabu/scatter search hybrid method for solving nonlinear MOPs. 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 with NSGA-II, SPEA2, and PESA on a set of unconstrained test functions.

Like MOSS, SSPMO [16] is a scatter search algorithm which includes tabu search, each uses different tabu search algorithms. SSPMO obtains part of the reference set by selecting the best solutions from the initial set for each objective function. The rest of the reference set is obtained using the usual approach of selecting the remaining solutions from the initial set which 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 using a benchmark of unconstrained test functions.

SSMO [17] is a scatter search-based 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 nondominated solutions found in the scatter search loop are stored. This algorithm is evaluated with a set of both unconstrained and constrained test functions.

A multiobjective scatter search algorithm, called *M-scatter search*, is presented by Vasconcelos *et al.* in [18]. The authors use the nondominated sorting technique and the niched-type penalty method of NSGA [19] to extend the scatter search algorithm to multiobjective optimization. M-scatter search also uses an *offline set* which stores nondominated solutions found during the computation. The NSGA niching method is applied in the updating procedure of the offline set, to keep nondominated solutions uniformly distributed along the Pareto front.

AbYSS is also applied to solve MOPs with continuous bounded variables. It follows the steps of the scatter search algorithm, but using mutation and crossover operators. Another

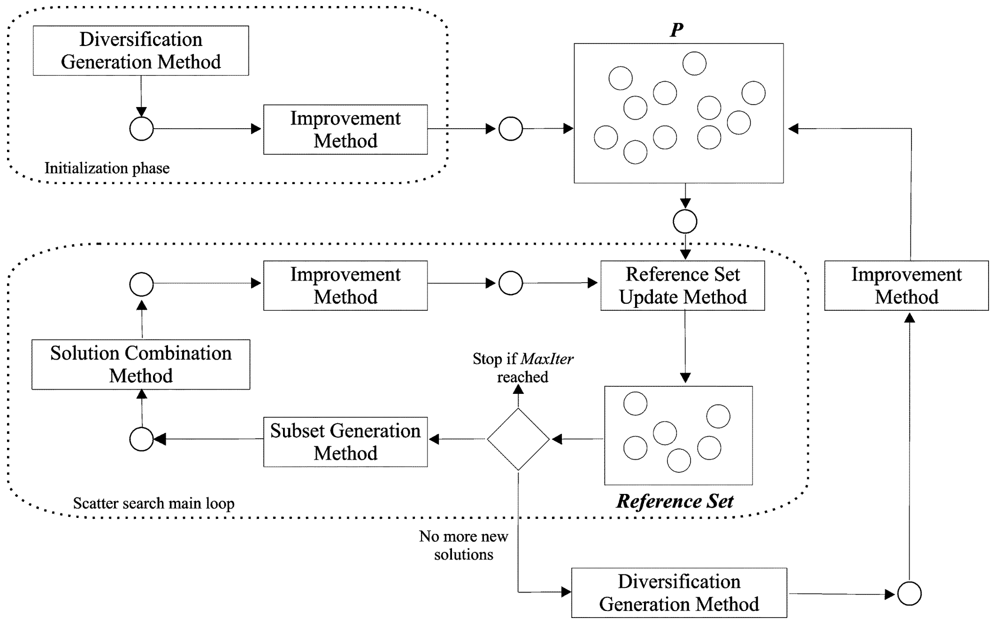

Fig. 1. Outline of the standard scatter search algorithm.

difference is that AbYSS uses two different density estimations in the algorithm.

III. Description of the Proposed Algorithm

AbYSS is based on the scatter search template proposed in [7] and its usage to solve bounded continuous single-objective optimization problems [9]. The template defines 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 should point out that scatter search is a generic strategy, and many decisions have to be made to design a specific scatter search algorithm. In particular, the balance between diversification and intensification must be carefully adjusted; otherwise, the algorithm may require a higher number of iterations to converge to reach accurate solutions. In this section, we give a generic description of AbYSS and consider that it is possible to combine different methods and parameters in the algorithm. These issues are studied in Section IV-C.

In Sections III-A–D, first we discuss the scatter search method in relation to the template; then we describe the five methods used in our approach, mainly focusing on the improvement and reference set update procedures, which constitute the basis of our proposal. We then look at how the external archive is managed. Finally, we present the overall algorithm.

A. The Scatter Search Template

The scatter search method starts by creating an initial set of diverse solutions 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 using the reference set update method. Then, solutions in the reference set are systematically grouped into subsets of two or more individuals using the subset generation method. In the next step, solutions in each subset are combined somehow to produce new individuals. This combination is defined by the solution combination method. The improvement method is applied to each newly generated solution, and the final step is to decide whether the resulting solution is inserted or not into the reference set. This loop is executed until a stopping condition is fulfilled (for example, a given number of iterations have been performed, or the subset generation method produces no more subsets).

Optionally, there is a restart process. The idea is to create a new initial set, containing the individuals currently in the reference set, and the remaining individuals are generated using the diversification generation and improvement methods, as in the initial phase.

B. Scatter Search Methods

In order to describe the algorithm, the initial set and the reference set are, respectively, named P and *RefSet* in the following.

1) Diversification Generation Method: The method is basically the same as that proposed in [9]. The goal is to generate an initial set P of diverse solutions. This is a simple method based on dividing the range of each variable into a number of subranges of equal size; then, the value for each decision variable of every solution is generated in two steps. First, a subrange of the variable is randomly chosen. The probability of selecting a subrange is inversely proportional to its frequency count (the number of times the subrange has already been selected). Second, a value is randomly generated within the selected range. This is repeated for all the solution decision variables.

2) Improvement Method: The idea behind this method is to use a local search algorithm to improve new solutions obtained from the diversification generation and solution combination methods (see Fig. 1). In [9], the *simplex* method is used; here we evaluate using this method (Section IV-C), but we also analyze an improvement strategy which is a simple $(1+1)$ EA. The $(1+1)\mathrm{EA}$ is based on using a mutation operator as perturbation and a Pareto dominance test. This way, we do not follow the scatter search guidelines on avoiding the use of stochastic operators. This approach has been chosen because of the simplicity of the final improvement method and also because of the benefits provided by using operators which have been proven to perform well in EAs. An outline of the method is shown in Fig. 2.

The improvement method takes an individual as a parameter, which is repeatedly mutated with the aim of obtaining a better individual. The term "better" is defined here in a similar way to the constrained-dominance approach used in NSGA-II [4]. A constraint violation test checks whether two individuals are feasible or not (line 6). If one of them is and the other is not, or both are infeasible but one has a smaller overall constraint violation, then the test returns the winner (line 7). Otherwise, a dominance test is used to decide whether one of the individuals dominates the other. 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 nondominated and the mutated individual is not dominated by the external archive, the original individual is moved into the external archive and the mutated individual becomes the new original one. This way we avoid getting worse solutions with respect to the archive as the improvement progresses.

Several features of the proposed improvement method should be noted. First, no nondominated solution is lost since in the case where several nondominated solutions are found in the procedure, they are inserted into the external archive. Second, by tuning the parameter *iter*, we can easily adjust the improvement effort and, therefore, the intensification capabilities of the optimizer as well.

We should point out here that the improvement method is based on a generic $(1+1)\mathrm{EA}$, not on $(1+1)$ evolution strategy (ES), since neither the use of a Gaussian mutation nor evolving the mutation strength are considered in AbYSS. This is because applying the self adaptation scheme typical of ES would require too many evaluations in the improvement method. We use a polynomial mutation operator, which is typically used by multiobjective genetic algorithms, such as NSGA-II and SPEA2.

```
1: Individual improvement(Individual originalIndividual, int iter) {
2:   Individual mutatedIndividual
3:   repeat iter times {
4:     mutatedIndividual = mutation(originalIndividual)
5:     if (the problem has constraints) {
6:       evaluateConstraints(mutatedIndividual)
7:       best = constraintTest(mutatedIndividual, originalIndividual)
8:       if (none of them is better than the other one) {
9:         evaluate(mutatedIndividual)
10:        best = dominanceTest(mutatedIndividual, originalIndividual)
11:      } // if
12:      else if (mutatedIndividual is best)
13:        evaluate (mutatedIndividual)
14:    } // if
15:    else { // the problem has no constraints
16:      evaluate(mutatedIndividual)
17:      best = dominanceTest(mutatedIndividual, originalIndividual)
18:    } // else
19:    if (mutatedIndividual is the best)
20:      originalIndividual = mutatedIndividual
21:    else if (originalIndividual is best)
22:      delete(mutatedIndividual)
23:    else if (mutated is not dominated by the external archive) {
24:      //both individuals are non-dominated as well
25:      insert originalIndividual into external archive
26:      originalIndividual = mutatedIndividual
27:    } // else if
28:    else
29:      delete(mutatedIndividual)
30: } // repeat
31: return originalIndividual
32:} // improvement
```

Fig. 2. Pseudocode describing the improvement method.

3) Reference Set Update Method: The reference set is a collection of both high-quality and diverse solutions that are used to generate new individuals. 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 promoting diversity. In [16], the set RefSet_2 is built by selecting those individuals from P whose minimum Euclidean distance to RefSet_1 is the highest. We use the same strategy to build RefSet_2, but, as is usual in the multiobjective optimization domain, we have to define the concept of "best individual" to build RefSet_1. 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). An outline of this method is included in Fig. 3.

To select the best p individuals of P (line 3), we use the approach used in SPEA2, i.e., the individuals are assigned a fitness

```
1: referenceSetUpdate(bool build) {
2:   if (build) { // build a new reference set
3:      select the p best individuals of P
4:      build the RefSet1 with these p individuals
5:      compute Euclidean distances in P to obtain q individuals
6:      build the RefSet2 with these q individuals
7:   } // if
8:   else { // update the reference set
9:      for (each new solution s) {
10:        test to insert s into RefSet1
11:        if (test fails)
12:           test to insert s into RefSet2
13:           if (test fails)
14:              delete s
15:     } // for
16:  } // else
17:} // referenceSetUpdate
```

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

```
// 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.

value which is the sum of their strength raw fitness and a density estimation [5]. The strength of an individual is the number of solutions it dominates in a population, and its strength raw fitness is the sum of the strengths of its dominator individuals. The density estimation is based on computing the distance to the kth nearest neighbor (see [5] for further details).

Once the reference set is filled, its solutions are combined to obtain new solutions, which are then improved. Afterwards, they are tested for inclusion in the reference set (line 8 in Fig. 3). 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 a 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 line 10 of Fig. 3) we cannot use a ranking procedure because the size of this population is usually 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 for dealing with constraints is as explained in the improvement method in Fig. 2.)

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.

4) Subset Generation Method: According to the scatter search template, this method generates subsets of individuals, which will be used to create new solutions with the solution combination method. Several kinds of subsets are possible [9]. The most usual strategy considers 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 produces, on the one hand, pairwise combinations of individuals from RefSet_1 and, on the other, pairwise combinations of individuals from RefSet_2. Our preliminary experiments (see Section IV-C) revealed that generating combinations of individuals from the two subsets makes the algorithm 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 imbalance between intensification and diversification. As as result, the algorithm requires a larger number of iterations for converging to an accurate Pareto front.

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 realized that the results were very competitive for many problems, but the algorithm failed when trying to solve some difficult problems. In Section IV-C we analyze the use of a simulated binary crossover operator (SBX) [20] instead, concluding that this crossover operator makes AbYSS more robust.

C. Managing the External Archive

The main objective of the external archive (or repository) is to store a record of the nondominated individuals found during the search process, in order to keep those individuals producing a well-distributed Pareto front. The key issue in 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 pairwise with the contents of the archive. If this new solution is dominated by an individual from the archive (i.e., the solution is dominated by the archive), then such solution is discarded; otherwise, the solution is stored. 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 to decide which individual has to be removed. The strategy used in other archive-based EAs when the archive is full, such as PAES [6] and MOPSO [21], 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 of NSGA-II [4]. 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. However, we decided to use two different density estimations with the aim of hopefully profiting from the combination of both in different parts of our algorithm, and thus obtaining a better distributed Pareto front. The rationale for this decision is that nondominated solutions may have to pass two filters: first, they are not in the densest region, according to the crowding distance, and second, they are obtained from the best individuals of the initial set according to the density estimation. We have done some experiments comparing the use of only one density estimator in AbYSS, but combining both yielded the best results.

D. 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 one is passed to the improvement method (line 1). The result is the initial set P. Then, a number of iterations is performed (the outer loop in Fig. 5).

```
1:  construct the initial set P
2:  // outer loop
3:  until (stop condition) {
4:     referenceSetUpdate(build=true)
5:     subsetGeneration()
6:     // scatter search main loop
7:     while (new subsets are generated) {
8:        combination()
9:        for (each combined individual) {
10:          improvement()
11:          referenceSetUpdate(build=false)
12:       } // for
13:       subsetGeneration()
14:    } // while
15:    // Re-start
16:    add RefSet_1 to P
17:    move the best n individuals from the archive to P
18:    fill P with new solutions
19: } // until
```

Fig. 5. Outline of the AbYSS algorithm.

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 stops producing new subsets of solutions (lines 4–13). Then, there is a restart phase, which consists of three steps. First, the individuals in RefSet_1 are inserted into P; second, the best n individuals from 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 for filling up the set P.

The idea of moving n individuals from the archive to the initial set (line 17) is to promote the intensification capabilities of the search towards the Pareto front already found. The intensification degree can vary depending on the number of chosen individuals. We use a value of n that is the minimum of the size of the archive and half 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 IV-D).

TABLE I
PROPERTIES OF THE MOPS CREATED USING THE WFG TOOLKIT

Problem	Separability	Modality	Bias	Geometry
WFG1	separable	uni	polynomial, flat	convex, mixed
WFG2	non-separable	f_1 uni, f_2 multi	no bias	convex, disconnected
WFG3	non-separable	uni	no bias	linear, degenerate
WFG4	non-separable	multi	no bias	concave
WFG5	separable	deceptive	no bias	concave
WFG6	non-separable	uni	no bias	concave
WFG7	separable	uni	parameter dependent	concave
WFG8	non-separable	uni	parameter dependent	concave
WFG9	non-separable	multi, deceptive	parameter dependent	concave

IV. EXPERIMENTATION

This section is devoted to presenting the experiments performed in this work. We first detail the set of MOPs used as a benchmark and the metrics applied for measuring the performance of the resulting Pareto fronts. Next, our preliminary experiments for tuning AbYSS are described and analyzed. Finally, we evaluate our proposal and compare it to NSGA-II and SPEA2.

A. Test Problems

In this section, we describe the different sets of both constrained and unconstrained problems solved in this work. These problems have been used in many studies in this area.

We have selected first the following biobjective unconstrained problems: Schaffer [22], Fonseca [23], and Kursawe [24], as well as the problems ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6, which are all defined in [25]. We also include in this set the biobjective version of the nine problems, WFG1 to WFG9, defined in [26] using the WFG Toolkit. The properties of these problems are detailed in Table I. The second set is composed of the following constrained biobjective problems: Osyczka2 [27], Tanaka [28], Srinivas [19], Constr_Ex [4], and Golinski [29]. Finally, we have included problems of more than two objectives. The first group of problems is Viennet2, Viennet3, Viennet4 [30], and Water [31]. The first two have three objectives and zero constraints, the third one has three objectives and three constraints, and the last one has five objectives and seven constraints. The second group is composed of the DTLZ family of scalable test problems [32].

B. Performance Metrics

To assess the performance of algorithms on the test problems, two different issues are normally taken into account: the distance between the Pareto front generated by the proposed algorithm to the exact Pareto front should be minimized and the spread of solutions found should be maximized in order to obtain as smooth and uniform a distribution of vectors as possible. To determine these issues, it is necessary to know the exact location of the true Pareto front. In most of the benchmark problems used in this work, their Pareto fronts are known (families ZDT, DTLZ, and WFG); in the other problems, we have obtained their Pareto fronts using an enumerative search strategy [33].

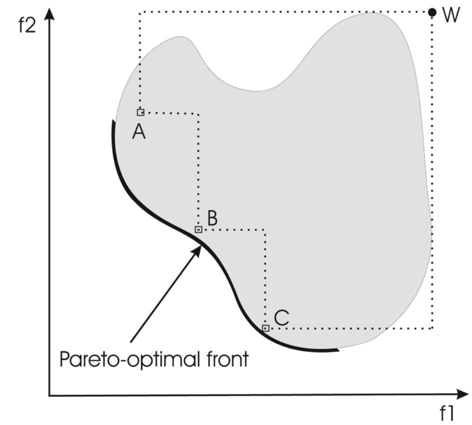

Fig. 6. The HV enclosed by the nondominated solutions.

The performance metrics can be classified into three categories depending on whether they evaluate the closeness to the Pareto front, the diversity in the solutions obtained, or both [2]. We have adopted one metric of each type.

- **Generational Distance (GD)**: This metric was introduced by Van Veldhuizen and Lamont [34] to measure how far the elements are in the set of nondominated vectors found from those in the Pareto optimal set and it is defined as

$$(\mathrm{GD}) = \frac{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{d}_{\mathrm{i}}^2}}{\mathrm{n}} \quad (1)$$

where n is the number of vectors in the set of nondominated 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 $\mathrm{GD} = 0$ indicates that all the generated elements are in the Pareto front. In order to obtain reliable results, nondominated sets are normalized before calculating this distance measure.

- **Spread**: The original Spread metric [4] is based on calculating the distance between two consecutive solutions, which works only for two-objective problems. We extend this metric by computing the distance from a given point to its nearest neighbor. This modification is based on the metric proposed in [35]

$$\Delta = \frac{\sum_{i=1}^{m} d(e_i, S) + \sum_{X \in S} \left| d(X, S) - \bar{d} \right|}{\sum_{i=1}^{m} d(e_i, S) + |S| * \bar{d}} \quad (2)$$

where S is a set of solutions, S^* is the set of Pareto optimal solutions, $(e_1, \ldots, e_m)$ are m extreme solutions in S^*, m is the number of objectives, and

$$d(X, S) = \min_{Y \in S, Y \neq X} ||F(X) - F(Y)||^2 \quad (3)$$

$$\bar{d} = \frac{1}{|S^*|} \sum_{X \in S^*} d(X, S). \quad (4)$$

TABLE II
SUMMARY OF THE DIFFERENT CONFIGURATIONS USED IN AbYSS

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6
Improvement Method	Simplex	(1 + 1) EA	(1 + 1) EA	(1 + 1) EA	(1 + 1) EA	(1 + 1) EA
Size of P	100	100	20	20	20	20
Pairwise Generation	$RefSet_1 \cup RefSet_2$	$RefSet_1 \cup RefSet_2$	$RefSet_1 \cup RefSet_2$	$RefSet_1, RefSet_2$	$RefSet_1, RefSet_2$	$RefSet_1, RefSet_2$
Solution Combination	Linear Combination	Linear Combination	Linear Combination	Linear Combination	SBX	SBX
Improvement Iters	5	1	1	1	1	3

If the achieved solutions are well distributed and include those extreme solutions, $\Delta = 0$. We apply this metric after a normalization of the objective function values (see the appendix for a further analysis of this issue).

- **Hypervolume (HV)**: This metric calculates the volume (in the objective space) covered by members of a nondominated set of solutions Q (the region enclosed within the discontinuous line in Fig. 6, $Q = \{A, B, C\}$) for problems where all objectives are to be minimized [36]. 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 be found simply by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its hypervolume (HV) is calculated

$$\mathrm{HV} = \mathrm{volume}\left(\bigcup_{i=1}^{|Q|} v_i\right). \tag{5}$$

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

C. Study of Different Parameter Configurations

As commented before, several issues should be studied in order to make decisions relating to parameters defining the behavior of AbYSS. Although an extensive analysis of the parameters of AbYSS is outside the scope of this paper, we study a number of them. In particular, we focus on the following issues.

- The use of the simplex method against the $(1+1)$ EA in the improvement phase.
- The application of linear combination versus 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.
- The number of iterations in the improvement method.

We have performed six experiments with the aim of clarifying the influence of these issues on the search capabilities of AbYSS. For this purpose, the problems ZDT1 (convex), ZDT2 (nonconvex), ZDT3 (nonconvex, disconnected), ZDT4 (convex, multimodal), and ZDT6 (nonconvex, nonuniformly spaced) [25] have been chosen from our benchmark (Section IV-A). We consider that the features of these problems make them meaningful enough for this preliminary study.

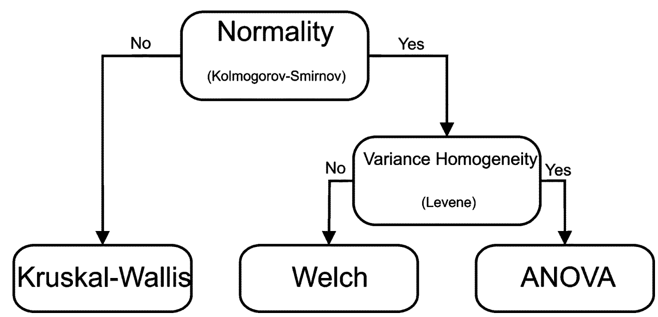

Fig. 7. Statistical analysis performed in this work.

Prior to describing the experiments, we detail the simplex method we have used. Taking the Nelder–Mead simplex algorithm [37] as starting point, we have had to adapt it to the multiobjective optimization domain. The original method is a nonlinear optimization algorithm commonly used to minimize an objective function in a many-dimensional space. It can be seen as a geometric method since it works on a shape (the simplex) of $m+1$ points (being m the number of dimensions of the problem) which is iteratively transformed by means of a set of rules (reflection, expansion, and contraction) so that the simplex is moved within the search space until convergence criteria are met. This method works as follows. The initial simplex is built by using nondominated solutions from the archive. (This is aimed at reducing the number of evaluations required.) Next, the vertexes are sorted using a nonweighted aggregative function (the simplest and fastest approach in this scenario with such a small number of nondominated solutions). The best vertex is then transformed with the standard Nelder–Mead operations so that the newly generated vertex is compared with the original untransformed one using a dominance test. If the former either dominates the later or both are nondominated, the new vertex is inserted into the simplex, it is discarded otherwise. This algorithms iterates only five times.

The following experiments were carried out (Table II includes a summary of all the configurations).

1) **Experiment 1**: We configure AbYSS with a typical configuration for a scatter search algorithm: we use the simplex algorithm in the improvement method, linear combinations to create new trial solutions are applied in the solution combination method, the size of P is 100, the subset generation method generates all pairwise combinations of individuals belonging to both RefSet_1 and RefSet_2, and the size of both RefSet_1 and RefSet_2 is 10.

TABLE III
RESULTS OF EXECUTING DIFFERENT CONFIGURATIONS OF ABYSS

	Generational Distance (GD)						
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6	
Problem	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	
ZDT1	4.364e−03 $_{9.5e-03}$	1.424e−01 $_{8.3e-02}$	1.773e−01 $_{1.0e-01}$	8.192e−02 $_{3.3e-02}$	1.826e−04 $_{4.0e-05}$	2.144e−04 $_{5.0e-05}$	+
ZDT2	1.803e+00 $_{8.6e-01}$	8.551e−01 $_{1.4e+00}$	1.050e+00 $_{1.4e+00}$	3.502e−02 $_{5.0e-02}$	1.057e−04 $_{5.8e-05}$	1.374e−04 $_{6.6e-05}$	+
ZDT3	1.244e−02 $_{1.7e-02}$	1.318e−01 $_{5.5e-02}$	1.592e−01 $_{6.6e-02}$	9.219e−02 $_{2.7e-02}$	1.954e−04 $_{2.2e-05}$	1.986e−04 $_{2.3e-05}$	+
ZDT4	0.000e−00 $_{0.0e-00}$	2.576e+01 $_{7.8e+00}$	2.482e+01 $_{9.2e+00}$	3.336e−01 $_{3.5e-01}$	5.204e−04 $_{3.9e-04}$	7.277e−04 $_{4.9e-04}$	+
ZDT6	8.402e−01 $_{7.2e-01}$	3.429e−01 $_{1.1e+00}$	6.106e−01 $_{1.2e+00}$	1.642e−01 $_{1.1e-01}$	5.514e−04 $_{1.8e-05}$	5.633e−04 $_{2.5e-05}$	+
	Spread (Δ)						
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6	
Problem	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	
ZDT1	7.708e−01 $_{1.6e-01}$	8.740e−01 $_{9.1e-02}$	9.039e−01 $_{8.1e-02}$	7.439e−01 $_{4.6e-02}$	1.063e−01 $_{1.5e-02}$	1.290e−01 $_{2.2e-02}$	+
ZDT2	1.111e+05 $_{1.1e+08}$	9.910e−01 $_{6.7e-02}$	9.949e−01 $_{2.6e+00}$	7.665e−01 $_{2.0e-01}$	1.085e−01 $_{1.8e-02}$	1.359e−01 $_{2.6e-02}$	+
ZDT3	8.259e−01 $_{1.8e-01}$	9.218e−01 $_{6.5e-02}$	9.230e−01 $_{7.5e-02}$	8.407e−01 $_{5.1e-02}$	1.365e−01 $_{2.1e-01}$	1.644e−01 $_{4.5e-02}$	+
ZDT4	1.000e−00 $_{0.0e-00}$	9.841e−01 $_{7.7e-02}$	9.712e−01 $_{8.7e-02}$	9.736e−01 $_{7.4e-02}$	1.273e−01 $_{3.7e-02}$	1.442e−01 $_{3.3e-02}$	+
ZDT6	1.190e+00 $_{3.7e-01}$	1.322e+00 $_{5.2e-01}$	1.116e+00 $_{4.2e-01}$	9.587e−01 $_{3.2e-01}$	9.002e−02 $_{1.7e-02}$	1.188e−01 $_{2.0e-02}$	+
	Hypervolume (HV)						
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Experiment 6	
Problem	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	$\tilde{x}_{IQR}$	
ZDT1	6.019e−01 $_{8.5e-02}$	0.000e+00 $_{1.1e-03}$	0.000e+00 $_{0.0e+00}$	4.015e−02 $_{8.2e-02}$	6.614e−01 $_{3.3e-04}$	6.607e−01 $_{5.4e-04}$	+
ZDT2	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	8.275e−02 $_{1.9e-01}$	3.281e−01 $_{3.7e-04}$	3.276e−01 $_{5.3e-04}$	+
ZDT3	3.832e−01 $_{1.1e-01}$	0.000e+00 $_{1.5e-04}$	0.000e+00 $_{0.0e+00}$	3.481e−03 $_{2.6e-02}$	5.158e−01 $_{3.4e-03}$	5.155e−01 $_{2.8e-04}$	+
ZDT4	0.000e−00 $_{0.0e-00}$	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	6.553e−01 $_{6.1e-03}$	6.519e−01 $_{7.2e-03}$	+
ZDT6	1.616e−01 $_{2.8e-01}$	3.130e−01 $_{3.8e-01}$	1.717e−01 $_{3.5e-01}$	3.894e−01 $_{3.4e-01}$	4.004e−01 $_{1.8e-04}$	3.996e−01 $_{5.0e-04}$	+

2) **Experiment 2**: We replace the simplex method by the $(1+1)$ EA in the improvement phase, using a distribution index of 20 in the polynomial mutation. We start by using a number of iterations equal to one.
3) **Experiment 3**: We keep the same parameter settings as in the previous experiment, but the size of P is reduced to 20 individuals. As the restart strategy works, this means that ten solutions come from RefSet_1 and the other ten are copied from the external archive, so no newly random solution is included when restarting and the intensification of AbYSS around the current nondominated solutions is stressed.
4) **Experiment 4**: AbYSS is configured as in Experiment 3, but the subset generation method produces pairs of individuals belonging only to RefSet_1 or RefSet_2. The aim is to intensify the search in two ways. Explicitly, by reducing the number of combinations with diverse solutions coming from RefSet_2, and also implicitly, since the lower the number of combinations the shorter the inner loop of AbYSS and, therefore, the higher the number of restarts promoting feedback of nondominated solutions from the external archive.
5) **Experiment 5**: Using stochastic procedures instead of linear combinations in the solution combination method has produced improved results due to their enhanced exploration capabilities in single-objective scatter search [38]. For this reason, we repeat Experiment 4, but using the SBX operator in the solution combination method, with a distribution index of 20.
6) **Experiment 6**: In the last experiment, we repeat Experiment 5, but apply three iterations in the improvement method. The idea is to enhance the intensification of the search in AbYSS at a low cost.

The algorithm stops when 25 000 function evaluations have been computed. We have made 100 independent runs of each experiment, and the results obtained are shown in Table III. Since

TABLE IV
MEDIAN AND INTERQUARTILE RANGE OF THE GENERATIONAL DISTANCE (GD) METRIC (25 000 FUNCTION EVALUATIONS)

Problem	AbYSS $\tilde{x}_{IQR}$	NSGA-II $\tilde{x}_{IQR}$	SPEA2 $\tilde{x}_{IQR}$	
Schaffer	2.330e−04 $_{1.8e-05}$	2.348e−04 $_{1.7e-05}$	2.374e−04 $_{1.5e-05}$	+
Fonseca	2.007e−04 $_{2.4e-05}$	4.687e−04 $_{5.3e-05}$	2.254e−04 $_{3.3e-05}$	+
Kursawe	1.421e−04 $_{1.4e-05}$	2.055e−04 $_{3.1e-05}$	1.611e−04 $_{1.9e-05}$	+
ZDT1	1.826e−04 $_{4.0e-05}$	2.196e−04 $_{4.8e-05}$	1.981e−04 $_{1.7e-05}$	+
ZDT2	1.057e−04 $_{5.8e-05}$	1.667e−04 $_{4.2e-05}$	1.267e−04 $_{3.0e-05}$	+
ZDT3	1.954e−04 $_{2.2e-05}$	2.165e−04 $_{2.1e-05}$	2.330e−04 $_{2.1e-05}$	+
ZDT4	5.204e−04 $_{3.9e-04}$	4.145e−04 $_{3.5e-04}$	5.743e−02 $_{3.3e-02}$	+
ZDT6	5.514e−04 $_{1.8e-05}$	9.911e−04 $_{9.8e-05}$	8.200e−04 $_{7.3e-05}$	+
WFG1	2.651e−02 $_{1.3e-02}$	2.424e−02 $_{9.6e-03}$	4.802e−02 $_{1.5e-04}$	+
WFG2	4.335e−04 $_{2.4e-05}$	4.598e−04 $_{1.5e-04}$	2.365e−02 $_{4.9e-04}$	+
WFG3	1.381e−04 $_{1.2e-05}$	1.856e−04 $_{2.7e-05}$	1.657e−02 $_{4.7e-06}$	+
WFG4	6.273e−04 $_{2.6e-05}$	6.464e−04 $_{7.5e-05}$	2.032e−02 $_{7.6e-05}$	+
WFG5	2.638e−03 $_{1.1e-05}$	2.661e−03 $_{3.3e-05}$	2.677e−03 $_{1.7e-05}$	+
WFG6	3.282e−03 $_{3.5e-03}$	5.687e−04 $_{6.3e-04}$	1.368e−02 $_{1.5e-04}$	+
WFG7	3.058e−04 $_{2.3e-05}$	3.445e−04 $_{5.0e-05}$	3.675e−02 $_{3.6e-04}$	+
WFG8	1.482e−02 $_{4.9e-03}$	1.023e−02 $_{5.5e-03}$	3.843e−03 $_{1.6e-04}$	+
WFG9	1.050e−03 $_{8.1e-05}$	1.090e−03 $_{1.4e-04}$	1.955e−02 $_{3.0e-04}$	+
ConstrEx	2.030e−04 $_{4.6e-05}$	2.883e−04 $_{4.2e-05}$	2.070e−04 $_{2.3e-05}$	+
Srinivas	5.133e−05 $_{1.9e-05}$	1.884e−04 $_{4.7e-05}$	1.123e−04 $_{2.8e-05}$	+
Osyczka2	1.119e−03 $_{1.6e-02}$	1.049e−03 $_{8.3e-05}$	1.408e−03 $_{8.1e-05}$	+
Golinski	3.065e−04 $_{3.0e-05}$	3.198e−04 $_{3.5e-05}$	2.480e−04 $_{1.6e-04}$	+
Tanaka	7.784e−04 $_{1.0e-04}$	1.217e−03 $_{1.2e-04}$	7.225e−04 $_{8.9e-05}$	+
Viennet2	7.944e−04 $_{3.8e-04}$	8.012e−04 $_{4.6e-04}$	8.516e−04 $_{2.2e-04}$	-
Viennet3	2.364e−04 $_{1.7e-04}$	2.238e−04 $_{6.8e-05}$	2.976e−04 $_{1.3e-04}$	+
Viennet4	3.341e−04 $_{1.2e-04}$	4.590e−04 $_{1.6e-04}$	5.580e−04 $_{1.9e-04}$	+
Water	8.286e−03 $_{2.1e-03}$	6.160e−03 $_{1.5e-03}$	1.682e−02 $_{2.0e-03}$	+
DTLZ1	4.516e−03 $_{7.1e-02}$	1.819e−03 $_{5.4e-03}$	1.087e+00 $_{1.8e+00}$	+
DTLZ2	7.285e−04 $_{6.8e-05}$	1.254e−03 $_{1.9e-04}$	1.834e−03 $_{4.7e-04}$	+
DTLZ3	6.502e−01 $_{6.3e-01}$	1.693e+00 $_{1.9e+00}$	7.905e+00 $_{4.7e+00}$	+
DTLZ4	4.865e−03 $_{3.7e-04}$	4.430e−03 $_{4.3e-04}$	5.221e−03 $_{2.0e-03}$	+
DTLZ5	2.497e−04 $_{4.1e-05}$	6.342e−04 $_{5.7e-05}$	7.096e−04 $_{7.6e-05}$	+
DTLZ6	1.173e−01 $_{4.2e-02}$	1.053e−01 $_{1.1e-02}$	1.093e−01 $_{1.3e-02}$	+
DTLZ7	1.880e−03 $_{1.2e-03}$	2.512e−03 $_{3.8e-04}$	3.470e−03 $_{1.7e-03}$	+

TABLE V
MEDIAN AND INTERQUARTILE RANGE OF THE SPREAD (Δ) METRIC (25 000 FUNCTION EVALUATIONS)

Problem	AbYSS $\tilde{x}_{IQR}$	NSGA-II $\tilde{x}_{IQR}$	SPEA2 $\tilde{x}_{IQR}$	
Schaffer	1.197e−01 $_{2.5e-02}$	4.250e−01 $_{7.1e-02}$	1.293e−01 $_{1.9e-02}$	+
Fonseca	9.853e−02 $_{1.9e-02}$	3.759e−01 $_{4.9e-02}$	1.289e−01 $_{1.8e-02}$	+
Kursawe	2.385e−01 $_{9.3e-03}$	5.429e−01 $_{8.9e-02}$	3.018e−01 $_{1.8e-02}$	+
ZDT1	1.063e−01 $_{1.5e-02}$	3.869e−01 $_{6.1e-02}$	1.449e−01 $_{2.1e-02}$	+
ZDT2	1.085e−01 $_{1.8e-02}$	3.928e−01 $_{7.4e-02}$	1.495e−01 $_{3.3e-02}$	+
ZDT3	1.365e−01 $_{2.1e-01}$	3.831e−01 $_{6.8e-02}$	1.656e−01 $_{2.0e-02}$	+
ZDT4	1.273e−01 $_{3.7e-02}$	4.028e−01 $_{6.2e-02}$	4.550e−01 $_{1.8e-01}$	+
ZDT6	9.002e−02 $_{1.7e-02}$	4.356e−01 $_{4.7e-02}$	1.578e−01 $_{2.1e-02}$	+
WFG1	5.531e−01 $_{1.5e-01}$	7.556e−01 $_{1.1e-01}$	7.365e−01 $_{4.3e-02}$	+
WFG2	3.612e−01 $_{2.0e-02}$	5.373e−01 $_{1.7e-01}$	8.927e−01 $_{6.6e-02}$	+
WFG3	3.725e−01 $_{1.1e-02}$	6.122e−01 $_{4.3e-02}$	8.210e−01 $_{5.0e-03}$	+
WFG4	1.113e−01 $_{2.0e-02}$	3.567e−01 $_{5.8e-02}$	4.278e−01 $_{1.9e-02}$	+
WFG5	1.325e−01 $_{1.9e-02}$	3.968e−01 $_{6.4e-02}$	3.629e−01 $_{2.6e-02}$	+
WFG6	1.479e−01 $_{4.7e-02}$	3.757e−01 $_{6.5e-02}$	6.583e−01 $_{1.3e-02}$	+
WFG7	1.116e−01 $_{2.3e-02}$	3.731e−01 $_{6.3e-02}$	5.517e−01 $_{1.6e-02}$	+
WFG8	4.552e−01 $_{6.4e-02}$	5.289e−01 $_{5.5e-02}$	8.150e−01 $_{1.5e-02}$	+
WFG9	1.163e−01 $_{2.4e-02}$	3.590e−01 $_{5.8e-02}$	4.614e−01 $_{2.7e-02}$	+
ConstrEx	1.569e−01 $_{2.1e-02}$	4.153e−01 $_{5.6e-02}$	5.166e−01 $_{4.4e-02}$	+
Srinivas	7.212e−02 $_{1.6e-02}$	3.707e−01 $_{5.6e-02}$	1.495e−01 $_{2.4e-02}$	+
Osyczka2	4.181e−01 $_{1.3e-01}$	6.474e−01 $_{1.3e-01}$	6.077e−01 $_{8.0e-02}$	+
Golinski	1.379e−01 $_{2.8e-02}$	5.346e−01 $_{8.7e-02}$	7.041e−01 $_{1.3e-01}$	+
Tanaka	2.925e−01 $_{5.3e-02}$	4.111e−01 $_{5.9e-02}$	2.309e−01 $_{3.7e-02}$	+
Viennet2	2.637e−01 $_{3.6e-02}$	4.569e−01 $_{7.0e-02}$	2.059e−01 $_{3.0e-02}$	+
Viennet3	2.194e−01 $_{5.5e-02}$	4.604e−01 $_{6.5e-02}$	5.950e−01 $_{3.7e-02}$	+
Viennet4	4.381e−01 $_{4.4e-02}$	5.740e−01 $_{7.9e-02}$	3.699e−01 $_{4.1e-02}$	+
Water	4.774e−01 $_{5.6e-02}$	4.773e−01 $_{5.7e-02}$	4.852e−01 $_{4.7e-02}$	-
DTLZ1	5.126e−01 $_{1.6e-01}$	5.449e−01 $_{1.1e-01}$	1.428e+00 $_{5.9e-01}$	+
DTLZ2	5.000e−01 $_{5.9e-02}$	5.201e−01 $_{7.0e-02}$	1.042e−01 $_{1.4e-02}$	+
DTLZ3	8.148e−01 $_{2.0e-01}$	1.384e+00 $_{6.2e-01}$	1.263e+00 $_{2.9e-01}$	+
DTLZ4	4.634e−01 $_{6.5e-02}$	4.860e−01 $_{9.2e-02}$	1.153e−01 $_{4.5e-01}$	+
DTLZ5	1.497e−01 $_{2.1e-02}$	4.456e−01 $_{7.0e-02}$	1.892e−01 $_{2.8e-02}$	+
DTLZ6	7.059e−01 $_{5.5e-02}$	6.326e−01 $_{6.7e-02}$	2.756e−01 $_{2.5e-02}$	+
DTLZ7	5.439e−01 $_{1.0e-01}$	5.091e−01 $_{6.0e-02}$	2.577e−01 $_{3.8e-02}$	+

we are dealing with stochastic algorithms and we want to provide the results with confidence, the following statistical analysis has been performed throughout this work [39], [40]. First, a Kolmogorov–Smirnov test was performed in order to check whether the values of the results follow a normal (Gaussian) distribution or not. If the distribution is normal, the Levene test checks for the homogeneity of the variances. If samples have equal variance (positive Levene test), an ANOVA test is done;

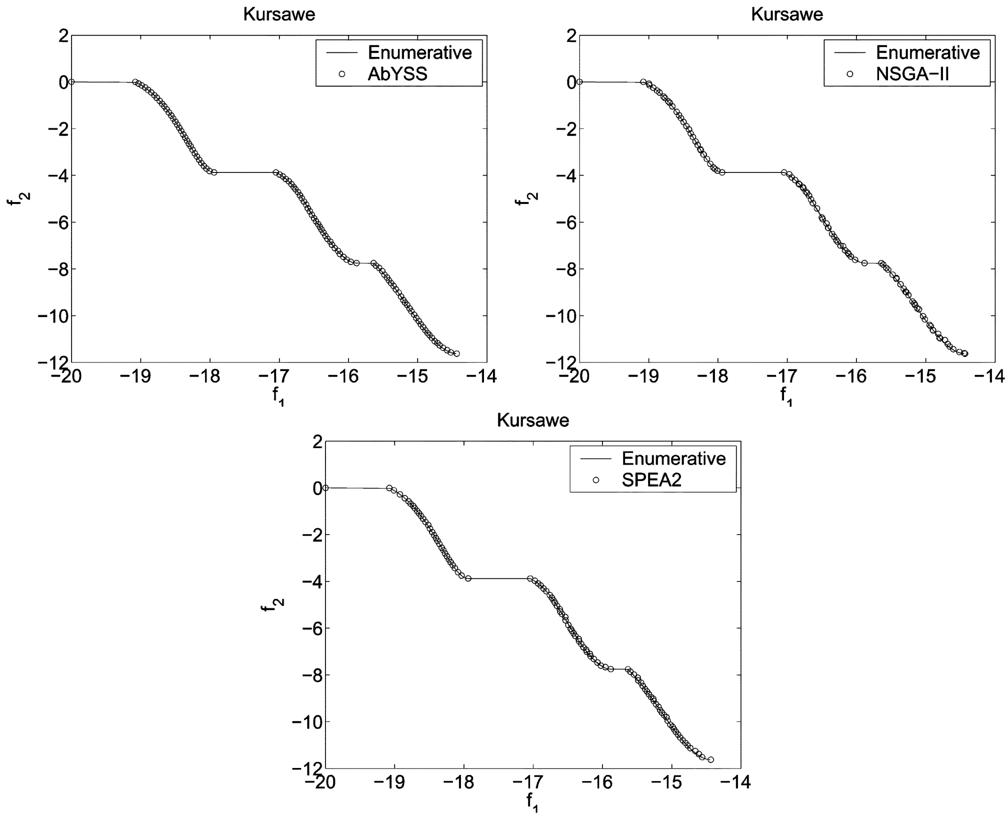

Fig. 8. AbYSS finds a better spread of solutions than SPEA2 and NSGA-II in the Kursawe problem.

otherwise, a Welch test is performed. For non-Gaussian distributions, the nonparametric Kruskal–Wallis test is used to compare the medians of the algorithms. Fig. 7 summarizes the statistical analysis. We always consider in this work a confidence level of 95% (i.e., significance level of 5% or p-value under 0.05) in the statistical tests, which means that the differences are unlikely to have occurred by chance with a probability of 95%. Successful tests are marked with "+" symbols in the last column in all the tables; conversely, "−" means that no statistical confidence was found (p-value > 0.05). Looking for homogeneity in the presentation of the results, Table III, respectively, includes the median, $\tilde{x}$, and interquartile range (IRQ), as measures of location (or central tendency) and statistical dispersion, because, on the one hand, some samples follow a Gaussian distribution while others do not, and, on the other hand, mean and median are theoretically the same for Gaussian distributions. Indeed, whereas the median is a feasible measure of central tendency in both Gaussian and non-Gaussian distributions, using the mean only makes sense for Gaussian ones. The same holds for the standard deviation and the IRQ. The best result for each problem has a gray colored background. We now analyze each experiment in detail.

1) Experiment 1: The results show that the configuration used in this experiment reaches the third best values in GD and HV for ZDT1, but it obtains the worst values for ZDT2. Indeed, the HV metric gets a value of zero in ZDT2, which means that all the solutions are outside the limits of the true Pareto front; when applying this metric, these solutions are not considered because, otherwise, the obtained value would be unreliable. A major issue emerges when solving ZDT4 with this configuration: AbYSS converges to one single solution which is one of the extreme solutions of the optimal Pareto front. This therefore means that GD values are perfect (zero distance), but no volume is covered by these solutions, so $\mathrm{HV} = 0$. Since obtaining a Pareto front composed of only one point is not a satisfactory solution to the problem, we have considered the GD value over ZDT4 as the worst of all the compared algorithms in order to avoid wrong conclusions when examining the values in Table III.

2) Experiment 2: In this experiment (see Table III, second column), the simplex algorithm is replaced by the $(1+1)\mathrm{EA}$. Compared with the previous experiment, it improves upon the GD values in ZDT2, ZDT4, and ZDT6. Concerning the Spread metric, this configuration reaches an acceptable result for ZDT2, while keeping the same behavior for the remaining ones. However, the zero HV values in all the MOPs but ZDT6 shows that here AbYSS hardly converges to the true Pareto front. The metrics indicate that ZDT4 is not satisfactorily solved.

3) Experiment 3: As stated before, the goal is to enhance the intensification capabilities of AbYSS by reducing the size of the initial set. We consider this reduction because preliminary experiments seemed to indicate that using large sizes for P had a negative influence on the convergence of the algorithm when solving some problems. The results of this experiment (Table III, third column) show that reducing the size of the initial set P to 20 individuals produces similar metric values as in Experiments 1 and 2 (the exception is GD for ZDT1 in Experiment 1). Again, HV shows that the algorithm converges poorly to the optimal Pareto front.

4) Experiment 4: With the idea in mind of investigating whether the diversification/intensification balance of AbYSS is penalized if the subset generation method produces pairs of

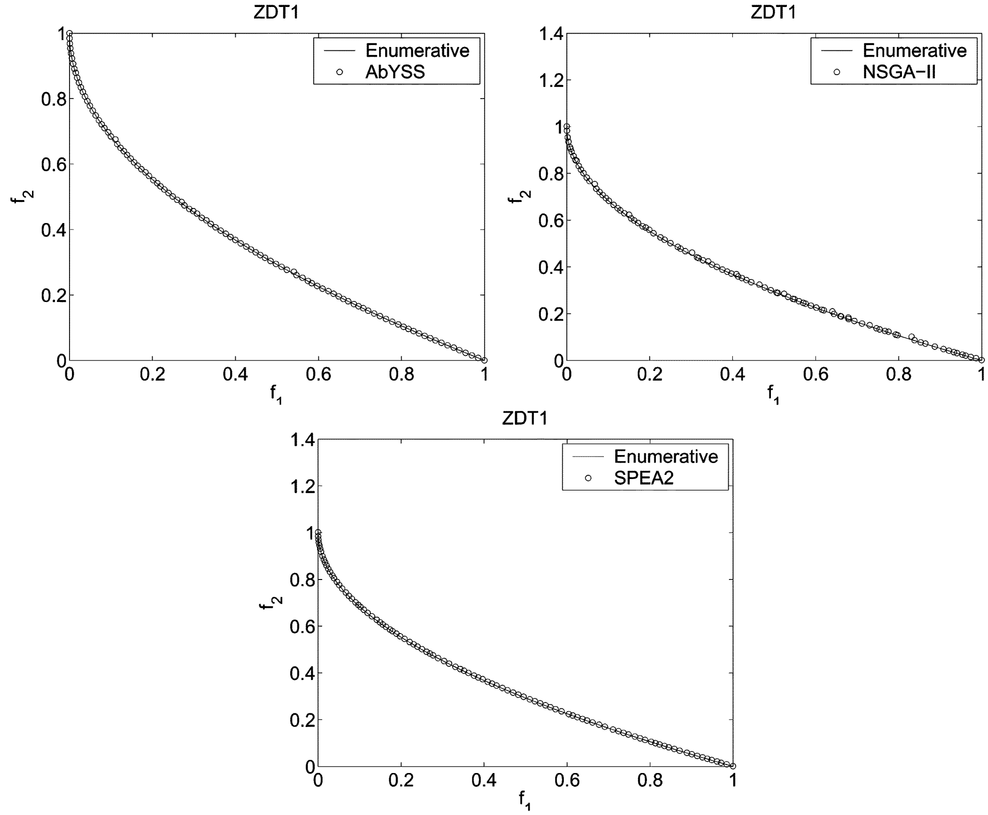

Fig. 9. Pareto fronts obtained with the three algorithms on problem ZDT1.

individuals belonging to RefSet_1 and RefSet_2, in this experiment, we only allow combinations of individuals belonging to the same subset. The results (Table III, fourth column) show that this configuration, although it does not yield the best absolute value in any case, produces general improvements in the metric values for all the problems. These improvements are specially captured by the HV metric, which indicates that the algorithm is now able to converge to the optimal Pareto fronts because nonzero values are reached. ZDT4 is the exception yet.

5) Experiment 5: This experiment keeps the previous configuration in all the parameter settings (see Table II) except in the solution combination method, where the SBX crossover operator is used instead of linear combinations. This operator is commonly applied in the algorithms NSGA-II and SPEA2, so it is of interest to study its application in AbYSS. The results (Table III, fifth column) show that this configuration is clearly the best out of the six experiments and with statistical confidence (see the "+" symbols in the last column of the table). Indeed, this parameter setting makes AbYSS reach, on the one hand, the lowest GD and Δ values and, on the other hand, the highest HV values for the five MOPs considered.

6) Experiment 6: In Experiment 5, we achieved a configuration that successfully solved the problems considered. We now perform three steps in the improvement method to study whether intensifying the search is capable of improving AbYSS. The obtained results (Table III, last column) are satisfactory, but this configuration does not improve the results of Experiment 5. Increasing the number of steps in this method has two contradictory effects on AbYSS. On the one hand, more accurate solutions are hopefully computed because of the higher number of trials explored but, on the other hand, more improvement steps mean more function evaluations and, therefore, the scatter search iterates fewer times. As learned from Experiments 3 and 4 with the current configurations, this leads to less search intensification since the number of restarts, and thus the feedback of nondominated solutions from the external archive is reduced. Further experiments, which are not presented here in the interest of clarity, point out that using no improvement does not work because no alterations on a single solution are induced and, therefore, the search gets stuck easily (just combining solutions is not enough in our context).

Therefore, we can conclude from these experiments that applying stochastic operators in AbYSS leads the algorithm to be more robust and accurate. However, it is worth mentioning that many other configurations are possible, so there is still room for improvement. To finish this section, we can state that the parameter settings of Experiment 5 are the most promising of the ones tested for AbYSS. Now that we have decided the set of parameters characterizing AbYSS, we are ready to make a more in-depth evaluation of our proposal, including a comparison with other well-known metaheuristic algorithms for solving MOPs.

D. Results

In order to know how competitive AbYSS is, we decided to compare it to two algorithms which represent the state-of-the-art. These algorithms are NSGA-II and SPEA2. Next, we briefly

TABLE VI
MEDIAN AND INTERQUARTILE RANGE OF THE HYPERVOLUME (HV) METRIC (25 000 FUNCTION EVALUATIONS)

Problem	AbYSS $\tilde{x}_{IQR}$	NSGA-II $\tilde{x}_{IQR}$	SPEA2 $\tilde{x}_{IQR}$	
Schaffer	8.299e−01 $_{4.1e-05}$	8.287e−01 $_{3.2e-04}$	8.296e−01 $_{7.4e-05}$	+
Fonseca	3.108e−01 $_{2.7e-04}$	3.064e−01 $_{6.3e-04}$	3.105e−01 $_{3.2e-04}$	+
Kursawe	4.013e−01 $_{1.5e-04}$	3.997e−01 $_{3.5e-04}$	4.009e−01 $_{2.1e-04}$	+
ZDT1	6.614e−01 $_{3.3e-04}$	6.594e−01 $_{4.5e-04}$	6.601e−01 $_{4.0e-04}$	+
ZDT2	3.281e−01 $_{3.7e-04}$	3.262e−01 $_{4.6e-04}$	3.267e−01 $_{5.1e-04}$	+
ZDT3	5.158e−01 $_{3.4e-03}$	5.148e−01 $_{2.0e-04}$	5.140e−01 $_{4.0e-04}$	+
ZDT4	6.553e−01 $_{6.1e-03}$	6.555e−01 $_{5.5e-03}$	1.068e−01 $_{2.1e-01}$	+
ZDT6	4.004e−01 $_{1.8e-04}$	3.860e−01 $_{1.6e-03}$	3.926e−01 $_{1.1e-03}$	+
WFG1	2.550e−01 $_{1.4e-01}$	2.420e−01 $_{9.3e-02}$	5.750e−02 $_{2.5e-02}$	+
WFG2	5.611e−01 $_{8.2e-04}$	5.612e−01 $_{2.5e-03}$	6.479e−02 $_{7.1e-02}$	+
WFG3	4.419e−01 $_{3.4e-04}$	4.407e−01 $_{3.3e-04}$	1.166e−01 $_{4.8e-05}$	+
WFG4	2.190e−01 $_{2.9e-04}$	2.172e−01 $_{4.1e-04}$	4.928e−02 $_{1.7e-04}$	+
WFG5	1.962e−01 $_{6.7e-05}$	1.950e−01 $_{4.4e-04}$	1.964e−01 $_{9.4e-05}$	+
WFG6	1.696e−01 $_{3.9e-02}$	2.032e−01 $_{9.7e-03}$	5.974e−02 $_{5.4e-04}$	+
WFG7	2.107e−01 $_{1.1e-04}$	2.089e−01 $_{3.9e-04}$	4.634e−03 $_{5.0e-05}$	+
WFG8	1.437e−01 $_{3.0e-03}$	1.471e−01 $_{2.9e-03}$	7.745e−02 $_{4.3e-04}$	+
WFG9	2.379e−01 $_{3.5e-03}$	2.375e−01 $_{2.6e-03}$	6.767e−02 $_{1.6e-03}$	+
ConstrEx	7.763e−01 $_{3.2e-04}$	7.746e−01 $_{3.8e-04}$	7.751e−01 $_{4.7e-04}$	+
Srinivas	5.407e−01 $_{8.8e-05}$	5.383e−01 $_{4.7e-04}$	5.400e−01 $_{2.1e-04}$	+
Osyczka2	7.441e−01 $_{3.0e-01}$	7.454e−01 $_{7.7e-03}$	7.238e−01 $_{2.1e-02}$	+
Golinski	9.694e−01 $_{1.2e-04}$	9.690e−01 $_{2.2e-04}$	9.667e−01 $_{6.6e-04}$	+
Tanaka	3.078e−01 $_{4.5e-04}$	3.075e−01 $_{4.7e-04}$	3.088e−01 $_{3.0e-04}$	+
Viennet2	9.222e−01 $_{9.4e-04}$	9.207e−01 $_{1.9e-03}$	9.253e−01 $_{6.1e-04}$	+
Viennet3	8.341e−01 $_{1.1e-03}$	8.337e−01 $_{6.8e-04}$	8.267e−01 $_{1.4e-03}$	+
Viennet4	8.563e−01 $_{2.7e-03}$	8.573e−01 $_{2.9e-03}$	8.637e−01 $_{1.0e-03}$	+
Water	4.188e−01 $_{6.1e-03}$	4.098e−01 $_{7.9e-03}$	4.038e−01 $_{7.5e-03}$	+
DTLZ1	7.342e−01 $_{6.8e-01}$	7.412e−01 $_{1.8e-02}$	5.842e−01 $_{6.5e-01}$	+
DTLZ2	3.837e−01 $_{5.7e-03}$	3.770e−01 $_{6.9e-03}$	3.938e−01 $_{3.9e-03}$	+
DTLZ3	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	-
DTLZ4	3.881e−01 $_{5.7e-03}$	3.771e−01 $_{9.1e-03}$	3.869e−01 $_{1.8e-01}$	+
DTLZ5	9.399e−02 $_{4.0e-05}$	9.371e−02 $_{2.5e-04}$	9.334e−02 $_{3.3e-04}$	+
DTLZ6	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	0.000e+00 $_{0.0e+00}$	-
DTLZ7	2.606e−01 $_{3.1e-02}$	2.840e−01 $_{4.6e-03}$	2.838e−01 $_{6.2e-03}$	+

describe these algorithms, and the parameter settings used in the subsequent experiments.

- **Nondominated Sorting Genetic Algorithm II**: The NSGA-II algorithm was proposed by Deb *et al.* [4]. It is based on obtaining a new population from the original one by 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.[1] We have used the real-coded version and the parameter settings used in [4]. The operators for crossover and mutation are SBX and polynomial mutation, with distribution indexes of $\eta_c = 20$ and $\eta_m = 20$, respectively. 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 population size is 100 individuals.
- **Strength Pareto EA**: SPEA2 was proposed by Zitler *et al.* in [5]. In this algorithm, each individual has a fitness value assigned which is the sum of its strength raw fitness and a density estimation (see Section III-B3). 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 the population size, a truncation 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.[2] The algorithm is implemented within the PISA framework [41]. However, the implementation of SPEA2 does not contain constraint-handling management, so we were forced to modify the original implementation in order to include the same constraint mechanism used in NSGA-II and AbYSS. We have used the following values for the parameters: both the population and the archive have a size of 100 individuals, and the crossover and mutation operators are the same as those 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 has been implemented in Java using jMetal, a framework aimed at facilitating the development of metaheuristics for solving multiobjective optimization problems [42]. It can be freely downloaded from: http://neo.lcc.uma.es/metal/index.html.

To evaluate each algorithm, we performed two series of experiments. First, we ran all the approaches for 25 000 function evaluations and then repeated them, this time with the execution of 50 000 function evaluations as the stopping condition. These values have been used in previous works in the area [4]. Test MOPs (see Section IV-A) have been grouped into five cat-

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

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

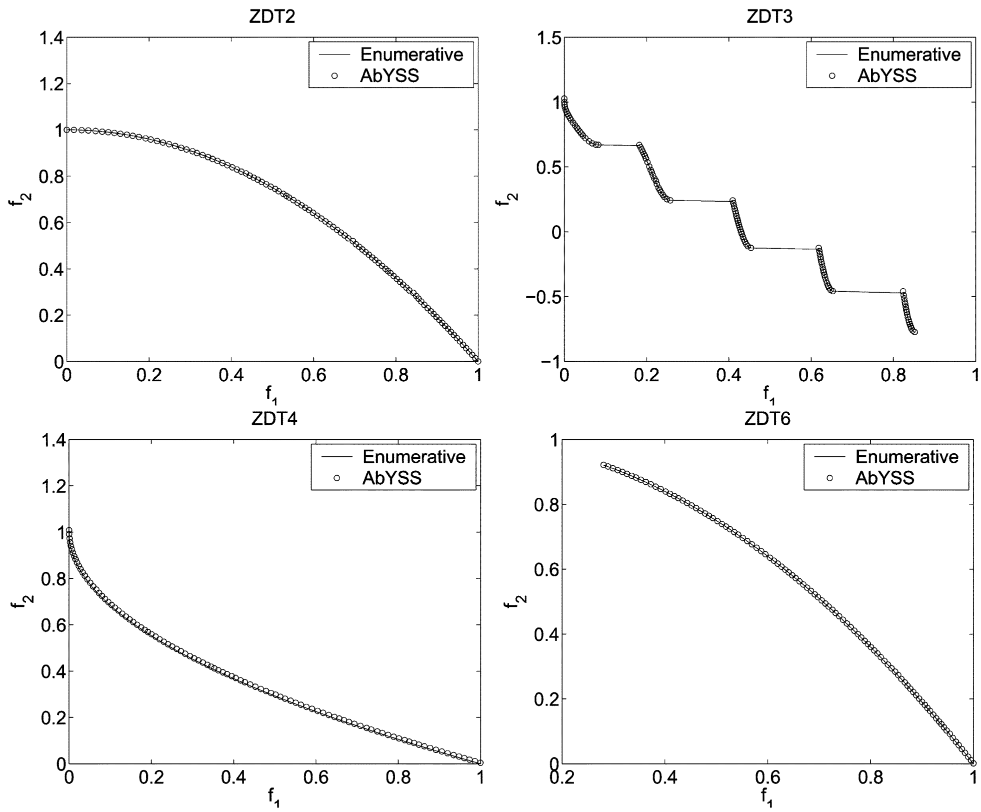

Fig. 10. Nondominated solutions with AbYSS on problems ZDT2, ZDT3, ZDT4, and ZDT6.

egories in the tables for a better presentation of the results. The first group is composed of the biobjective unconstrained problems Schaffer, Fonseca, and Kursawe, as well as the problems ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6. The second group includes the nine WFG problems. The third set embraces the constrained biobjective problems Constr_Ex, Srinivas, Osyczka2, Golinski, and Tanaka. The fourth group comprises the problems Viennet2, Viennet3, Viennet4, and Water. Finally, the last group is the family of scalable problems DLTZ1-7.

For each problem, we have executed 100 independent runs. The values included in the tables of results are the median, $\tilde{x}$, and the interquartile range, IQR. The best ones for each problem have a gray background. The same statistical analysis as in the previous section has been performed here.

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

Table IV shows that AbYSS obtains the lowest (best) values for the GD metric in 21 out of the 33 test MOPs, and with statistical confidence in almost all the cases (see "+" symbols in the last column of the table). This means that the resulting Pareto fronts from AbYSS are closer to the optimal Pareto fronts than those computed by NSGA-II and SPEA2. We would like to highlight the suitability of AbYSS for solving the first two groups of MOPs (biobjective ones), in which this algorithm obtains the closest fronts for 13 out of 17 problems. This is particularly relevant in the WFG family, where an advanced algorithm like SPEA2 reports GD values one or two orders of magnitude worse than AbYSS (WFG1 and WFG8 stand for the exceptions). In the three remaining groups of problems (biobjective constrained and more-than-two objectives), AbYSS obtains competitive results compared with NSGA-II and SPEA2. Particular accuracy is achieved in Srinivas, DTLZ2, and DTLZ3. Finally, we wish to point out that the differences in terms of raw (absolute) distances are, with some exceptions, not very large (close-to-zero values in the GD metric), thus indicating a similar ability of the three algorithms to compute accurate Pareto fronts. The reported results of Table IV have statistical confidence for all the problems, except for Viennet2.

The results obtained from the Spread metric (see Table V) indicate that AbYSS outperforms the other two algorithms concerning the diversity of the obtained Pareto fronts in the biobjective MOPs (it yields the lowest values in 21 of the 22 biobjective problems). The point here is that not only differences in the Δ values between AbYSS and NSGA-II/SPEA2 are, in general, greater than in the GD metric, but also these differences lead to Pareto fronts in which nondominated solutions are much better spread. To graphically illustrate this fact, we show in Fig. 8 typical simulation results of the three algorithms when solving the Kursawe problem. We can observe that, although SPEA2 and NSGA-II obtain a good spreadout, the set of nondominated solutions generated by AbYSS achieves an almost perfect distribution along the Pareto front. To reinforce this claim, we also include the fronts obtained when solving the problem ZDT1 in Fig. 9. If we consider MOPs with more than two objectives, SPEA2 provides the best values in 6 out of the 11 analyzed problems, while AbYSS performs best in four MOPs. Analyzing these results in detail, we observe that AbYSS is the only algorithm

TABLE VII
MEDIAN AND INTERQUARTILE RANGE OF THE GENERATIONAL DISTANCE (GD) METRIC (50 000 FUNCTION EVALUATIONS)

Problem	AbYSS $\tilde{x}_{IQR}$	NSGA-II $\tilde{x}_{IQR}$	SPEA2 $\tilde{x}_{IQR}$	
Schaffer	2.321e−04 $_{2.1e-05}$	2.318e−04 $_{1.7e-05}$	2.393e−04 $_{1.1e-05}$	+
Fonseca	1.647e−04 $_{1.5e-05}$	4.598e−04 $_{5.0e-05}$	2.216e−04 $_{2.6e-05}$	+
Kursawe	1.328e−04 $_{1.1e-05}$	2.080e−04 $_{3.8e-05}$	1.588e−04 $_{2.2e-05}$	+
ZDT1	1.509e−04 $_{2.3e-05}$	1.881e−04 $_{4.3e-05}$	1.601e−04 $_{1.7e-05}$	+
ZDT2	5.000e−05 $_{7.4e-06}$	1.208e−04 $_{5.1e-05}$	4.854e−05 $_{4.6e-06}$	+
ZDT3	2.016e−04 $_{9.5e-06}$	2.099e−04 $_{1.8e-05}$	2.110e−04 $_{1.5e-05}$	+
ZDT4	2.386e−04 $_{1.2e-04}$	1.806e−04 $_{7.6e-05}$	5.774e−02 $_{4.9e-02}$	+
ZDT6	5.424e−04 $_{2.0e-05}$	5.624e−04 $_{5.3e-05}$	5.477e−04 $_{2.5e-05}$	+
WFG1	2.323e−02 $_{1.3e-02}$	2.238e−02 $_{9.1e-03}$	4.915e−02 $_{9.7e-05}$	+
WFG2	4.352e−04 $_{1.8e-05}$	4.542e−04 $_{1.3e-04}$	2.382e−02 $_{5.0e-04}$	+
WFG3	1.389e−04 $_{1.7e-05}$	1.693e−04 $_{2.4e-05}$	1.657e−02 $_{4.0e-06}$	+
WFG4	6.348e−04 $_{2.1e-05}$	6.565e−04 $_{6.6e-05}$	2.032e−02 $_{8.2e-05}$	+
WFG5	2.635e−03 $_{1.2e-05}$	2.652e−03 $_{2.3e-05}$	2.668e−03 $_{1.5e-05}$	+
WFG6	2.537e−03 $_{3.4e-03}$	5.963e−04 $_{5.9e-04}$	1.365e−02 $_{9.7e-05}$	+
WFG7	3.028e−04 $_{2.1e-05}$	3.344e−04 $_{5.1e-05}$	3.677e−02 $_{3.2e-04}$	+
WFG8	1.455e−02 $_{4.9e-03}$	1.489e−02 $_{5.4e-03}$	3.735e−03 $_{1.4e-04}$	+
WFG9	1.057e−03 $_{5.0e-05}$	1.107e−03 $_{1.4e-04}$	1.943e−02 $_{6.2e-04}$	+
ConstrEx	1.539e−04 $_{2.3e-05}$	2.847e−04 $_{5.3e-05}$	2.013e−04 $_{3.5e-05}$	+
Srinivas	4.085e−05 $_{7.5e-06}$	1.798e−04 $_{5.2e-05}$	1.170e−04 $_{3.0e-05}$	+
Osyczka2	1.102e−03 $_{2.4e-03}$	1.014e−03 $_{8.2e-05}$	1.346e−03 $_{8.8e-05}$	+
Golinski	3.058e−04 $_{2.7e-05}$	3.271e−04 $_{3.0e-05}$	2.383e−04 $_{7.5e-05}$	+
Tanaka	8.017e−04 $_{6.7e-05}$	1.227e−03 $_{8.5e-05}$	7.805e−04 $_{6.8e-05}$	+
Viennet2	8.209e−04 $_{4.0e-04}$	7.655e−04 $_{5.1e-04}$	8.606e−04 $_{1.8e-04}$	-
Viennet3	2.353e−04 $_{1.7e-04}$	2.152e−04 $_{7.1e-05}$	2.741e−04 $_{8.9e-05}$	+
Viennet4	3.180e−04 $_{8.3e-05}$	4.690e−04 $_{2.1e-04}$	4.824e−04 $_{1.6e-04}$	+
Water	8.426e−03 $_{2.5e-03}$	6.181e−03 $_{1.0e-03}$	1.697e−02 $_{2.3e-03}$	+
DTLZ1	6.137e−04 $_{8.1e-05}$	5.803e−04 $_{2.6e-04}$	1.132e−01 $_{5.1e-01}$	+
DTLZ2	7.038e−04 $_{4.9e-05}$	1.244e−03 $_{2.3e-04}$	1.767e−03 $_{4.1e-04}$	+
DTLZ3	4.545e−03 $_{1.0e-01}$	4.692e−03 $_{1.6e-02}$	2.721e+00 $_{3.0e+00}$	+
DTLZ4	4.970e−03 $_{2.8e-04}$	4.477e−03 $_{4.4e-04}$	5.158e−03 $_{1.7e-03}$	+
DTLZ5	2.500e−04 $_{3.8e-05}$	6.359e−04 $_{7.3e-05}$	6.795e−04 $_{5.5e-05}$	+
DTLZ6	1.106e−02 $_{6.3e-03}$	1.199e−02 $_{5.5e-03}$	1.874e−02 $_{1.6e-03}$	+
DTLZ7	2.021e−03 $_{9.8e-04}$	2.475e−03 $_{4.3e-04}$	2.425e−03 $_{5.4e-04}$	+

TABLE VIII
MEDIAN AND INTERQUARTILE RANGE OF THE SPREAD METRIC (50 000 FUNCTION EVALUATIONS)

Problem	AbYSS $\tilde{x}_{IQR}$	NSGA-II $\tilde{x}_{IQR}$	SPEA2 $\tilde{x}_{IQR}$	
Schaffer	1.074e−01 $_{3.1e-02}$	4.246e−01 $_{6.3e-02}$	1.287e−01 $_{1.7e-02}$	+
Fonseca	8.230e−02 $_{2.0e-02}$	3.820e−01 $_{6.0e-02}$	1.278e−01 $_{2.0e-02}$	+
Kursawe	2.274e−01 $_{4.8e-03}$	5.445e−01 $_{1.0e-01}$	2.980e−01 $_{2.0e-02}$	+
ZDT1	7.632e−02 $_{2.2e-02}$	3.815e−01 $_{7.2e-02}$	1.414e−01 $_{1.8e-02}$	+
ZDT2	7.728e−02 $_{1.4e-02}$	4.101e−01 $_{6.9e-02}$	1.472e−01 $_{3.0e-02}$	+
ZDT3	1.019e−01 $_{2.5e-02}$	3.745e−01 $_{5.4e-02}$	1.619e−01 $_{2.7e-02}$	+
ZDT4	8.657e−02 $_{1.6e-02}$	4.282e−01 $_{7.5e-02}$	4.585e−01 $_{1.6e-01}$	+
ZDT6	5.749e−02 $_{1.4e-02}$	5.054e−01 $_{6.5e-02}$	1.418e−01 $_{1.9e-02}$	+
WFG1	4.585e−01 $_{1.1e-01}$	6.449e−01 $_{7.1e-02}$	5.792e−01 $_{1.4e-02}$	+
WFG2	3.543e−01 $_{2.3e-02}$	5.606e−01 $_{1.9e-01}$	8.643e−01 $_{6.6e-02}$	+
WFG3	3.638e−01 $_{9.5e-03}$	6.119e−01 $_{4.1e-02}$	8.204e−01 $_{5.0e-03}$	+
WFG4	1.077e−01 $_{1.9e-02}$	3.654e−01 $_{5.5e-02}$	4.294e−01 $_{2.0e-02}$	+
WFG5	1.282e−01 $_{1.8e-02}$	3.920e−01 $_{4.6e-02}$	3.601e−01 $_{2.8e-02}$	+
WFG6	1.321e−01 $_{4.6e-02}$	3.681e−01 $_{6.1e-02}$	6.558e−01 $_{1.1e-02}$	+
WFG7	1.102e−01 $_{2.3e-02}$	3.828e−01 $_{6.3e-02}$	5.483e−01 $_{1.7e-02}$	+
WFG8	4.184e−01 $_{5.9e-02}$	5.463e−01 $_{5.0e-02}$	8.022e−01 $_{1.4e-02}$	+
WFG9	1.018e−01 $_{2.3e-02}$	3.620e−01 $_{6.5e-02}$	4.410e−01 $_{5.3e-02}$	+
ConstrEx	1.368e−01 $_{1.9e-02}$	4.194e−01 $_{5.9e-02}$	5.085e−01 $_{5.0e-02}$	+
Srinivas	6.841e−02 $_{1.3e-02}$	3.496e−01 $_{5.8e-02}$	1.521e−01 $_{2.1e-02}$	+
Osyczka2	3.362e−01 $_{9.3e-02}$	6.663e−01 $_{1.2e-01}$	5.529e−01 $_{5.9e-02}$	+
Golinski	9.581e−02 $_{2.0e-02}$	5.101e−01 $_{1.1e-01}$	7.069e−01 $_{5.8e-02}$	+
Tanaka	2.353e−01 $_{3.5e-02}$	4.226e−01 $_{6.9e-02}$	1.881e−01 $_{2.7e-02}$	+
Viennet2	2.576e−01 $_{5.0e-02}$	4.510e−01 $_{5.8e-02}$	9.650e−01 $_{1.1e-03}$	+
Viennet3	2.241e−01 $_{5.8e-02}$	4.598e−01 $_{6.8e-02}$	9.025e−01 $_{3.6e-03}$	+
Viennet4	4.219e−01 $_{4.4e-02}$	5.756e−01 $_{4.8e-02}$	8.672e−01 $_{4.7e-03}$	+
Water	4.690e−01 $_{5.2e-02}$	4.876e−01 $_{7.2e-02}$	4.924e−01 $_{5.7e-02}$	+
DTLZ1	4.374e−01 $_{5.5e-02}$	5.731e−01 $_{8.1e-02}$	7.130e−01 $_{1.2e+00}$	+
DTLZ2	4.972e−01 $_{4.2e-02}$	5.260e−01 $_{5.1e-02}$	1.054e−01 $_{1.5e-02}$	+
DTLZ3	5.949e−01 $_{1.1e-01}$	5.706e−01 $_{1.4e-01}$	1.357e+00 $_{4.7e-01}$	+
DTLZ4	4.458e−01 $_{5.3e-02}$	4.915e−01 $_{6.1e-02}$	1.109e−01 $_{4.5e-01}$	+
DTLZ5	1.417e−01 $_{1.7e-02}$	4.450e−01 $_{7.1e-02}$	1.800e−01 $_{2.4e-02}$	+
DTLZ6	3.991e−01 $_{9.0e-02}$	5.853e−01 $_{9.3e-02}$	2.081e−01 $_{2.2e-02}$	+
DTLZ7	5.442e−01 $_{1.0e-01}$	5.104e−01 $_{7.0e-02}$	2.585e−01 $_{3.4e-02}$	+

reporting values in the order of $e{-}01$ in all the problems; SPEA2 performance in problems DTLZ1 and DTLZ3 is in the order of $e-00$. So, AbYSS appears to be more robust concerning these MOPs.

As a measure of both convergence and diversity, the HV metric should prove the results of the two other metrics. This is true, in general, if we observe the results in the three first groups of problems, where AbYSS obtains the best (highest)

TABLE IX
MEDIAN AND INTERQUARTILE RANGE OF THE HYPERVOLUME (HV) METRIC (50 000 FUNCTION EVALUATIONS)

Problem	AbYSS $\tilde{x}_{IQR}$	NSGA-II $\tilde{x}_{IQR}$	SPEA2 $\tilde{x}_{IQR}$	
Schaffer	8.299e−01 $_{4.3e-05}$	8.287e−01 $_{3.4e-04}$	8.296e−01 $_{9.4e-05}$	+
Fonseca	3.114e−01 $_{2.0e-04}$	3.064e−01 $_{6.2e-04}$	3.106e−01 $_{3.0e-04}$	+
Kursawe	4.015e−01 $_{8.1e-05}$	3.998e−01 $_{3.2e-04}$	4.010e−01 $_{2.2e-04}$	+
ZDT1	6.620e−01 $_{3.8e-05}$	6.604e−01 $_{3.9e-04}$	6.615e−01 $_{9.3e-05}$	+
ZDT2	3.287e−01 $_{6.3e-05}$	3.273e−01 $_{3.1e-04}$	3.283e−01 $_{1.4e-04}$	+
ZDT3	5.160e−01 $_{2.7e-05}$	5.155e−01 $_{1.3e-04}$	5.156e−01 $_{1.0e-04}$	+
ZDT4	6.598e−01 $_{2.1e-03}$	6.595e−01 $_{1.8e-03}$	1.068e−01 $_{2.1e-01}$	+
ZDT6	4.007e−01 $_{1.4e-04}$	3.943e−01 $_{3.2e-04}$	3.994e−01 $_{2.9e-04}$	+
ConstrEx	7.768e−01 $_{1.8e-04}$	7.745e−01 $_{3.8e-04}$	7.752e−01 $_{4.4e-04}$	+
Srinivas	5.408e−01 $_{6.8e-05}$	5.383e−01 $_{4.2e-04}$	5.400e−01 $_{2.1e-04}$	+
Osyczka2	7.459e−01 $_{1.3e-01}$	7.523e−01 $_{7.8e-03}$	7.345e−01 $_{2.5e-02}$	+
Golinski	9.695e−01 $_{7.3e-05}$	9.691e−01 $_{2.3e-04}$	9.671e−01 $_{7.0e-04}$	+
Tanaka	3.085e−01 $_{3.0e-04}$	3.078e−01 $_{3.6e-04}$	3.092e−01 $_{1.9e-04}$	+
WFG1	3.143e−01 $_{1.4e-01}$	3.225e−01 $_{9.3e-02}$	9.949e−02 $_{8.5e-05}$	+
WFG2	5.612e−01 $_{6.9e-04}$	5.613e−01 $_{2.7e-03}$	1.001e−01 $_{7.1e-02}$	+
WFG3	4.419e−01 $_{3.2e-04}$	4.410e−01 $_{3.1e-04}$	1.166e−01 $_{2.8e-05}$	+
WFG4	2.193e−01 $_{5.3e-05}$	2.175e−01 $_{4.6e-04}$	4.929e−02 $_{1.3e-04}$	+
WFG5	1.962e−01 $_{6.2e-05}$	1.950e−01 $_{4.0e-04}$	1.964e−01 $_{6.5e-05}$	+
WFG6	1.782e−01 $_{4.0e-02}$	2.029e−01 $_{9.2e-03}$	5.977e−02 $_{3.5e-04}$	+
WFG7	2.108e−01 $_{4.9e-05}$	2.091e−01 $_{3.7e-04}$	4.635e−03 $_{7.1e-05}$	+
WFG8	1.472e−01 $_{2.9e-03}$	1.492e−01 $_{1.7e-03}$	7.817e−02 $_{2.8e-04}$	+
WFG9	2.389e−01 $_{3.9e-03}$	2.380e−01 $_{1.3e-03}$	6.657e−02 $_{1.6e-03}$	+
Viennet2	9.221e−01 $_{9.9e-04}$	9.206e−01 $_{1.8e-03}$	9.252e−01 $_{5.7e-04}$	+
Viennet3	8.342e−01 $_{8.9e-04}$	8.336e−01 $_{7.9e-04}$	8.270e−01 $_{9.8e-04}$	+
Viennet4	8.563e−01 $_{2.3e-03}$	8.576e−01 $_{2.7e-03}$	8.637e−01 $_{1.2e-03}$	+
Water	4.200e−01 $_{5.9e-03}$	4.106e−01 $_{6.9e-03}$	4.036e−01 $_{7.3e-03}$	+
DTLZ1	7.633e−01 $_{6.9e-03}$	7.381e−01 $_{3.0e-02}$	7.738e−01 $_{1.2e-02}$	+
DTLZ2	3.856e−01 $_{6.4e-03}$	3.772e−01 $_{6.4e-03}$	3.956e−01 $_{3.1e-03}$	+
DTLZ3	3.213e−01 $_{3.6e-01}$	3.394e−01 $_{5.2e-02}$	0.000e+00 $_{0.0e+00}$	-
DTLZ4	3.880e−01 $_{5.6e-03}$	3.776e−01 $_{8.0e-03}$	3.886e−01 $_{1.9e-01}$	+
DTLZ5	9.403e−02 $_{3.6e-05}$	9.376e−02 $_{2.3e-04}$	9.363e−02 $_{2.5e-04}$	+
DTLZ6	4.595e−02 $_{1.8e-02}$	4.961e−02 $_{1.8e-02}$	2.814e−02 $_{3.1e-03}$	+
DTLZ7	2.638e−01 $_{3.3e-02}$	2.873e−01 $_{6.3e-03}$	2.969e−01 $_{4.0e-03}$	+

HV values in 15 out of the 22 problems comprising these groups (Table VI). Indeed, AbYSS is always the best for the greatest number of MOPs in each group (seven out of eight, five out of nine, and three out of five for the first, second, and third group, respectively). When we shift to more-than-two objective MOPs (last two groups) differences vanish and AbYSS is competitive against SPEA2 and NSGA-II in the fourth and fifth groups of MOPs. If we further analyze the HV values of the DTLZ family, we observe that problems DTLZ3 and DTLZ6 yield a value of zero, which means that their fronts are outside the limits of the true Pareto front of the problems, as we explained in Section IV-C.

With the aim of giving a complete graphical overview of the behavior of AbYSS, we include the fronts it computes for ZDT2, ZDT3, ZDT4, and ZDT6 in Fig. 10. The graphics show that our proposal is particularly accurate on biobjective MOPs.

We now turn to analyze the results obtained when running 50 000 function evaluations. The aim of this set of experiments is to check whether AbYSS can also be competitive and even outperform both NSGA-II and SPEA2 when we allow the algorithms to run for longer.

Concerning GD (Table VII), the results are similar to those reached when computing 25 000 function evaluations. Now, AbYSS is not the best algorithm in MOPs Schaffer, ZDT2, and Viennet2, but it outperforms the other two algorithms in DTLZ6. In general, the three algorithms obtain slightly better results when performing 50 000 function evaluations, but in some cases the metric values are worse. Some examples are ZDT3, DTLZ5, and WFG8 for AbYSS, NSGA-II, and SPEA2, respectively. If we pay attention to the DTLZ problem family, there are noticeable improvements; hence, ABYSS enhances the GD values in all the problems except DTLZ4, and similar behavior is observed in NSGA-II and SPEA2. In DTLZ problems, all the algorithms compute Pareto fronts which are closer to the optimal Pareto front, thus profiting from the deeper exploration they are allowed to carry out. However, the important point here is that AbYSS again outperforms both NSGA-II and SPEA2 in most MOPs of the benchmark used.

In the Spread metric (see Table VIII), AbYSS improves the results in all the problems, but Viennet3, and DTLZ7; however, this does not hold with NSGA-II and SPEA2, which obtain worse values in many problems. It is worth mentioning that AbYSS is the optimizer which major improvements have reached (e.g., the problems ZDT1, ZDT2, ZDT4, and Golinski), computing the best values of the Spread metric for 27 out of 33 benchmarking MOPs.

Finally, the HV values included in Table IX confirm that making more function evaluations improves the obtained Pareto fronts in most of the experiments. As in the case of 25 000 function evaluations, the HV metric does not allow to decide a clear winner when solving the four proposed set of problems, because, although SPEA2 obtains the best HV values in four out of the seven DTLZ problems, it fails in DTLZ3.

A general conclusion that can be drawn from the two sets of experiments is that AbYSS provides the best values in convergence in most of the problems, and it also outperforms both NSGA-II and SPEA2 in terms of diversity when solving biobjective MOPs. The quality of the fronts in these problems is remarkable, as is pointed out by both the values of the three metrics and Figs. 8–10.

V. Conclusion and Future Work

We have presented a proposal to adapt the scatter search method to handle MOPs. The proposed algorithm, AbYSS, is a

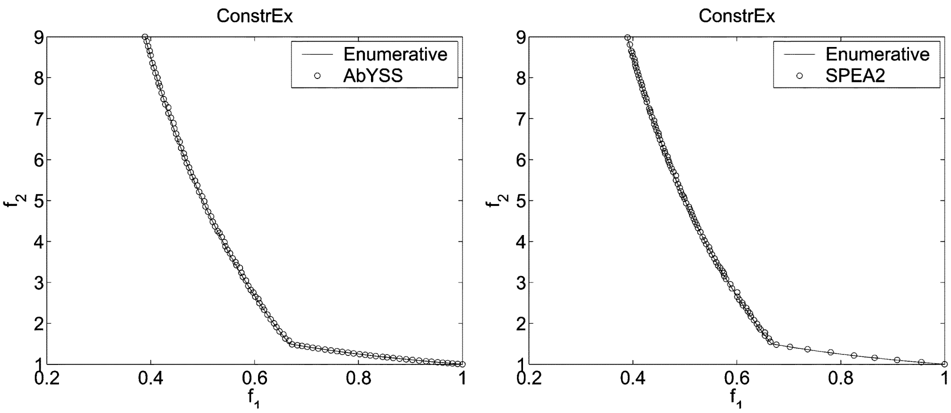

Fig. 11. Pareto fronts of the problem Constr_Ex with AbYSS and SPEA2.

hybridized scatter search which uses an external archive to store the nondominated solutions found during the search. Salient features of AbYSS are the feedback of individuals from the archive to the initial set in the restart 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 taken from the archive to feed the initial set in the restart. On the other hand, the density estimator used in SPEA2 obtains 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 with two state-of-the-art multiobjective optimization algorithms, NSGA-II and SPEA2; for that purpose, 33 test problems, including unconstrained and constrained ones with two or more objectives, were chosen and three metrics were used to assess the performance of the algorithms. The results reveal that AbYSS outperforms all the proposals on most test problems considered when using 25 000 and 50 000 function evaluations according to the Spread metric, and it obtains very competitive fronts concerning both GD and HV metrics. Since the operators which guide the search in the three algorithms are the same (SBX and polynomial mutation), we can state that the search engine of the scatter search approach is responsible for the improvements on NSGA-II and SPEA2.

An in-depth study of the parameters defining the behavior of the algorithm, as well as its application to solve real-world problems are the focus of future work. In this sense, we intend to use AbYSS to solve combinatorial problems in the telecommunications field.

APPENDIX
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 [4], the metric can give misleading results if we compare two fronts and the two objective functions range between values of different order of magnitude. We observe this behavior when comparing the fronts of problem Constr_Ex produced by the algorithms AbYSS and SPEA2.

As can be seen in Fig. 11, the Pareto front computed by AbYSS achieves a better spread that the one obtained by SPEA2 when making 25 000 function evaluations; after applying the Spread metric, the values reported are 0.5112 for AbYSS and 0.1827 in the case of SPEA2. Thus, according to the metric, SPEA2 is better than AbYSS on this problem.

If we observe the Pareto front of problem Constr_Ex, 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 spread among the two parts. As the two sets of solutions are composed of the same number of points (100 solutions), the Spread metric favors SPEA2 because the distances measured in the lower front are negligible compared with those of the upper 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 is kept identical, and the results of applying the Spread metric yield 0.16 to AbYSS and 0.5166 to SPEA2 (see Table V).

REFERENCES

[1] C. Coello, D. Van Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, ser. Genetic Algorithms and Evolutionary Computation. Norwell, MA: Kluwer Academic, 2002.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.

[3] F. W. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, ser. Int. Series in Operations Research and Management Sciences. Norwell, MA: Kluwer Academic, 2003.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, 2002.

[5] E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm Comput. Eng. Netw. Lab. (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.

[6] J. Knowles and D. Corne, “The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization,” in *Proc. 1999 Congr. Evol. Comput.*, Piscataway, NJ, 1999, pp. 9–105.

[7] F. Glover, *A Template for Scatter Search and Path Relinking*, ser. Lecture Notes in Computer Science, J. Hao, E. Lutton, E. Ronald, M. Shoenauer, and D. Snyers, Eds. Berlin, Germany: Springer-Verlag, 1997, pp. 13–54.

[8] F. Glover, M. Laguna, and R. Martí, “Fundamentals of scatter search and path relinking,” *Contr. Cybern.*, vol. 29, no. 3, pp. 653–684, 2000.

[9] F. Glover, M. Laguna, and R. Martí, "Scatter search," in *Advances in Evolutionary Computing: Theory and Applications*, A. Ghosh and S. Tsutsui, Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 519–537.
[10] R. Martí, H. Lourenço, and M. Laguna, "Assigning proctors to exams with scatter search," in *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna and J. L. González-Velarde, Eds. Norwell, MA: Kluwer Academic, 2000, pp. 215–227.
[11] A. Corberán, E. Fernández, M. Laguna, and R. Martí, "Heuristic solutions to the problem of routing school buses with multiple objectives," *J. Oper. Res. Soc.*, vol. 53, no. 4, pp. 427–435, 2002.
[12] A. R. M. Rao and N. Arvind, "A scatter search algorithm for stacking sequence optimisation of laminate composites," *Composite Structures*, vol. 70, no. 4, pp. 383–402, Oct. 2005.
[13] A. Krishna and K. Rao, "Multi-objective optimisation of surface grinding operations using scatter search approach," *Int. J. Adv. Manuf. Technol.*, vol. 29, no. 5, pp. 475–480, 2006.
[14] C. G. da Silva, J. Clímaco, and J. Figueira, "A scatter search method for the bi-criteria multi-dimensional $\{0,1\}$-knapsack problem using surrogate relaxation," *J. Math. Modelling Algorithms*, vol. 3, no. 3, pp. 183–208, Jan. 2004.
[15] R. P. Beausoleil, "Moss: Multiobjective scatter search applied to non-linear multiple criteria optimization," *Eur. J. Oper. Res.*, vol. 169, no. 2, pp. 426–449, Mar. 2006.
[16] J. Molina, M. Laguna, R. Martí, and R. Caballero, "SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization," *INFORMS J. Computing*, vol. 19, no. 1, pp. 91–100, 2007.
[17] A. J. Nebro, F. Luna, and E. Alba, C. Coello, A. Hernández, and E. Zitler, Eds., "New ideas in applying scatter search to multiobjective optimization," in *Proc. 3rd Int. Conf. Evol. MultiCriterion Opt., EMO 2005*, 2005, vol. 3410, pp. 443–458.
[18] J. A. Vasconcelos, J. H. R. D. Maciel, and R. O. Parreiras, "Scatter search techniques applied to electromagnetic problems," *IEEE Trans. Magn.*, vol. 4, pp. 1804–1807, 2005.
[19] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1995.
[20] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995.
[21] C. Coello, G. Toscano, and M. Salazar, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–278, Jun. 2004.
[22] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms*, J. Grefensttete, Ed., 1987, pp. 93–100.
[23] C. Fonseca and P. Flemming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 38–47, 1998.
[24] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving for Nature*, H. Schwefel and R. Männer, Eds. Berlin, Germany: Springer-Verlag, 1990, pp. 193–197.
[25] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
[26] S. Huband, L. Barone, R. L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. Evol. Multi-Criterion Opt. (EMO)*, C. Coello, A. Hernández, and E. Zitler, Eds., 2005, vol. 3410, pp. 280–295.
[27] A. Osyczka and S. Kundo, "A new method to solve generalized multicriteria optimization problems using a simple genetic algorithm," *Structural Opt.*, vol. 10, pp. 94–99, 1995.
[28] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, "GA-based decision support system for multicriteria optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 1995, vol. 2, pp. 1556–1561.
[29] A. Kurpati, S. Azarm, and J. Wu, "Constraint handling improvements for multi-objective genetic algorithms," *Structural and Multidisciplinary Optimization*, vol. 23, no. 3, pp. 204–213, 2002.
[30] R. Viennet, C. Fontiex, and I. Marc, "Multicriteria optimization using a genetic algorithm for determining a Pareto set," *J. Syst. Sci.*, vol. 27, no. 2, pp. 255–260, 1996.
[31] T. Ray, K. Tai, and K. Seow, "An evolutionary algorithm for multiobjective optimization," *Eng. Opt.*, vol. 33, no. 3, pp. 399–424, 2001.
[32] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 105–145.
[33] A. J. Nebro, E. Alba, and F. Luna, "Multi-objective optimization using grid computing," *Soft Computing*, vol. 11, no. 6, pp. 531–540, 2007.
[34] D. A. Van Veldhuizen and G. B. Lamont, Multiobjective Evolutionary Algorithm Research: A History and Analysis Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., Wright-Patterson, AFB, OH, Tech. Rep. TR-98-03, 1998.
[35] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *Proc. 2006 IEEE Congr. Evol. Comput.*, 2006, pp. 3234–3241.
[36] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 257–271, 1999.
[37] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.
[38] F. Herrera, M. Lozano, and D. Molina, "Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies," *Eur. J. Oper. Res.*, vol. 169, pp. 450–476, 2006.
[39] J. Demšar, "Statistical comparison of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
[40] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. Boca Raton, FL: CRC Press, 2003.
[41] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "Pisa—A platform and programming language independent interface for search algorithms," in *Proc. Conf. Evol. Multi-Criterion Opt. (EMO 2003)*, 2003, pp. 494–508.
[42] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba, jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, Tech. Rep. ITI-2006-10, 2006.

Antonio J. Nebro received the M.S. and Ph.D. degrees in computer science from the University of Málaga, Málaga, Spain, in 1992 and 1999, respectively.

He is currently an Associate Professor of Computer Science at the University of Malaga. He has coauthored several book chapters, and over 30 papers. His current research interests include the design and implementation of parallel evolutionary algorithms, multiobjective optimization metaheuristics, grid computing applied to metaheuristic techniques, and applications to telecommunications and bioinformatics.

Francisco Luna (S'05) received the Engineering degree in computer science from the University of Málaga, Málaga, Spain, in 2002. He is currently working towards the Ph.D. degree in the design of parallel heterogeneous metaheuristics and their application to solve complex problems in the domain of telecommunications and combinatorial optimization at the University of Málaga.

Enrique Alba received the Ph.D. degree from the University of Málaga, Málaga, on designing and analyzing parallel and distributed genetic algorithms.

He is a Professor of Computer Science at the University of Málaga, Málaga, Spain. He has published many scientific papers in international conferences and journals, as well as holds national and international awards to his research results. His current research interests involve the design and application of evolutionary algorithms, neural networks, and other bio-inspired systems to real problems including telecommunications, combinatorial optimization, and bioinformatics. The main focus of all his work is on parallelism.

Bernabé Dorronsoro (S'05) received the Degree in computer science and the Ph.D. degree from the University of Málaga, Málaga, Spain, in 2002 and 2007, respectively.

At present, he is working in the design of new evolutionary algorithms, specifically on structured algorithms and their application to complex real-world problems in the domains of logistics, telecommunications, bioinformatics, combinatorial, and multiobjective optimization.

Juan J. Durillo (S'08) received the Engineering degree in computer science from the University of Málaga, Málaga, Spain, in 2006. He is currently working towards the Ph.D. degree at the University of Málaga.

His research topics are multiobjective optimization and the design of new models for parallel multiobjective metaheuristics.

Andreas Beham (S'04) received the Diploma degree from the Johannes Kepler University Linz, Linz, Austria, in 2007.

He is a member of the Research Group HEAL (Heuristic and Evolutionary Algorithms Laboratory) since 2004 and works currently as a Research Associate at the Upper Austrian University of Applied Sciences, Hagenberg, Austria. His current research focuses in the field of operational research, production planning, and priority rule generation. His research interests include evolutionary hybrid algorithms and multiobjective optimization.