SPAM: Set Preference Algorithm for Multiobjective Optimization

Eckart Zitzler, Lothar Thiele, and Johannes Bader

Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland
{zitzler,thiele,bader}@tik.ee.ethz.ch

Abstract. This paper pursues the idea of a general multiobjective optimizer that can be flexibly adapted to arbitrary user preferences—assuming that the goal is to approximate the Pareto-optimal set. It proposes the Set Preference Algorithm for Multiobjective Optimization (SPAM) the working principle of which is based on two observations: (i) current multiobjective evolutionary algorithms (MOEAs) can be regarded as hill climbers on set problems and (ii) specific user preferences are often (implicitly) expressed in terms of a binary relation on Pareto set approximations. SPAM realizes a $(1 + 1)$-strategy on the space of Pareto set approximations and can be used with any type of set preference relations, i.e., binary relations that define a total preorder on Pareto set approximations. The experimental results demonstrate for a range of set preference relations that SPAM provides full flexibility with respect to user preferences and is effective in optimizing according to the specified preferences. It thereby offers a new perspective on preference-guided multiobjective search.

1 Motivation

By far most publications within the field of evolutionary multiobjective optimization (EMO) focus on the issue of generating a suitable approximation of the Pareto-optimal set, or Pareto set approximation for short. For instance, the first book on EMO by Kalyanmoy Deb [1] is mainly devoted to techniques of finding multiple trade-off solutions using evolutionary algorithms.

Taking this view, one can state that EMO in general deals with set problems: the search space Ψ consists of all potential Pareto set approximations rather than single solutions, i.e., Ψ is a set of sets. When applying an evolutionary algorithm to the problem of approximating the Pareto-optimal set, the population itself can be regarded as the current Pareto set approximation. The subsequent application of mating selection, variation, and environmental selection heuristically produces a new Pareto set approximation that—in the ideal case—is better than the previous one. In the light of the underlying set problem, the population represents a single element of the search space which is in each iteration replaced by another element of the search space. Consequently, selection and variation can be regarded as a mutation operator on populations resp. sets. Somewhat simplified, one may say that a classical multiobjective evolutionary algorithm

G. Rudolph et al. (Eds.): PPSN X, LNCS 5199, pp. 847–858, 2008.

(MOEA) used to approximate the Pareto-optimal set is a $(1, 1)$-strategy on a set problem. Furthermore, MOEAs are usually not preference-free. The main advantage of generating methods such as MOEAs is that the objectives do not need to be aggregated or ranked a priori; but nevertheless preference information is required to guide the search, although it is usually weaker and less stringent. In the environmental selection step, for instance, an MOEA has to choose a subset of individuals from the parents and the offspring which constitutes the next Pareto set approximation. To this end, the algorithm needs to know the criteria according to which the subset should be selected, in particular when all parents and children are incomparable, i.e., mutually nondominating. That means the generation of a new population usually relies on set preference information.

These observations led to the concept presented in this paper which separates preference information and search method. Firstly, we regard preference information as an appropriate order on Ψ required to fully specifiy the set problem—this order will here be denoted as set preference relation. A set preference relation provides the information on the basis of which the search is carried out; for any two Pareto set approximations, it says whether one set is better or not. Secondly, we propose a general, extended $(1+1)$-strategy for this set problem which is only based on pairwise comparisons of sets in order to guide the search. The resulting algorithm (SPAM) is fully independent of the set preference relation used and thereby decoupled from the user preferences.

This complete separation of concerns is the novelty of the suggested approach. It builds upon the idea presented in [2], but is is more general—as it is not restricted to a single binary quality indicator—and possess in addition desirable convergence properties. Furthermore, there are various studies that focus on the issue of preference articulation in EMO, in particular integrating additional preferences such as priorities, goals, and reference points [3,4,5,6,7,8,9]. However, these studies mainly cover preferences on solutions and not preferences on sets, and the search procedures used are based on hard-coded preferences.

In the following, we first discuss the issue of designing set preference relations and then present the full SPAM method. Finally, simulation results are provided and compared for several example set preference relations.

2 Set Preference Relations

Consider a multiobjective optimization problem with the decision space X, the objective space Z, n objectives $f_1, \ldots, f_n$ to be minimized, and a relation $\leq$ on Z, which induces a preference relation $\preceq$ on X with $a \preceq b :\Leftrightarrow f(a) \leq f(b)$ for $a, b \in X$. This problem is transformed into a corresponding set problem where the search space Ψ includes all possible solution sets $A \subseteq X$, i.e., $\Psi = 2^X$. The preference relation $\preceq$ can be used to define a corresponding set preference relation $\preccurlyeq$ on Ψ where

$$A \preccurlyeq B :\Leftrightarrow \forall b \in B \, \exists a \in A : \; a \preceq b$$

for all Pareto set approximations $A, B \in \Psi$. Here, we will assume that weak Pareto dominance, represented by $\preceq_{\text{par}}$ and $\preccurlyeq_{\text{par}}$, is the underlying preference relation resp. set preference relation. Most existing MOEAs are designed for such a type of set problem where the goal is to find a good Pareto set approximation $A \in \Psi$.

The set preference relation deduced from the preference relation on solutions is usually not total, i.e., there are incomparable Pareto set approximations which are hard to deal with by any optimization method. Therefore, additional preferences are needed to refine $\preccurlyeq$ such that no incomparable pairs remain. Next, we will discuss principles to design set preference relation that represent total preorders and then provide several example relations.

2.1 Refinements and Sequences

Unary quality indicators are a possible means to construct set preference relations that are total preorders. They represent set quality measures that map each set $A \in \Psi$ to a real number $I(A) \in \mathbb{R}$. Given an indicator I, one can define the corresponding set preference relation as $A \preccurlyeq_{\mathrm{I}} B := (I(A) \leq I(B))$ where we assume that smaller indicator values stand for higher quality, in other words, A is as least as good as B if the indicator value of A is not larger than the one of B. For instance, several recent approaches make implicitly use of the unary hypervolume indicator in this way [10,11,12]. Alternatively, one may consider binary quality indicators that assign a real value to ordered pairs of sets (A, B) with $A, B \in \Psi$. Assuming that smaller indicator values stand for higher quality, a corresponding set preference relation can be defined as $A \preccurlyeq_{\mathrm{I}} B := (I(A, B) \leq I(B, A))$. For instance, IBEA [2] uses this type of preference information.

When defining set preference relations based on indicators (or using other principles), we would like to guarantee that weak Pareto dominance is not violated, i.e., $\preccurlyeq_{\mathrm{I}}$ should refine $\preccurlyeq_{\mathrm{par}}$. This can be formalized as follows.

Definition 2.1. *Given a set Ψ. Then the preference relation $\preccurlyeq_{ref}$ refines $\preccurlyeq$ if for all $A, B \in \Psi$ we have*

$$(A \preccurlyeq B) \wedge (B \not\preccurlyeq A) \Rightarrow (A \preccurlyeq_{ref} B) \wedge (B \not\preccurlyeq_{ref} A)$$

That means a set that is strictly better than another set in the original set preference relation should remain strictly better in the refined relation. The hypervolume indicator [13,12], for instance, induces a refinement of weak Pareto dominance, cf. [14,15]. Many other indicators only fulfill a weaker property which we here denote as weak refinement.

Definition 2.2. *Given a set Ψ. Then the set preference relation $\preccurlyeq_{ref}$ weakly refines $\preccurlyeq$ if for all $A, B \in \Psi$ we have*

$$(A \preccurlyeq B) \wedge (B \not\preccurlyeq A) \Rightarrow (A \preccurlyeq_{ref} B)$$

A weak refinement may make two sets A and B indifferent ($A \preccurlyeq_{\mathrm{ref}} B \ \wedge \ B \preccurlyeq_{\mathrm{ref}} A$), although A is actually strictly better than B; this is for instance the case for the unary epsilon indicator [15]. Nevertheless, a weak refinement never contradicts the original order, i.e., B cannot be strictly preferable to A with regard to

$\preccurlyeq_{\text{ref}}$, whenever A is strictly better than B regarding $\preccurlyeq$. However, many practically interesting indicators do not induce a weak refinement of the weak Pareto dominance relation.

For optimization purposes, it is desirable to have a set preference relation that represents a refinement of the dominance relation because this is a prerequisite to achieve convergence to the Pareto-optimal set, see [16]. The following construction shows how such refinements can be defined on the basis of arbitrary indicators; it resembles the concept of hierarchy used in [3] for pairs of solutions, but here (a) we are dealing with preference relations on sets and (b) the hierarchical construction is different.

Definition 2.3. *Given a set Ψ and a sequence S of k preference relations over Ψ with $S = (\preccurlyeq^1, \preccurlyeq^2, \ldots, \preccurlyeq^k)$. Then the preference relation $\preccurlyeq_S$ associated with S is defined as follows: Let $A, B \in \Psi$. Then $A \preccurlyeq_S B$ if and only if $\exists 1 \leq i \leq k$ such that the following two conditions are satisfied:*

$$\text{(i)}\ (i < k\ \wedge\ (A \preccurlyeq^i B \wedge B \not\preccurlyeq^i A)) \vee (i = k\ \wedge\ (A \preccurlyeq^k B))$$
$$\text{(ii)}\ \forall 1 \leq j < i : (A \preccurlyeq^j B\ \wedge\ B \preccurlyeq^j A)$$

With this definition, we can derive the following procedure to determine $A \preccurlyeq_{\text{S}} B$ for two sets A and B:

- Start from the first preference relation, i.e. $j = 1$. Repeat the following step: If A and B are indifferent with respect to $\preccurlyeq^j$, then increase j to point to the next relation in the sequence if it exists.
- If the final j points to the last preference relation ($j = k$), then set $A \preccurlyeq_{\text{S}} B \Leftrightarrow A \preccurlyeq^k B$. Otherwise, set $A \preccurlyeq_{\text{S}} B \Leftrightarrow A \prec^k B$.

This procedure allows to use indicators inducing only weak refinements or no refinements at all in combination with refinements; the resulting set preference relation is again a refinement.

Theorem 2.4. *Given a sequence of preference relations according to Def. 2.3. Suppose there is a $k' \leq k$ such that $\preccurlyeq^{k'}$ is a refinement of a given preference relation $\preccurlyeq$. In addition, all relations $\preccurlyeq^j$, $1 \leq j < k'$ are weak refinements of $\preccurlyeq$ and all relations $\preccurlyeq^j$, $k' < j \leq k$ are preorders. Then $\preccurlyeq_S$ is a refinement of $\preccurlyeq$.*

For reasons of space limitations, the proof for this theorem is omitted here; it is provided in [16]. Fig. 1 visualizes the resulting construction principle. This will be used in Section 2.2 to design indicators combinations representing different types of preference information.

2.2 Design of Indicator-Based Relations

In the following, we present some examples for combined set preference relations that illustrate different application scenarios. All of these relations are refinements of the set preference relation $\preccurlyeq_{\text{par}}$.

The first combination is based on the unary epsilon indicator $I_{\epsilon 1}$ [15] with a reference set R in objective space which is defined as $I_{\epsilon 1}(A) = E(A, R)$ with

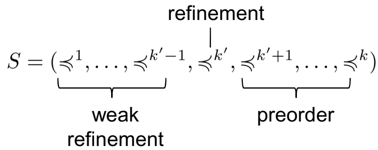


Fig. 1. Representation of the hierarchical construction of refinements according to Theorem 2.4

$E(A, R) = \max_{r \in R} \min_{a \in A} \epsilon(a, r)$ where $\epsilon(a, r) = \max\{f_i(a) - r_i \mid 1 \leq i \leq n\}$ and r_i is the ith component of the objective vector r. Since this indicator induces only a weak refinement of the weak Pareto-dominance relation $\preccurlyeq_{\text{par}}$, we will use the hypervolume indicator to distinguish between sets indifferent with respect $I_{\epsilon 1}$. The resulting set preference relation is denoted as $\preccurlyeq_{\epsilon 1,H}$.

The second combination uses the R_2 indicator proposed in [17] for which the following definition is used here:

$$I_{R2}(A) = R_2(A, R) = \frac{\sum_{\lambda \in \Lambda} u^*(\lambda, R) - u^*(\lambda, f(A))}{|\Lambda|}$$

where the function u^* is a utility function based on the weighted Tchebycheff function $u^*(\lambda, T) = -\min_{z \in T} \max_{1 \leq j \leq n} \lambda_j |z_j^* - z_j|$ and Λ is a set of weight vectors $\lambda \in \mathbb{R}^n$, $R \subset \mathcal{Z}$ is a reference set, and $z^* \in \mathcal{Z}$ is a reference point. In this paper, we will set $R = \{z^*\}$. Also the R_2 indicator provides only a weak refinement; as before, the hypervolume indicator is added in order to achieve a refinement. This set preference relation will be denoted as $\preccurlyeq_{R2,H}$.

Third combination: The previous two indicator combinations couple a weak refinement with a refinement. To demonstrate that also non-refining indicators can be used, we propose the following sequence of indicators $S = (I_H, I_C, I_D)$ where I_C measures the largest distance of a solution to the closest minimal element in a set and I_D reflects the diversity of the solutions in the objective space. The latter two indicators, which both do not induce weak refinements of $\preccurlyeq_{\text{par}}$, are defined as follows: $I_C(A) = \max_{a \in A} \min_{b \in \text{Min}(A, \preceq)} dist(f(a), f(b))$ and

$$I_D(A) = \max_{a \in A} \left(\frac{1}{nn_1(a, A \setminus \{a\})} + \frac{1}{nn_2(a, A \setminus \{a\})} \right)$$

where $nn_1(a, B) = \min_{b \in B} dist(f(a), f(b))$ gives the smallest and $nn_2(a, B) = \max_{c \in B} \min_{b \in B \setminus \{c\}} dist(f(a), f(b))$ the second smallest distance of a to any solution in B. For the distance function $dist(z^1, z^2)$, Euclidean distance is used here, i.e., $dist(z^1, z^2) = \sqrt{\sum_{1 \leq i \leq n} (z_i^1 - z_i^2)^2}$. The I_C indicator resembles the generational distance measure proposed in [18] and I_D resembles the nearest neighbor niching mechanism in SPEA2 [19]. We will refer to the overall set preference relation as $\preccurlyeq_{H,C,D}$. According to Theorem 2.4, $\preccurlyeq_{H,C,D}$ is a refinement of $\preccurlyeq_{\text{par}}$.

Finally, note that set preference relations may be insensitive to dominated solutions in a set, i.e., adding dominated solutions to A or B does not affect the relation between these two sets. This holds for instance for the set preference relations induced by the hypervolume indicator and other popular quality indicators. Nevertheless, to guide the search efficiently it is crucial that prefered solutions are taken into account. One possibiliy is to integrate indicators that are sensitive to dominated solutions such as I_C and I_D defined above. Alternatively, the sets can be partitioned into dominance classes to which the set preference relation is applied subsequently. More precisely, we here use nondominated sorting [20,21] for partitioning and then use the same construction as in Theorem 2.4: to compare A and B with respect to $\preccurlyeq$ we first compare only the nondominted fronts; if this comparison yields indifference, then the second level of nondominance is considered to decide whether A or B is better, and so forth. Whenever this principle of partitioning is used, we write $\preccurlyeq^{\mathsf{minpart}}$; note that $\preccurlyeq^{\mathsf{minpart}}$ is a refinement of $\preccurlyeq$.

3 A General Set Preference Guided Search Algorithm

In the following, we introduce the **S**et **P**reference **A**lgorithm for **M**ultiobjective Optimization (SPAM) which can be used with any set preference relation and resembles a standard hill climber with the difference that two new elements of the search space Ψ are created using two types of mutation operators. The main part of SPAM is given by Algorithm 1.

Starting with a randomly chosen set $P \in \Psi_m$ of size m, first a random mutation operator is applied to generate another set P'. This operator should be designed such that every element in Ψ could be possibly generated, i.e., the neighborhood is in principle the entire search space. In practice, the operator will usually have little effect on the optimization process; however, its property of exhaustivness is important from a theoretical perspective, in particular to show convergence, see [16].

Second, a heuristic mutation operator is employed. This operator mimics the mating selection, variation, and environmental selection steps as used in most MOEAs. The goal of this operator is to create a third set $P'' \in \Psi$ that is better than P in the context of a predefined set preference relation $\preccurlyeq$. However, since it is heuristic it cannot guarantee to improve P; there may be situations where it

Algorithm 1. SPAM Main Loop

Require: set preference relation $\preccurlyeq$

generate initial set P of size m, i.e., randomly choose $A \in \Psi_m$ and set $P \leftarrow A$
while termination criterion not fulfilled **do**
    $P' \leftarrow \mathit{randomSetMutation}(P)$
    $P'' \leftarrow \mathit{heuristicSetMutation}(P)$
    if $P'' \preccurlyeq P$ **then**
        $P \leftarrow P''$
    else if $P' \preccurlyeq P$ **then**
        $P \leftarrow P'$
return P

is not able to escape local optima of the landscape of the underlying set problem. Finally, P is replaced by P'', if the latter is weakly preferable to the former; otherwise, P is either replaced by P' (if $P' \preccurlyeq P$) or remains unchanged. Note that in the last step, weak preferability ($\preccurlyeq$) and not preferability ($\prec$) needs to be considered in order to allow the algorithm to cross landscape plateaus, cf. [22].

For the mutation operators, we propose Algorithms 2 and 3. Algorithm 2 (random set mutation) randomly chooses k decision vectors from $\mathcal{X}$ and uses them to replace k elements in P.[1] Algorithm 3 (heuristic set mutation) generalizes the iterative truncation procedures used in NSGA-II [23], SPEA2 [19], and others. First, k new solutions are created based on P; this corresponds to mating selection plus variation in a standard MOEA. While the variation is problem-specific, for mating selection either uniform random selection (used in the following) or fitness-based selection can be used (using the fitness values computed by Algorithm 4). Then, these k solutions are added to P, and finally the resulting set of size $m+k$ is iteratively truncated to size m by removing the solution with the worst fitness values in each step. Here, the fitness value of $a \in P$ reflects the loss in quality for the entire set P if a is deleted: the lower the fitness, the larger the loss.

Algorithm 2. Random Set Mutation

procedure *randomSetMutation*(P)
  randomly choose $r_1, \ldots, r_k \in \mathcal{X}$ with $r_i \neq r_j$
  randomly select $p_1, \ldots, p_k$ from P with $p_i \neq p_j$
  $P' \leftarrow P \setminus \{p_1, \ldots, p_k\} \cup \{r_1, \ldots, r_k\}$
  return P'

Algorithm 3. Heuristic Set Mutation

procedure *heuristicSetMutation*(P)
  generate $r_1, \ldots, r_k \in \mathcal{X}$ based on P
  $P'' \leftarrow P \cup \{r_1, \ldots, r_k\}$
  while $|P''| > m$ **do**
    for all $a \in P''$ **do**
      $\delta_a \leftarrow$ *fitnessAssignment(a, P")*
    choose $p \in P''$ with $\delta_p = \min_{a \in P''} \delta_a$
    $P'' \leftarrow P'' \setminus \{p\}$
  return P''

To estimate how useful a particular solution $a \in P$ is, Algorithm 4 compares all sets $A_i \subset P$ with $|A_i| = |P| - 1$ to $P \setminus \{a\}$ using the predefined set preference relation $\preccurlyeq$. The fewer sets A_i are weakly preferable to $P \setminus \{a\}$, the better the set $P \setminus \{a\}$ and the less important is a. This procedure has a runtime complexity of $\mathcal{O}((m+k)g)$, where g stands for the runtime needed to compute the preference

[1] Note that for both mutation operators the same k is used here, although they can be chosen independently. The safe version ($k = m$) for the random mutation operator means that a random walk is carried out on Ψ.

Algorithm 4. Fitness Assignment

procedure *fitnessAssignment*(a, P'')
    $\delta_a \leftarrow 0$
    for all $b \in P''$ **do**
        if $P'' \setminus \{b\} \preccurlyeq P'' \setminus \{a\}$ **then**
            $\delta_a \leftarrow \delta_a + 1$
    return δ_a

relation comparisons which usually depends on $m+k$ and the number of objective functions. It can be made faster, when using unary indicators, see [16].

4 Experiments

This section investigates the practicability of the proposed approach. The main questions are: (i) can different user preferences be expressed in terms of set preference relations, (ii) is it feasible to use a general search algorithm for arbitrary set preference relations, i.e., is SPAM effective in finding appropriate sets, and (iii) how well are set preference relations suited to guide the optimization process? However, the purpose is not to carry out a performance comparison of SPAM to existing MOEAs; the separation of user preferences and search algorithm is the focus of our study.

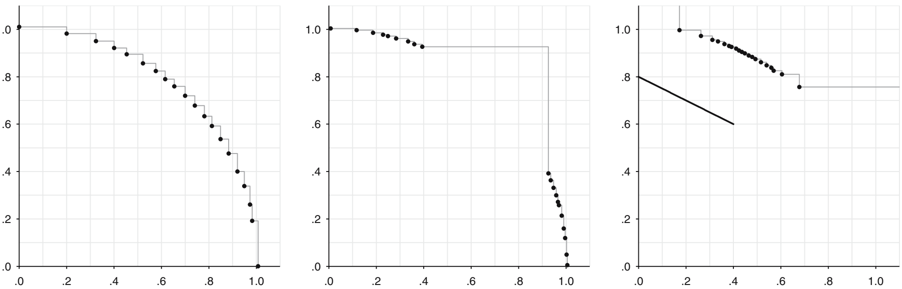
In the following, we consider the different set preference relations presented in Section 2.2 for integration in SPAM where $\preccurlyeq^{\mathsf{minpart}}_{R2,H}$ is parameterized to focus on the outer regions of the Pareto front.[2] In addition, the relations $\preccurlyeq^{\mathsf{minpart}}_{H}$ and $\preccurlyeq^{\mathsf{minpart}}_{\epsilon}$ induced by the unary hypervolume indicator resp. the binary epsilon indicator are used. All of them are refinements of the set dominance relation $\preccurlyeq_{\text{par}}$. As reference algorithms, NSGA-II [23] and IBEA[3] [2] are used. The test problem is DTLZ2 [24] with 20 decision variables and 2 resp. 5 objectives [4]. To compare the outcomes of the algorithms with respect to multiple runs (in this study 30 runs) statistically, we use the Mann-Whitney U test where the significance of the test statistics U is calculated on the basis of the one-tailed normal approximation, correcting the variance for ties, see [16] for details. Furthermore, multiple testing issues need to be taken into account when comparing multiple algorithms with each other; here, the significance levels are Bonferroni corrected.

Figure 2 shows the Pareto-set approximations generated by SPAM with three selected set preference relations. The plots well reflect the chosen user preferences: (a) a set maximizing hypervolume, (b) focus on the extremes using corresponding weight combinations, and (c) closeness to a given reference set. This

[2] The full parameterization of the indicators is given in [16].

[3] With parameters $\kappa = 0.05$ and $\rho = 1.1$.

[4] Other parameters: set size / population size $m = 20$ (visual comparisons) resp. $m = 50$ (statistical comparisons); newly created solutions / offspring individuals $k = 20$ resp. $k = 50$; number of iterations 1000; further details are provided in [16].



(a) SPAM with $\preccurlyeq_H^{\mathsf{minpart}}$ (b) SPAM with $\preccurlyeq_{R2,H}^{\mathsf{minpart}}$ (c) SPAM with $\preccurlyeq_{\epsilon 1,H}^{\mathsf{minpart}}$

Fig. 2. Pareto-set approximations found after 1000 generations on a biobjective DTLZ2 problem for a set size / population size of $m = 20$. All algorithms were started with the same initial set / population.

demonstrates that SPAM is in principle capable of optimizing towards the user preferences that are encoded in the corresponding set preference relation.

The quantitative comparisons for all set preference relations are provided in Table 1. The hypothesis is that SPAM used in combination with a specific set preference relation $\preccurlyeq_A$ (let us say SPAM-A) yields better Pareto set approximations than if used with any other set preference relation $\preccurlyeq_B$ (let us say SPAM-B)—better here means with respect to $\preccurlyeq_A$. Ideally, for every set A generated by SPAM-A and every set B generated by SPAM-B, it would hold $A \preccurlyeq_A B$ or even $A \prec_A B$. Clearly, this describes an ideal situtation. A set preference relation that is well suited for representing certain preferences may not be well suited for search per se. With only few exceptions, the above hypothesis is confirmed: using $\preccurlyeq_A$ in SPAM yields the best Pareto-set approximations with regard to $\preccurlyeq_A$, independently of the problem and the number of objectives under consideration. These results are highly significant at a significance level of 0.001.

Concerning the exceptions, first it can be noticed that there is no significant difference between $\preccurlyeq_H^{\mathsf{minpart}}$ and $\preccurlyeq_{H,C,D}$ when used in SPAM—both times, the hypervolume indicator value is optimized. This actually shows that dominance class partitioning may be replaced by a corresponding sequence of quality indicators. Second, the algorithm based on the set preference relation $\preccurlyeq_\epsilon$ using the binary epsilon indicator performs slighlty worse than IBEA with respect to $\preccurlyeq_\epsilon$. This is not suprising since IBEA has been designed mainly for the epsilon indicator and exploits certain characteristics; for instance, all population members are compared to each other and not only those in the current front. In addition, SPAM with the binary epsilon indicator performs significantly worse than SPAM with any of the two hypervolume-based relations $\preccurlyeq_H^{\mathsf{minpart}}$ and $\preccurlyeq_{H,C,D}$ in the case of two objectives. This may indicate that the binary epsilon indicator is not sensitive enough to differentiate between small improvements. That means that in Step 7 of Algorithm 3 too many solutions may achieve the minimum δ-value, and therefore a choice needs to be done at random.

Table 1. Pairwise statistical comparison of 30 runs per algorithm after 1000 generations. In the notation $U : U'$, U (resp. U') stands for the number of times a set generated by algorithm A (resp. B) beats a set of algorithm B (resp. A) with regard to the test relation associated with the corresponding row. A star next to these numbers indicates a significant difference, the few cases where this was not the case are shown in bold. Per cell, the upper number pair corresponds to the biobjective DTLZ2 problem, the lower number pairs to the 5-objective DTLZ2 problem.

alg. A \ alg B.		SPAM with set preference relation ... $\preccurlyeq^{\text{minpart}}_{H}$	SPAM with set preference relation ... $\preccurlyeq^{\text{minpart}}_{R2,H}$	SPAM with set preference relation ... $\preccurlyeq^{\text{minpart}}_{\epsilon 1,H}$	SPAM with set preference relation ... $\preccurlyeq^{\text{minpart}}_{\epsilon}$	SPAM with set preference relation ... $\preccurlyeq_{H,C,D}$	IBEA	NSGA-II	test relation
SPAM with set preference relation ...	$\preccurlyeq^{\text{minpart}}_{H}$	- -	900: 0* 900: 0*	900: 0* 900: 0*	900: 0* 900: 0*	**456:444** **445:455**	900: 0* 900: 0*	900: 0* 900: 0*	$\preccurlyeq_{H}$
	$\preccurlyeq^{\text{minpart}}_{R2,H}$	900: 0* 900: 0*	- -	900: 0* 900: 0*	900: 0* 900: 0*	900: 0* 900: 0*	900: 0* 900: 0*	900: 0* 900: 0*	$\preccurlyeq_{R2,H}$
	$\preccurlyeq^{\text{minpart}}_{\epsilon 1,H}$	900: 0* 891: 9*	900: 0* 900: 0*	- -	900: 0* 900: 0*	889: 1* 897: 3*	900: 0* 900: 0*	900: 0* 900: 0*	$\preccurlyeq_{\epsilon 1,H}$
	$\preccurlyeq^{\text{minpart}}_{\epsilon}$	**63:837** **60:840**	900: 0* 900: 0*	900: 0* 900: 0*	- -	**81:819** **57:843**	**274:626** **349:551**	896: 4* 889: 11*	$\preccurlyeq_{\epsilon}$
	$\preccurlyeq_{H,C,D}$	**444:456** **455:445**	900: 0* 900: 0*	900: 0* 900: 0*	853: 47* 900: 0*	- -	820: 80* 900: 0*	900: 0* 900: 0*	$\preccurlyeq_{H,C,D}$

* Preference is significant at the 0.001 level (1-tailed, Bonferroni-adjusted).

5 Conclusions

This paper proposed a general way to separate preference formalization from algorithm design and presented SPAM, a flexible multiobjective optimizer, which is basically a hill climber and generalizes the concepts found in most modern MOEAs. SPAM can be used in combination with any type of set preference relation and thereby offers full flexibility for the decision maker. Furthermore, a novel scheme to design set preference relations by putting multiple quality indicators in sequence was introduced.

References

1. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)
2. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN VIII 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
3. Fonseca, C.M., Fleming, P.J.: Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. IEEE Transactions on Systems, Man, and Cybernetics 28(1), 26–37 (1998)
4. Branke, J., Kaußler, T., Schmeck, H.: Guidance in Evolutionary Multi-Objective Optimization. Advances in Engineering Software 32, 499–507 (2001)
5. Cvetković, D., Parmee, I.C.: Preferences and their Application in Evolutionary Multiobjective Optimisation. IEEE Transactions on Evolutionary Computation 6(1), 42–57 (2002)
6. Branke, J., Deb, K.: Integrating User Preferences into Evolutionary Multi-Objective Optimization. Technical Report 2004004, Indian Institute of Technology,

Kanpur, India (2004); In: Jin, Y. (ed). Knowledge Incorporation in Evolutionary Computation, pp. 461–477. Springer, Heidelberg (2004)

7. Deb, K., Sundar, J.: Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In: Keijzer, M., et al. (eds.) Conference on Genetic and Evolutionary Computation (GECCO 2006), pp. 635–642. ACM Press, New York (2006)
8. Rachmawati, L., Srinivasan, D.: Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. In: IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver, BC, Canada, pp. 3385–3391. IEEE Press, Los Alamitos (2006)
9. Mehnen, J., Trautmann, H., Tiwari, A.: Introducing User Preference Using Desirability Functions in Multi-Objective Evolutionary Optimisation of Noisy Processes. In: IEEE Congress on Evolutionary Computation (CEC 2007), pp. 2687–2694. IEEE Press, Los Alamitos (2007)
10. Emmerich, M., Beume, N., Naujoks, B.: An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 62–76. Springer, Heidelberg (2005)
11. Igel, C., Hansen, N., Roth, S.: Covariance Matrix Adaptation for Multi-objective Optimization. Evolutionary Computation 15(1), 1–28 (2007)
12. Zitzler, E., Brockhoff, D., Thiele, L.: The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In: Obayashi, S., et al. (eds.) EMO 2007. LNCS, vol. 4403, pp. 862–876. Springer, Heidelberg (2007)
13. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
14. Knowles, J., Corne, D.: On Metrics for Comparing Non-Dominated Sets. In: Congress on Evolutionary Computation (CEC 2002), pp. 711–716. IEEE Computer Society Press, Piscataway (2002)
15. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
16. Zitzler, E., Thiele, L., Bader, J.: On Set-Based Multiobjective Optimization. Technical Report 300, Computer Engineering and Networks Laboratory, ETH Zurich (2008)
17. Hansen, M.P., Jaszkiewicz, A.: Evaluating the quality of approximations of the nondominated set. Technical report, Institute of Mathematical Modeling, Technical University of Denmark, IMM Technical Report IMM-REP-1998-7 (1998)
18. Veldhuizen, D.A.V., Lamont, G.B.: Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation 8(2), 125–147 (2000)
19. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou, K., et al. (eds.) Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE), pp. 95–100 (2002)
20. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
21. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221–248 (1994)

22. Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., Zitzler, E.: Do Additional Objectives Make a Problem Harder? In: Thierens, D., et al. (eds.) Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 765–772. ACM Press, New York (2007)
23. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: Schoenauer, M., et al. (eds.) PPSN VI 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
24. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Multi-Objective Optimization Test Problems. In: Congress on Evolutionary Computation (CEC 2002), pp. 825–830. IEEE Press, Los Alamitos (2002)