Convergence of set-based multi-objective optimization, indicators and deteriorative cycles☆

Rudolf Berghammer [a], Tobias Friedrich [b,*], Frank Neumann [c]

[a] *Institut für Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany*
[b] *Institut für Informatik, Friedrich-Schiller-Universität Jena, 07743 Jena, Germany*
[c] *School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia*

ARTICLE INFO

ABSTRACT

Multi-objective optimization deals with the task of computing a set of solutions that represents possible trade-offs with respect to a given set of objective functions. Set-based approaches such as evolutionary algorithms are very popular for solving multi-objective optimization problems. Convergence of set-based approaches for multi-objective optimization is essential for their success. We take an order-theoretic view on the convergence of set-based multi-objective optimization and examine how the use of indicator functions can help to direct the search towards Pareto optimal sets. In doing so, we point out that set-based multi-objective optimization working on the dominance relation of search points has to deal with a cyclic behavior that may lead to worsening with respect to the Pareto-dominance relation defined on sets. Later on, we show in which situations well-known binary and unary indicators can help to avoid this cyclic behavior and therefore guarantee convergence of the algorithm. We also study the impact of deteriorative cycles on the runtime behavior and give an example in which they provably slow down the optimization process.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Set-based optimization methods such as evolutionary algorithms have been widely used for multi-objective optimization problems [9]. Often multi-objective problems are hard to tackle, as there is no total ordering of the underlying search space. Instead, a preorder on the search space is induced by the different objective functions that should be optimized. The minimal elements of such a preorder are called Pareto optimal solutions and the set of minimal elements with respect to the partial order on the objective space is called the Pareto front. The goal of an algorithm for a given multi-objective optimization problem is to compute for each objective vector of the Pareto front a corresponding Pareto optimal search point. Often the size of the Pareto front grows exponentially with respect to the size of the input. This situation calls for a smaller set of minimal elements that fulfills given user preferences.

Convergence is an important property of a set-based algorithm for multi-objective optimization and decisive for its success. Initial studies on this topic have been carried out by Rudolph [25] who stated conditions for the convergence of evolutionary algorithms to the Pareto optimal set for finite search spaces. Convergence for continuous search spaces

---

has been studied by Hanne [14] and the convergence to good approximations of the Pareto front has been investigated in [19,20,26,27]. We study convergence of set-based algorithms for multi-objective optimization in a general and abstract way by examining the relation of sets of search points that such algorithms work with.

Recently, Zitzler et al. [31] have extended the preorder on the search space to a preorder on sets of search points. This leads to the term *set-based multi-objective optimization*, which stresses the point that it is not single solutions that should be compared with respect to a preorder, but rather sets of search points that have to be compared. Zitzler et al. [31] introduced a preorder on the set of sets of search points that is based on the preorder on the underlying search space. They also use this preorder as a basic relation that may be later refined by special user preferences. We examine the approach of working with a preorder on sets in greater detail. In particular, we discuss the preorder on sets from a theoretical point of view and point out criteria for convergence of set-based algorithms for multi-objective optimization.

Evolutionary algorithms are a prominent type of set-based algorithm, and we will consider them as an example throughout this paper. We do not impose any restrictions on how such an algorithm produces, from a current set of search points called the parent population, a new set of search points which is usually called the offspring population. Our only assumption is that a parent population consisting of μ individuals produces, in each iteration, an offspring population consisting of λ individuals. Having produced an offspring population, the task of the selection operator is then to select a new parent population.

We examine how to design a selection operator such that the newly chosen parent population consists of a set of μ individuals that is minimal among all sets that can be obtained by selecting μ individuals from the set of parents and children. Later on, we relate this to well known selection methods, such as non-dominated sorting used in NSGA-II [10] and ranking ideas used in SPEA2 [29].

Having examined how to compute a minimal set of search points in each iteration, we investigate the run of such algorithms from a theoretical point of view with respect to the Pareto-dominance order defined on sets. We show that just working with the dominance relation on sets may lead to a cyclic behavior that can lead to worsening with respect to the dominance relation. This behavior has already been observed in experimental studies of NSGA-II and SPEA2 (see [21]) and may prevent those algorithms from convergence.

Examining this topic in greater detail, we relate convergence of an algorithm to deteriorative cycles in the underlying relation of the algorithm. Interesting studies related to this topic in the context of coevolutionary algorithms have recently been carried out in [11,24]. We show that the absence of these deteriorative cycles is a key property that guarantees convergence. We point out conditions that avoid such an undesired cyclic behavior and therefore lead to convergence. Based on these conditions, we examine how refinements using indicator functions for incomparable sets may help to solve this problem. We show that well-known binary indicators such as the additive and multiplicative ε-indicator can resolve this problem for the case that the parent population is of size 1. On the negative side, we show that even a parent population of size 2 can again run into the non-described cyclic behavior.

Furthermore, we examine the use of unary indicators. Based on our conditions for avoiding cyclic behavior, we show that the well-known hypervolume indicator is able to deal with this problem in a successful manner if a fixed reference point is used during the whole run of the algorithm. Some popular variants of hypervolume-based algorithms work with a dynamically changing reference point. A well-known example is the SMS-EMOA [4]. For its simplest version, called (1+1) SMS-EMOA, convergence rates have been examined for convex functions with two objectives [3]. We show that even the simple (1+1) SMS-EMOA has to face the problem of deteriorative cycles when it is applied to problems with at least three objectives. This shows that the use of a dynamically changing reference point as used in the SMS-EMOA may prevent the algorithm from converging. Our investigations add to the foundations for this indicator and give further insight into its optimization behavior.

In the last part of this paper, we study the impact of deteriorative cycles by rigorous runtime analysis. This type of analysis has significantly increased the theoretical understanding of how and why randomized search heuristics such as evolutionary algorithms and ant colony optimization work for certain types of problems [1,23]. We consider simple randomized search heuristics introduced by Giel and Lehre [13] and study them on a multi-objective example problem motivated by rigorous studies on plateau functions carried out by Jansen and Wegener [16]. The behavior of randomized search heuristics on multi-objective plateau functions has been studied in [12,8] and our investigations also contribute to this understanding. First, we emphasize the importance of moving between incomparable solutions, which is one of the key issues in set-based multi-objective optimization. Afterwards, we show that averting the problem of deteriorative cycles by using Pareto compliant unary indicators, such as the hypervolume indicator, significantly reduces the runtime of simple randomized search heuristics on our example problem. This article extends the results of its conference version [2] by the investigations on the hypervolume indicator with changing reference point carried out in Section 5.2 and the runtime analysis presented in Section 7.

The outline of the paper is as follows. In Section 2 we recall some basic properties of set-based multi-objective optimization. Section 3 deals with the task of computing a minimal set from the parents and children. The problem of deteriorative cycles and conditions for avoiding them are pointed out in Section 4. In Section 5, we examine whether it is possible to avoid this cyclic behavior by using unary indicators, and we study binary indicators in Section 6. Investigations regarding the impact of deteriorative cycles on the runtime behavior are carried out in Section 7. Finally, we present some conclusions.

2. Set-based multi-objective optimization

A multi-objective optimization problem is given by a vector-valued objective function $f = (f_1, \ldots, f_d) \colon X \to \mathbb{R}^d_{\geq 0}$ on a search space X. W. l. o. g. we assume that each function f_i, $1 \leq i \leq d$, should be minimized. We first define a partial order on the objective space. An objective vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ weakly dominates an objective vector $y = (y_1, \ldots, y_d) \in \mathbb{R}^d$ ($x \preceq_{Par} y$) if it is not worse in any objective, i. e.,

$$x \preceq_{Par} y :\Leftrightarrow x_i \leq y_i \quad \text{for } 1 \leq i \leq d.$$

The objective function f also induces a preorder $\preceq_{Par}$ on the search space X. More precisely, we say a search point $a \in X$ weakly dominates a search point $b \in X$ ($a \preceq_{Par} b$) if it is not worse in any of its objectives, i. e.,

$$a \preceq_{Par} b :\Leftrightarrow f(a) \preceq_{Par} f(b).$$

Note that we use $\preceq_{Par}$ as a relation on search points as well as a relation on the corresponding objective vectors. The relation $\preceq_{Par}$ is a preorder on the set of search points and a partial order on the set of objective vectors.

In investigating sets of search points in this paper, we assume that these sets are finite. Given a set of search points A, we denote by $\text{Min}(A, \preceq_{Par})$ the set of minimal elements of A with respect to the preorder $\preceq_{Par}$. Let $f(A)$ be the set of objective vectors of the search points in A, i. e.,

$$f(A) = \{f(a) \ : \ a \in A\}.$$

Then, we denote by $\text{Min}(f(A), \preceq_{Par})$ the set of minimal objective vectors in $f(A)$ with respect to the partial order $\preceq_{Par}$ on $f(A)$.

The goal in multi-objective optimization is to compute a set X^* with $f(X^*) = \text{Min}(f(X), \preceq_{Par})$, where X is the considered search space. Often the size of $\text{Min}(f(X), \preceq_{Par})$ is large, i. e., exponential with respect to the given input. In this case, it is not possible to compute the whole set of minimal elements of $f(X)$ efficiently, and $f(X^*)$ should be a smaller subset of them.

In this paper, we consider set-based algorithms for multi-objective optimization. They attempt to construct a set X^* with $f(X^*) \subseteq \text{Min}(f(X), \preceq_{Par})$ in an iterative way by starting with an initial set of search points and producing a new set of search points in each iteration. As these work with sets of search points, we want to compare sets of search points against each other. Let 2^X be the power-set of X, i. e., $2^X := \{R \mid R \subseteq X\}$. Based on a preorder $\preceq_{Par}$ (reflexive and transitive) on single search points, Zitzler et al. [31] have defined the following preorder on sets of search points that conforms to the preorder $\preceq_{Par}$ on the underlying set of search points X.

Definition 2.1. Let $A, B \in 2^X$ then

$$A \preceq_{dom} B :\Leftrightarrow (\forall b \in B \, \exists a \in A \colon a \preceq_{Par} b).$$

We consider an arbitrary relation $\preceq$ on sets and introduce the following definitions.

Definition 2.2. Let $A, B \in 2^X$ and $\preceq$ be an arbitrary relation on 2^X. Then we use

$$A \prec B :\Leftrightarrow (A \preceq B) \wedge (B \npreceq A)$$

to denote that A is strictly better than B. We further write

$$A \equiv B :\Leftrightarrow (A \preceq B) \wedge (B \preceq A)$$

to denote that A and B are equivalent, and

$$A \parallel B :\Leftrightarrow (A \npreceq B) \wedge (B \npreceq A)$$

to denote that A and B are incomparable. Further, let

$$A \succeq B :\Leftrightarrow B \preceq A,$$
$$A \succ B :\Leftrightarrow B \prec A.$$

Set based algorithms such as evolutionary algorithms often work at each time step with a fixed population size μ. Therefore, we consider subsets of X containing exactly μ elements. The goal is to obtain a set that is a minimal set with respect to the order $\preceq_{dom}$ among all subsets of X having exactly μ elements.

We denote the set of minimal elements containing exactly μ elements of X by $\text{Min}_\mu(2^X, \preceq)$, i. e.

$$\text{Min}_\mu(2^X, \preceq) := \text{Min}\{R \mid R \in 2^X \wedge |R| = \mu\}.$$

Note that such a set is also a minimal set among all subsets having less than μ elements according to Definition 2.1.

The relation between two sets A and B is determined by the set of their minimal elements. The following lemma relates equivalent sets and the set of their minimal objective vectors.

Algorithm 1: Set-based evolutionary algorithm

---

1 Create an initial population P consisting of μ individuals;
2 Produce from P an offspring population C consisting of λ individuals;
3 Select a set P' with $P' \in \mathrm{Min}_\mu(2^{(P \cup C)}, \preceq_{dom})$;
4 Set $P = P'$;
5 If no termination condition is fulfilled go to step 2;

Lemma 2.3. *If A and B are sets of search points, then*

$$A \equiv_{dom} B \Leftrightarrow \mathrm{Min}(f(A), \preceq_{Par}) = \mathrm{Min}(f(B), \preceq_{Par}).$$

Proof. "$\Rightarrow$": Let $A, B \in 2^X$ with $A \equiv_{dom} B$. We assume that $\mathrm{Min}(f(A), \preceq_{Par}) \neq \mathrm{Min}(f(B), \preceq_{Par})$ and show a contradiction. Let $c \in A$ be such that $f(c) \in \mathrm{Min}(f(A), \preceq_{Par})$ but $f(c) \notin \mathrm{Min}(f(B), \preceq_{Par})$. Since $B \preceq_{dom} A$ and $f(c) \in f(A)$ but $f(c) \notin \mathrm{Min}(f(B), \preceq_{Par})$ there exists $b \in B$ such that $f(b) \in \mathrm{Min}(f(B), \preceq_{Par})$ and $b \prec_{Par} c$. From $A \preceq_{dom} B$ we get $a \preceq_{Par} b$ for some $a \in A$. Hence, we have $a \preceq_{Par} c$ and $f(c) \in \mathrm{Min}(f(A), \preceq_{Par})$ implies $f(a) = f(c)$. As a result, we arrive at $f(b) \prec_{Par} f(a)$ and $f(a) \preceq_{Par} f(b)$, i. e., the desired contradiction.

"$\Leftarrow$": Let $f(a) \in f(A)$. Then there exists $f(b) \in \mathrm{Min}(f(A), \preceq_{Par})$ such that $f(b) \preceq_{Par} f(a)$. Now $\mathrm{Min}(f(A), \preceq_{Par}) = \mathrm{Min}(f(B), \preceq_{Par})$ implies $f(b) \in f(B)$. Hence, we get $B \preceq_{dom} A$. Since in the same way $A \preceq_{dom} B$ can be shown, we have $A \equiv_{dom} B$. $\square$

Evolutionary algorithms work in each iteration with a parent population that creates an offspring population by some variation operators, such as crossover and mutation. We will treat set-based algorithms in a general way and do not make any assumptions on how the offspring population is produced. Note that both the parent and offspring population are multi-sets, i. e., they may contain search points more than once. Comparing multi-sets with respect to the dominance relation, we ignore duplicates, i. e. we treat multi-sets as their corresponding sets in 2^X.

After the set of offspring has been obtained, the goal is to choose a new parent population such that the process can be iterated. We assume that the parent population has a fixed number of individuals μ, which is very common in evolutionary computation. The size of the offspring population is denoted by λ and the goal is to select out of the $\mu + \lambda$ individuals a new parent population that is a minimal set among all possible subsets consisting of μ individuals.

To make the setting more precise, we examine Algorithm 1. Our algorithm starts with a population consisting of μ individuals. In each iteration, λ offspring are produced. The new parent population is afterwards chosen as a minimal element with respect to the set preference relation $\preceq_{dom}$ among all possible subsets of the parents and offspring that consist of exactly μ individuals.

Consider a set $P' \in \mathrm{Min}_\mu(2^{(P \cup C)}, \preceq)$ and compare it to P. P' is minimal and $P \in \{R : R \in 2^{(P \cup C)} \wedge |R| = \mu\}$ which implies that either

$$P' \preceq_{dom} P$$

or

$$P' \parallel_{dom} P$$

holds. $P \prec_{dom} P'$ would contradict the assumption that P' is minimal. If $P' \prec_{dom} P$ we have obtained a strict improvement with respect to the dominance relation on sets. $P' \equiv_{dom} P$ gives us an equivalent set and $P' \parallel_{dom} P$ a set that is incomparable.

Evolutionary algorithms in our setting work implicitly on a total relation defined on sets, as an algorithm has to make the decision as to which set to take for the next iteration. Let $\preceq_{Alg}$ be the total relation on 2^X that an algorithm *Alg* implicitly uses. In the case that $P' \parallel_{dom} P$ it is not clear which set to favor over the other. In the following discussion, we treat incomparable sets in the same way as indifferent sets. This is very common in evolutionary multi-objective optimization if no additional information is available. Later on, we will examine how additional information based on an indicator function can influence the search.

Algorithm 1 is based on the dominance relation $\preceq_{dom}$ on sets. However, it may also move from a set P to P' iff $P \parallel_{dom} P'$, which is often the case for evolutionary algorithms. The algorithm works implicitly on the total relation $\preceq_{Alg_1}$ on 2^X defined as

$$A \preceq_{Alg_1} B \Leftrightarrow (A \preceq_{dom} B) \vee (A \parallel_{dom} B)$$

and may move from P to P' iff $P' \preceq_{Alg_1} P$ holds. Note that $\preceq_{Alg_1}$ is not necessarily a transitive relation.

3. Computing minimal sets

In this section, we examine how to compute a set contained in $\mathrm{Min}_\mu(2^{(P \cup C)}, \preceq_{dom})$. We will see that this can be done by using an iterative algorithm that chooses, in each iteration, a minimal element with respect to the Pareto dominance relation on single points. This is similar to how well known evolutionary algorithms for multi-objective optimization choose their offspring population. In the following, we present the ideas and basic properties that are necessary to compute such a set in a precise way.

Algorithm 2: $\text{Min}_\mu(2^S, \preceq_{dom})$

Input: S with $|S| \geq \mu$
1   $T = \emptyset$;
2   **while** $|T| < \mu$ **do**
3   Choose an element $x \in \text{Min}(S, \preceq_{Par})$;
4      $T = T \cup \{x\}$;
5      $S = S \setminus \{x\}$;

Algorithm 3: $\text{Min}_\mu(2^S, \preceq_{dom})$ with preferences

---

Input: S with $|S| \geq \mu$
1   $T = \emptyset$;
2   **while** $|P| < \mu$ **do**
3   Choose an element $x \in \text{Min}(S, \preceq_{Par})$ for which $h(P, x)$ is maximal;
4      $T = T \cup \{x\}$;
5      $S = S \setminus \{x\}$;

We consider the preorder $\preceq_{dom}$ on the subsets of $S := P \cup C$ with exactly μ elements. To obtain a set T for which $T \in \text{Min}_\mu(2^S, \preceq_{dom})$ holds we consider Algorithm 2.

In each iteration, Algorithm 2 chooses one individual x that is minimal with respect to S and $\preceq_{Par}$. This individual is inserted into T and deleted from S. Iterations continue until μ individuals have been chosen in this way. The following theorem shows that Algorithm 2 computes a minimal set among all subsets of S that have exactly μ elements.

Theorem 3.1. *For the set T produced by Algorithm 2, $T \in \text{Min}_\mu(2^S, \preceq_{dom})$ holds.*

Proof. Algorithm 2 selects an element x if $x \in \text{Min}(S, \preceq_{Par})$. Hence, x can only be dominated by another $y \in S$ that has been included into T before the selection of x has taken place. This implies that there is no element z in $S \setminus P$ for which $z \preceq_{Par} x$ holds. Therefore, $T \in \text{Min}_\mu(2^S, \preceq_{dom})$ holds. $\square$

Next, we want to modify Algorithm 2 such that user preferences can be incorporated. Often, one is not interested only in an arbitrary set $T \in \text{Min}_\mu(2^S, \preceq_{dom})$, but rather a set from $\text{Min}_\mu(2^S, \preceq_{dom})$ having additional properties. In the process of constructing the set T, we use a heuristic function

$$h \colon 2^X \times X \to \mathbb{R}$$

which determines the choice of the next minimal element that should be included into T depending on the already chosen elements and the available minimal elements.

Algorithm 3 computes a minimal set of elements taking into account a heuristic function h. This function can be used to incorporate information on how the chosen points should relate to each other. As Algorithm 3 selects in each iteration a minimal element from the remaining set, we can state the following corollary.

Corollary 3.2. *For the set T produced by Algorithm 3, $T \in \text{Min}_\mu(2^S, \preceq_{dom})$ holds.*

We have shown two simple algorithms for computing a minimal set. Note that many well-known evolutionary algorithms for multi-objective optimization such as NSGA-II [10] and SPEA2 [29] use a similar approach to compute the next parent population. In fact, it can be shown by similar arguments that they also compute a set in $\text{Min}_\mu(2^S, \preceq_{dom})$.

4. Deteriorative cycles

In this section, we examine algorithms that always compute a minimal set among all possible sets consisting of μ individuals. We point out that using just the preference order on the different subsets of $P \cup C$ may lead to cycles in the optimization process, i. e., the algorithm may return to a set of search points that has already been obtained. Even worse, we show that the algorithm may return to a set of search points that is strictly dominated by another set of search points that has been obtained at an earlier stage of the optimization process.

4.1. The problem of deteriorative cycles

It should be noted that using an approach that chooses an *arbitrary* set of $\text{Min}_\mu(2^{(P \cup C)}, \preceq_{dom})$ may create cycles. In particular, we show by example that things might get worse during the optimization process according to the preorder $\preceq_{dom}$. We call such cycles *deteriorative*; a precise definition follows.
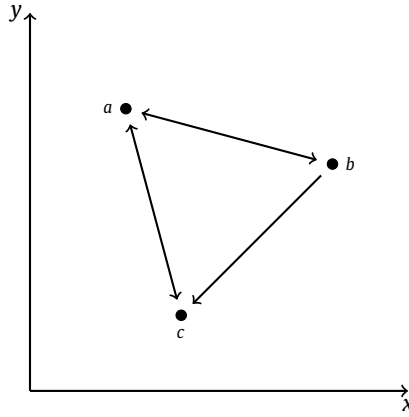
Fig. 1. Objective space for deteriorative cycle based on Pareto-dominance relation.

Definition 4.1 (*Deteriorative Cycle*)**.** A relation $\preceq$ on 2^X contains a deteriorative cycle iff there is a sequence of sets $A_1, A_2, \ldots A_r \in 2^X$ with

$$A_1 \preceq A_2 \preceq \cdots \preceq A_{r-1} \preceq A_r \preceq A_1$$

and $A_i \prec_{dom} A_1$ for some $i \in \{1, \ldots, r\}$.

As a special case, we also introduce the notion of a *strict* deteriorative cycle, which is the one used in [2]. This definition requires that at least one step in the cycle represents a strict improvement with respect to the dominance relation.

Definition 4.2 (*Strict Deteriorative Cycle*)**.** A relation $\preceq$ on 2^X contains a strict deteriorative cycle iff there is a sequence of sets $A_1, A_2, \ldots A_r \in 2^X$ with

$$A_1 \preceq A_2 \preceq \cdots \preceq A_{r-1} \preceq A_r \preceq A_1$$

and $A_r \prec_{dom} A_1$.

Note that if a cycle C is a strict deteriorative cycle according to Definition 4.2, then C is also a deteriorative cycle according to Definition 4.1. The reverse is not true, and we will discuss the definitions in greater detail to follow. Note that for transitive relations, the two definitions are equivalent.

To illustrate the problem that deteriorative cycles may produce in the optimization process, we consider a simple example given in Fig. 1, together with Algorithm 1. For simplicity, we assume that $|P| = |C| = 1$ and that the three sets consist of exactly one element. We have

$$\{a\} \parallel_{dom} \{b\} \Rightarrow \{a\} \preceq_{Alg_1} \{b\} \wedge \{b\} \preceq_{Alg_1} \{a\}$$
$$\{b\} \succ_{dom} \{c\} \Rightarrow \{b\} \succ_{Alg_1} \{c\}$$
$$\{a\} \parallel_{dom} \{c\} \Rightarrow \{a\} \preceq_{Alg_1} \{c\} \wedge \{c\} \preceq_{Alg_1} \{a\}.$$

The relation $\preceq_{Alg_1}$ contains a strict deteriorative cycle according to Definition 4.2, as

$$\{b\} \succeq_{Alg_1} \{c\} \succeq_{Alg_1} \{a\} \succeq_{Alg_1} \{b\}.$$

and

$$\{b\} \succ_{dom} \{c\}.$$

Next, we describe how Algorithm 1 may produce this cycle during the optimization process. Assume that the algorithm starts with the population $P_1 = \{b\}$ and produces the first offspring c. Due to Pareto-dominance $\{c\} \prec_{Alg_1} P_1$ holds, and $P_2 = \{c\}$ becomes the new parent population. The offspring of P_2 is a, $\{a\}$ is incomparable to P_2, and therefore $\{a\} \preceq_{Alg_1} P_2$. Hence, $P_3 = \{a\}$ may be the new parent population. Similar, b may be the next offspring, and the set $\{b\} = P_1$ is incomparable to P_3 ($\{b\} \preceq_{Alg_1} P_3$), so the algorithm may proceed to $P_4 = \{b\} = P_1$, creating a deteriorative cycle.

Looking at the example, we can also observe that deteriorative cycles do not have to contain single steps that are strict improvements with respect to the dominance relation.

According to Definition 4.1, the cycle

$$\{b\} \succeq_{Alg_1} \{a\} \succeq_{Alg_1} \{c\} \succeq_{Alg_1} \{a\} \succeq_{Alg_1} \{b\}$$

and

$$\{b\} \succ_{dom} \{c\}$$

is a (non-strict) deteriorative cycle. It consists only of steps between incomparable search points and allows going from $\{b\}$ to $\{c\}$, where $\{b\} \succ_{dom} \{c\}$, and afterwards back to $\{b\}$. However, it is not a strict deteriorative cycle according to Definition 4.2, as it does not include a single step that is a strict improvement with respect to the dominance relation.

Algorithm 4: Cycle-free Optimizer

---

1 Choose an initial population P consisting of μ individuals;
2 Produce an offspring population C;
3 Compute a minimal set $P' \in \text{Min}_\mu(2^{(P \cup C)}, \preceq_{dom})$ using Algorithm 2 or 3;
4 **if** $(P' \prec_{dom} P) \vee (P' \equiv_{dom} P) \vee (P' \preceq_I P)$ **then** $P := P'$;
5 **if** no stopping criteria is fulfilled **then** go to step 2;

4.2. Coping with deteriorative cycles

We have seen that using just the preference relation on sets and allowing moves between incomparable sets may lead to a deteriorative cyclic behavior. This is due to the fact that evolutionary multi-objective optimization has to deal with incomparable sets. An algorithm relying just on the generalized dominance relation on sets and selecting arbitrarily between incomparable sets may accept a set of search points A that is strongly dominated by another set B that has been obtained before A.

Now, we examine how to deal with the cyclic behavior. We discuss the properties of $\preceq$ that result in it not containing a deteriorative cycle. The first criterion is that it is compliant with the Pareto dominance relation.

Definition 4.3. A relation $\preceq$ on 2^X is Pareto-compliant if

$$A \prec_{dom} B \Rightarrow A \prec B.$$

A relation $\preceq$ is strictly Pareto-compliant if, additionally,

$$A \equiv_{dom} B \Rightarrow A \equiv B.$$

This definition deviates slightly from the use of the terms Pareto-compliant and strictly Pareto-compliant in [18,32]. Note that if $\preceq$ is strictly Pareto-compliant, then

$$A \preceq_{dom} B \Rightarrow A \preceq B.$$

Definition 4.3 states that the relation $\preceq$ is Pareto-compliant if it can also distinguish between sets that strictly dominate each other. Note that $\preceq$ is Pareto compliant iff it is a refinement of the Pareto dominance relation $\preceq_{dom}$ on sets in terms of Zitzler et al. [31]. We have seen that just Pareto-compliance alone does not preclude cyclic behavior, as the Pareto-dominance relation is itself Pareto-compliant. To avoid cyclic behavior, we need an additional property.

Definition 4.4. A relation $\preceq$ on 2^X is called transitive if

$$\big((A \preceq B) \wedge (B \preceq C) \big) \Rightarrow A \preceq C.$$

We now show that a relation $\preceq$ that is Pareto compliant and transitive does not contain a deteriorative cycle. We will use this property later on to show which algorithms do not encounter a deteriorative cycle.

Theorem 4.5. *If $\preceq$ is Pareto-compliant and transitive then $\preceq$ does not contain a deteriorative cycle.*

Proof. We prove the theorem by contradiction. Assume that $\preceq$ contains a deteriorative cycle consisting of sets $A_1, A_2, \ldots, A_r \in 2^X$ with

$$A_1 \preceq A_2 \preceq \cdots \preceq A_{r-1} \preceq A_r \preceq A_1$$

and $A_r \prec_{dom} A_1$.

By transitivity of $\preceq$, we get $A_1 \preceq A_r$. As $\preceq$ is also Pareto-compliant, we know from $A_r \prec_{dom} A_1$ that $A_r \prec A_1$ and hence by definition of $\prec$, $A_1 \npreceq A_r$ which contradicts $A_1 \preceq A_r$. Hence $\preceq$ cannot contain a deteriorative cycle. $\square$

We want to modify Algorithm 1 such that the underlying relation does not contain a deteriorative cycle. Algorithm 4 differs from Algorithm 1 by using an additional relation $\preceq_I$ in the case that $P' \parallel_{dom} P$ holds.

Step 4 should be read like a short-circuit evaluation of Boolean operators in most modern programming languages. That is, if $P' \prec_{dom} P$ or $P' \equiv_{dom} P$, the algorithm can decide to choose P' without calculating the (usually much more expensive) relation $\preceq_I$. This way, the algorithm works on the underlying relation $\preceq_{Alg_4}$ given by

$$A \preceq_{Alg_4} B :\Leftrightarrow (A \preceq_{dom} B) \vee ((A \parallel_{dom} B) \wedge (A \preceq_I B)).$$

If $\preceq_I$ is strictly Pareto compliant then $\preceq_{Alg_4} = \preceq_I$. If $\preceq_I$ is in also transitive, then $\preceq_{Alg_4} = \preceq_I$ does not contain a deteriorative cycle. We state this property in the following corollary.

Corollary 4.6. *If $\preceq_I$ is strictly Pareto compliant and transitive then $\preceq_{Alg_4}$ does not contain a deteriorative cycle.*

5. Unary indicators

We now want to examine the common approach to define refinements via indicator functions (see e.g. [31]). In this section, we focus on unary indicator functions, while the next section examines binary indicator functions. Unary indicator functions assign to each set a real number that in some way reflects the set's quality, i.e.,

$$I_1 \colon 2^X \to \mathbb{R}.$$

To define a relation based on a unary indicator function, we use the following definition.

Definition 5.1. For a unary indicators I_1 we set

$$A \preceq_{I_1} B :\Leftrightarrow I_1(A) \leq I_1(B),$$
$$A \prec_{I_1} B :\Leftrightarrow I_1(A) < I_1(B).$$

Note that the relation $\preceq_{I_1}$ is total and also transitive, as the order on real values forms a transitive relation. Also observe the following simple property, which follows directly from Theorem 4.5 and Corollary 4.6.

Lemma 5.2. *Let I be a unary indicator.*

- *If the corresponding relation $\preceq_I$ is Pareto-compliant, then $\preceq_I$ contains no deteriorative cycles.*
- *If the corresponding relation $\preceq_I$ is strictly Pareto-compliant, then $\preceq_{Alg_4} = \preceq_I$ contains no deteriorative cycles.*

5.1. The hypervolume indicator with a fixed reference point

Lemma 5.2 shows that unary indicators can, in a natural way, help to avoid the problem of cyclic behavior. It also shows why the property of Pareto-compliance is especially important for unary indicators. Unfortunately, there is currently only one unary indicator known that is Pareto-compliant [32], that is, given two sets A and B the indicator values A higher than B if A dominates B. This is the hypervolume indicator. For minimization problems, it measures the volume of the dominated portion of the objective space relative to a reference point $R = (R_1, R_2, \ldots, R_d) \in \mathbb{R}^d$ that lies above the Pareto front.

For our following investigations, we assume that the reference point does not change during the run of the algorithm. In our setting, where an indicator should be minimized, the hypervolume indicator of a set of solutions $A \in 2^X$ can be defined as

$$I_{\mathrm{HYP}}^R(A) := -\mathrm{VOL}\left(\bigcup_{x \in A} [f_1(x), R_1] \times \cdots \times [f_d(x), R_d] \right)$$

with $\mathrm{VOL}(\cdot)$ being the usual Lebesgue measure. Note that minimizing $I_{\mathrm{HYP}}^R(A)$ is equivalent to maximizing the volume covered by A with respect to the reference point R. As the reference point R is fixed in this section, we just set $I_{\mathrm{HYP}} := I_{\mathrm{HYP}}^R$. The hypervolume indicator was first introduced for performance assessment in multiobjective optimization by Zitzler and Thiele [30] and hypervolume-based optimizers have become very popular in recent years (see e.g. [4,15,28]). Besides being the only known Pareto-compliant indicator, another distinctive theoretical property is that the worst-case approximation factor of all possible Pareto fronts obtained by any hypervolume-optimal set of fixed size μ is asymptotically equal to the best worst-case approximation factor achievable by any set of size μ, namely $\Theta(1/\mu)$ for additive approximation and $1+\Theta(1/\mu)$ for multiplicative approximation [7,6].

The problem with the hypervolume indicator is that it is computationally expensive, i.e., the runtime for the computation of the hypervolume for a given set of search points grows exponentially with the number of objectives [5]. In contrast, the test of whether $A \preceq_{dom} B$ holds can always be done in time polynomial in the size of the given two sets and the number of objectives.

The following theorem describes another nice property of the hypervolume indicator.

Theorem 5.3. *Let $A, B \in 2^X$. If $A \equiv_{dom} B$ then $I_{\mathrm{HYP}}(A) = I_{\mathrm{HYP}}(B)$ holds.*

Proof. Let $A \equiv_{dom} B$. Then Lemma 2.3 implies $\mathrm{Min}(f(A), \preceq_{Par}) = \mathrm{Min}(f(B), \preceq_{Par})$. As the hypervolume of a given set of points is only determined by its set of minimal elements in the objective space, this implies $\mathrm{HYP}(A) = \mathrm{HYP}(B)$. $\square$

Note that the above theorem actually holds not only for HYP, but for all unary indicators whose value depends only on the minimal elements in the objective space.

5.2. The hypervolume indicator with a dynamically changing reference point

Some variants of hypervolume-based multi-objective evolutionary algorithms such as the SMS-EMOA [4] work with a dynamically changing reference point. Convergence rates of a $(1+1)$-SMS-EMOA have been recently examined in [3]. The authors of this paper showed that the $(1+1)$-SMS-EMOA has a sub-linear convergence rate for strongly convex quadratic

objective functions with two objectives. In the following discussion, we show that the $(1+1)$-SMS-EMOA working on at least three objectives may encounter the problem of deteriorative cycles. This matches the experimental findings of Judt et al. [17], who observed that the dominated hypervolume of SMS-EMOA with incorporated reference point adaptation can decrease in the course of the optimization process.

In the SMS-EMOA, the reference point is updated in each iteration of the algorithm and depends on the current set of search points.

Let S be a given set of search points. Then the reference point $R(S) = (R_1, R_2, \ldots, R_d) \in \mathbb{R}^d$ is given by

$$R_i(S) = 1 + \max_{x \in S} f_i(x), \quad 1 \le i \le d.$$

This leads to a new relation $\preceq_{I_{\text{HYP}}^{dyn}}$ on 2^X, defined as

$$A \preceq_{I_{\text{HYP}}^{dyn}} B :\Leftrightarrow \left(I_{\text{HYP}}^{R(A \cup B)}(A) \le I_{\text{HYP}}^{R(A \cup B)}(B) \right).$$

The hypervolume indicator working with a dynamically changing reference point is still Pareto compliant. This is a direct consequence of the Pareto compliance of the hypervolume indicator for a fixed reference point. It holds as the set under consideration completely determines the reference point to work with.

Theorem 5.4. *Let* $A, B \in 2^X$. *If* $A \preceq_{dom} B$ *then* $A \preceq_{I_{\text{HYP}}^{dyn}} B$.

However, by changing the reference point, we lose transitivity, which can lead to the problem of deteriorative cycles. We will now show that a changing reference point may produce the problem of deteriorative cycles.

Theorem 5.5. *For sets of size one in more than two dimensions the hypervolume based evolutionary algorithms using a dynamically changing reference point may encounter deteriorative cycles.*

Proof. We choose four search points $a, b, c, d \in X$ with objective vectors $f(a) = (2, 2, 4), f(b) = (1, 2, 4), f(c) = (3, 4, 1)$, and $f(d) = (7, 1, 1)$ and show that these four points build a deteriorative cycle for $\preceq_{I_{\text{HYP}}^{dyn}}$. As we are looking at minimization, b clearly dominates a, that is, $\{b\} \prec_{dom} \{a\}$. We further observe $\{a\} \prec_{I_{\text{HYP}}^{dyn}} \{c\}$ as $R(\{a, c\}) = (4, 5, 5)$ and

$$I_{\text{HYP}}^{R(\{a,c\})}(\{a\}) = -2 \cdot 3 \cdot 1 = -6 <= -4 = -1 \cdot 1 \cdot 4 = I_{\text{HYP}}^{R(\{a,c\})}(\{c\}).$$

Also $\{c\} \prec_{I_{\text{HYP}}^{dyn}} \{d\}$ as $R(\{c, d\}) = (8, 5, 2)$ and

$$I_{\text{HYP}}^{R(\{c,d\})}(\{c\}) = -5 \cdot 1 \cdot 1 = -5 <= -4 = -1 \cdot 4 \cdot 1 = I_{\text{HYP}}^{R(\{c,d\})}(\{d\}).$$

The deteriorative cycle is then closed by $\{d\} \prec_{I_{\text{HYP}}^{dyn}} \{b\}$ as $R(\{d, b\}) = (8, 3, 5)$ and

$$I_{\text{HYP}}^{R(\{d,b\})}(\{d\}) = -1 \cdot 2 \cdot 4 = -8 <= -7 = -7 \cdot 1 \cdot 1 = I_{\text{HYP}}^{R(\{d,b\})}(\{b\}).$$

Overall, this shows

$$\{b\} \prec_{dom} \{a\} \prec_{I_{\text{HYP}}^{dyn}} \{c\} \prec_{I_{\text{HYP}}^{dyn}} \{d\} \prec_{I_{\text{HYP}}^{dyn}} \{b\},$$

which finishes the proof. □

We have seen that the hypervolume-based multi-objective evolutionary algorithms working with a fixed reference point during the whole run of the algorithm do not encounter the problem of deteriorative cycles. This is different for hypervolume-based evolutionary algorithms such as the SMS-EMOA [4] that choose the reference point in each step depending on the current set of search points. Even simple variants of these algorithms such as $(1+1)$-SMS-EMOA [3] encounter the problem of deteriorative cycles if the number of objectives is at least three and therefore may not converge.

6. Binary indicators

Though the hypervolume indicator is currently the only known Pareto-compliant unary indicator, there are several Pareto-compliant binary indicators. This give some hope of finding a Pareto-compliant indicator that also contains no deteriorative cycles but is computationally less expensive than the hypervolume indicator.

Binary indicator functions assign to pairs of sets a real number that in some way reflects their relative performance, i.e.,

$$I_2 : 2^X \times 2^X \to \mathbb{R}.$$

To define a relation based on a binary indicator function, we use the following definition.

Definition 6.1. For a binary indicator I_2 we set

$$A \preceq_{I_2} B :\Leftrightarrow I_2(A, B) \le I_2(B, A),$$
$$A \prec_{I_2} B :\Leftrightarrow I_2(A, B) < I_2(B, A).$$

Zitzler et al. [32] mention four Pareto-compliant binary indicators: the multiplicative ε-indicator, the additive ε-indicator, the coverage indicator [30], and the binary hypervolume indicator. However, binary indicators are not transitive in general. Therefore there is no equivalent of Lemma 5.2 for binary indicators.

6.1. ε-indicator

In the following, we focus on the ε-indicators, which are very popular in evolutionary multi-objective optimization. We follow the definitions of Zitzler et al. [32] for the multiplicative and additive ε-dominance relation. We assume $\varepsilon > 0$. Recall that the objective space is non-negative, that is, $\mathbb{R}^d_{\geq 0}$.

Definition 6.2. A search point $a \in X$ is said to multiplicatively ε-dominate another search point $b \in X$ written as $a \preceq_{\varepsilon*} b$, if and only if

$$f_i(a) \leq \varepsilon f_i(b) \quad \text{for all } 1 \leq i \leq n.$$

The binary multiplicative ε-indicator $I_{\varepsilon*}$ on $2^X \times 2^X$ is

$$I_{\varepsilon*}(A, B) := \max_{b \in B} \min_{a \in A} \max_{1 \leq i \leq n} f_i(a) / f_i(b).$$

Definition 6.3. A search point $a \in X$ is said to additively ε-dominate another search point $b \in X$ written as $a \preceq_{\varepsilon+} b$, if and only if

$$f_i(a) \leq \varepsilon + f_i(b) \quad \text{for all } 1 \leq i \leq n.$$

The binary additive ε-indicator $I_{\varepsilon+}$ on $2^X \times 2^X$ is

$$I_{\varepsilon+}(A, B) := \max_{b \in B} \min_{a \in A} \max_{1 \leq i \leq n} f_i(a) - f_i(b).$$

We will analyze the corresponding relations $\preceq_{I_{\varepsilon*}}$ and $\preceq_{I_{\varepsilon+}}$ as defined in Definition 6.1. To simplify the notation, we set

$$\preceq_{\varepsilon*} := \preceq_{I_{\varepsilon*}} \quad \text{and} \quad \preceq_{\varepsilon+} := \preceq_{I_{\varepsilon+}}.$$

Let further $\prec_{\varepsilon*}$ and $\prec_{\varepsilon+}$ have their obvious meaning. We can show that $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ are strictly Pareto-compliant.

Lemma 6.4. $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ are strictly Pareto-compliant.

Proof. Let $A \prec_{dom} B$. Then by Definition 6.2, $I_{\varepsilon*}(A, B) \leq 1$, and $I_{\varepsilon*}(B, A) > 1$ and therefore $A \prec_{\varepsilon*} B$. Analogously by Definition 6.2, $I_{\varepsilon+}(A, B) \leq 0$ and $I_{\varepsilon+}(B, A) > 0$ and therefore $A \prec_{\varepsilon+} B$. This shows that $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ are Pareto-compliant.

In order to prove that they are also strictly Pareto-compliant, let $A \equiv_{dom} B$. Then by Lemma 2.3, $\text{Min}(f(A), \preceq_{Par}) = \text{Min}(f(B), \preceq_{Par})$ and therefore $I_{\varepsilon*}(A, B) = I_{\varepsilon*}(B, A) = 1$ and $I_{\varepsilon+}(A, B) = I_{\varepsilon+}(B, A) = 0$, which is equivalent to $A \equiv_{\varepsilon*} B$ and $A \equiv_{\varepsilon+} B$. $\square$

This shows that $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ are strictly Pareto-compliant. Therefore by Corollary 4.6 it suffices to show that they are also transitive in order to prove that they contain no deteriorative cycle. Unfortunately, in most cases this does not hold.

In the remainder of this section, we examine for what sets the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ contain deteriorative cycles. More precisely, we will prove the following dichotomy.

Theorem 6.5. Let $s \geq 1$ and $d \geq 2$. Then the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ restricted to sets of size $\leq s$ in d dimensions do not contain deteriorative cycles if and only if $s = 1$ and $d = 2$.

This implies that the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ contain no deteriorative cycles only in the simplest case of singleton sets in two dimensions. We prove Theorem 6.5 in the following Lemmas 6.6–6.8.

6.2. Nonexistence of deteriorative cycles for sets of size one in two dimensions

We first give a positive result and show that the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ do not contain deteriorative cycles if we consider only singleton sets in two dimensions.

Lemma 6.6. For sets of size one in two dimensions, the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ do not contain deteriorative cycles.

Proof. According to Theorem 4.5 and Lemma 6.4, it suffices to show that $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ are transitive. Let us consider three arbitrary points $a, b, c \in X$ with $f(a) = (a_x, a_y), f(b) = (b_x, b_y)$ and $f(c) = (c_y, c_y)$. We want to show that

$$\big(\!\big(\{a\} \preceq_{\varepsilon*} \{b\}\big) \wedge \big(\{b\} \preceq_{\varepsilon*} \{c\}\big)\big) \Rightarrow \{a\} \preceq_{\varepsilon*} \{c\}. \tag{1}$$

By definition,

$$\{a\} \preceq_{\varepsilon*} \{b\}$$
$$\Leftrightarrow I_{\varepsilon*}(\{a\}, \{b\}) \leq I_{\varepsilon*}(\{b\}, \{a\})$$
$$\Leftrightarrow \left(\max\left\{ \frac{a_x}{b_x}, \frac{a_y}{b_y} \right\} \leq \max\left\{ \frac{b_x}{a_x}, \frac{b_y}{a_y} \right\} \right).$$

Analogously,

$$\{b\} \preceq_{\varepsilon*} \{c\}$$
$$\Leftrightarrow I_{\varepsilon*}(\{b\}, \{c\}) \le I_{\varepsilon*}(\{c\}, \{b\})$$
$$\Leftrightarrow \left(\max\left\{ \frac{b_x}{c_x}, \frac{b_y}{c_y} \right\} \le \max\left\{ \frac{c_x}{b_x}, \frac{c_y}{b_y} \right\} \right).$$

We first observe that if $(a \preceq b) \vee (b \preceq c)$, then in this case, $\{a\} \preceq_{\varepsilon*} \{b\}$ and $\{b\} \preceq_{\varepsilon*} \{c\}$ implies $\{a\} \preceq_{\varepsilon*} \{c\}$.

We now work under the assumption that $(a \parallel b) \wedge (b \parallel c)$. As $(a \parallel b) \wedge (b \parallel c)$, one of the terms in each maximum computation above is greater than 1 and one term is smaller than 1.

Without loss of generality, we can assume $\frac{a_x}{b_x} > 1$ and $\frac{a_y}{b_y} < 1$. The opposite case $\frac{a_x}{b_x} < 1$ and $\frac{a_y}{b_y} > 1$ can be handled in an analogous way by symmetric arguments.

As we want to show Eq. (1), ($\{a\} \preceq_{\varepsilon*} \{b\}$) implies that $\frac{a_x}{b_x} \le \frac{b_y}{a_y}$. We now distinguish the two cases (1) $\left(\frac{b_x}{c_x} > 1 \wedge \frac{b_y}{c_y} < 1 \right)$ and (2) $\left(\frac{b_x}{c_x} < 1 \wedge \frac{b_y}{c_y} > 1 \right)$.

(1) If $\frac{b_x}{c_x} > 1 \wedge \frac{b_y}{c_y} < 1$, then ($\{b\} \preceq_{\varepsilon*} \{c\}$) implies that $\frac{b_x}{c_x} \le \frac{c_y}{b_y}$, and we get

$$\frac{a_x}{b_x} \cdot \frac{b_x}{c_x} = \frac{a_x}{c_x} \le \frac{b_y}{a_y} \cdot \frac{c_y}{b_y} = \frac{c_y}{a_y}.$$

Since both sides of the inequality are greater than 1, we get ($\{a\} \preceq_{\varepsilon*} \{c\}$).

(2) If $\frac{b_x}{c_x} < 1 \wedge \frac{b_y}{c_y} > 1$, then ($\{b\} \preceq_{\varepsilon*} \{c\}$) implies that $\frac{b_y}{c_y} \le \frac{c_x}{b_x}$, and we get

$$\frac{a_x}{b_x} \cdot \frac{b_y}{c_y} \le \frac{b_y}{a_y} \cdot \frac{c_x}{b_x}.$$

We multiply both sides with $\frac{b_x}{c_x} \cdot \frac{c_y}{b_y} < 1$ and get

$$\frac{a_x}{c_x} \le \frac{c_y}{a_y}.$$

We get three subcases depending on the ratios $\frac{a_x}{c_x}$ and $\frac{c_y}{a_y}$.

(a) If $\frac{a_x}{c_x} \ge 1$ then $1 \le \frac{a_x}{c_x} \le \frac{c_y}{a_y}$ implies ($\{a\} \preceq_{\varepsilon*} \{c\}$).

(b) If $\frac{a_x}{c_x} < 1$ and $\frac{c_y}{a_y} < 1$ then $1 < \frac{a_y}{c_y} \le \frac{c_x}{a_x}$ and this implies ($\{a\} \preceq_{\varepsilon*} \{c\}$).

(c) If $\frac{a_x}{c_x} < 1$ and $\frac{c_y}{a_y} > 1$ then $a \preceq c$ and therefore ($\{a\} \preceq_{\varepsilon*} \{c\}$).

This shows Eq. (1) and the claim of the theorem for the multiplicative ε-relation. The proof for the additive ε-relation can be obtained in an analogous way by considering the differences $\alpha - \beta$ instead of the fractions $\frac{\alpha}{\beta}$. $\square$

6.3. Existence of deteriorative cycles for sets of size one in three dimensions

In Lemma 6.6 we showed that for sets of size one in two dimensions the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$ do not contain deteriorative cycles. We now prove that even for three dimensions this no longer holds.

Lemma 6.7. *For sets of size one in more than two dimensions the relations $\preceq_{\varepsilon*}$ and $x \preceq_{\varepsilon+}$ contain deteriorative cycles.*

Proof. We choose four points $a, b, c, d \in X$ with $f(a) := (1, 2, 4), f(b) := (1, 1, 4), f(c) := (3, 3, 1), f(d) := (4, 1, 2)$ and show that these four points build a deteriorative cycle for both ε-dominance relations. Let us first look at the multiplicative ε-dominance relation. As we are looking at minimization problems, b clearly dominates a, that is, $\{b\} \prec_{dom} \{a\}$. On the other hand, $\{c\} \prec_{\varepsilon*} \{b\}$, since

$$I_{\varepsilon*}(\{c\}, \{b\}) = 3 < 4 = I_{\varepsilon*}(\{b\}, \{c\}).$$

Also, $\{d\} \prec_{\varepsilon*} \{c\}$, since

$$I_{\varepsilon*}(\{d\}, \{c\}) = 2 < 3 = I_{\varepsilon*}(\{c\}, \{d\}).$$

The deteriorative cycle is then closed by $\{a\} \prec_{\varepsilon*} \{d\}$, since

$$I_{\varepsilon*}(\{a\}, \{d\}) = 2 < 4 = I_{\varepsilon*}(\{d\}, \{a\}).$$

Overall, $\{a\} \prec_{\varepsilon*} \{d\} \prec_{\varepsilon*} \{c\} \prec_{\varepsilon*} \{b\} \prec_{dom} \{a\}$.
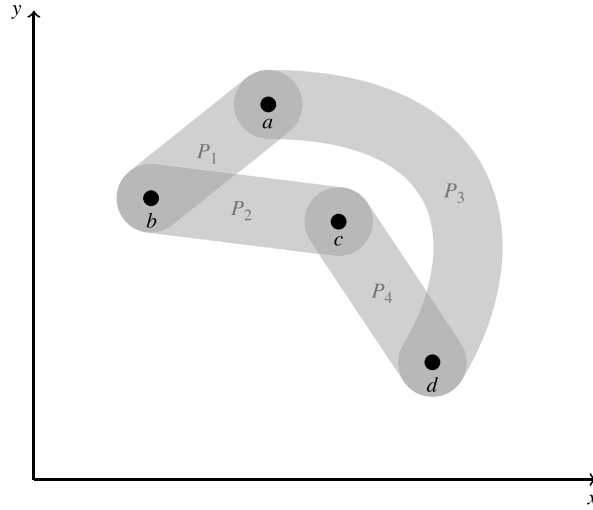
Fig. 2. Illustration of the sets used in the proof of Lemma 6.8.

It remains to show that the same points form a deteriorative cycle for the additive ε-dominance relation. It is easy to see that

$$I_{\varepsilon+}(\{c\}, \{b\}) = 2 < 3 = I_{\varepsilon+}(\{b\}, \{c\}),$$
$$I_{\varepsilon+}(\{d\}, \{c\}) = 1 < 2 = I_{\varepsilon+}(\{c\}, \{d\}),$$
$$I_{\varepsilon+}(\{a\}, \{d\}) = 2 < 3 = I_{\varepsilon+}(\{d\}, \{a\}),$$

and therefore $\{a\} \prec_{\varepsilon+} \{d\} \prec_{\varepsilon+} \{c\} \prec_{\varepsilon+} \{b\} \prec_{dom} \{a\}$, which finishes the proof. □

6.4. Existence of deteriorative cycles for sets of size two in two dimensions

To complete the dichotomy of the relations $\preceq_{\varepsilon*}$ and $\preceq_{\varepsilon+}$, it remains to prove that in two dimensions as well, sets of size more than one give deteriorative cycles.

Lemma 6.8. *For sets of size more than one in two dimensions, the relations $\preceq_{\varepsilon*}$ and $x \preceq_{\varepsilon+}$ contain deteriorative cycles.*

Proof. We choose $a, b, c, d \in X$ such that

$$
\begin{aligned}
P_1 &:= \{a, b\}, &\quad \text{with } f(a) &:= (10, 16), \\
P_2 &:= \{c, b\}, &\quad f(b) &:= (5, 12), \\
P_3 &:= \{a, d\}, &\quad f(c) &:= (13, 11), \\
P_4 &:= \{d, c\}, &\quad f(d) &:= (17, 5).
\end{aligned}
$$

Fig. 2 gives an illustration of the four sets.

We want to show that these four sets P_1, P_2, P_3, P_4 form a deteriorative cycle. It is easy to see that $P_1 \prec_{dom} P_2$. We first examine the multiplicative ε-dominance relation and prove that $P_1 \prec_{dom} P_2 \prec_{\varepsilon*} P_3 \prec_{\varepsilon*} P_4 \prec_{\varepsilon*} P_1$.

Observe that $P_2 \prec_{\varepsilon*} P_3$ holds, as

$$
\begin{aligned}
I_{\varepsilon*}(P_3, P_2) &= \max\left\{ \min\left\{ \max\left\{ \frac{10}{13}, \frac{16}{11}, \right\}, \max\left\{ \frac{17}{13}, \frac{5}{11}, \right\} \right\}, \right. \\
&\qquad \left. \min\left\{ \max\left\{ \frac{10}{5}, \frac{16}{12}, \right\}, \max\left\{ \frac{17}{5}, \frac{5}{12}, \right\} \right\} \right\} \\
&= \max\{17/13, 2\} = 2 < 2.2 = \max\{3/4, 11/5\} \\
&= \max\left\{ \min\left\{ \max\left\{ \frac{13}{10}, \frac{11}{16}, \right\}, \max\left\{ \frac{5}{10}, \frac{12}{16}, \right\} \right\}, \right. \\
&\qquad \left. \min\left\{ \max\left\{ \frac{13}{17}, \frac{11}{5}, \right\}, \max\left\{ \frac{5}{17}, \frac{12}{5}, \right\} \right\} \right\} \\
&= I_{\varepsilon*}(P_2, P_3),
\end{aligned}
$$

and $P_3 \prec_{\varepsilon*} P_4$ holds, as

$$
\begin{aligned}
I_{\varepsilon*}(P_4, P_3) &= \max\left\{\min\left\{\max\left\{\frac{17}{10}, \frac{5}{16},\right\}, \max\left\{\frac{13}{10}, \frac{11}{16},\right\}\right\},\right. \\
&\qquad\left. \min\left\{\max\left\{\frac{17}{17}, \frac{5}{5},\right\}, \max\left\{\frac{13}{17}, \frac{11}{5},\right\}\right\}\right\} \\
&= \max\left\{13/10, 1\right\} = 1.3 < 1.308\ldots = \max\left\{1, 17/13\right\} \\
&= \max\left\{\min\left\{\max\left\{\frac{10}{17}, \frac{16}{5},\right\}, \max\left\{\frac{17}{17}, \frac{5}{5},\right\}\right\},\right. \\
&\qquad\left. \min\left\{\max\left\{\frac{10}{13}, \frac{16}{11},\right\}, \max\left\{\frac{17}{13}, \frac{5}{11},\right\}\right\}\right\} \\
&= I_{\varepsilon*}(P_3, P_4).
\end{aligned}
$$

It remains to show $P_4 \prec_{\varepsilon*} P_1$, which holds since

$$
\begin{aligned}
I_{\varepsilon*}(P_1, P_4) &= \max\left\{\min\left\{\max\left\{\frac{10}{17}, \frac{16}{5},\right\}, \max\left\{\frac{5}{17}, \frac{12}{5},\right\}\right\},\right. \\
&\qquad\left. \min\left\{\max\left\{\frac{10}{13}, \frac{16}{11},\right\}, \max\left\{\frac{5}{13}, \frac{12}{11},\right\}\right\}\right\} \\
&= \max\left\{12/5, 12/11\right\} = 2.4 < 2.6 = \max\left\{13/10, 13/5\right\} \\
&= \max\left\{\min\left\{\max\left\{\frac{17}{10}, \frac{5}{16},\right\}, \max\left\{\frac{13}{10}, \frac{11}{16},\right\}\right\},\right. \\
&\qquad\left. \min\left\{\max\left\{\frac{17}{5}, \frac{5}{12},\right\}, \max\left\{\frac{13}{5}, \frac{11}{12},\right\}\right\}\right\} \\
&= I_{\varepsilon*}(P_4, P_1).
\end{aligned}
$$

This shows the claim for the multiplicative ε-dominance relation. We now prove that the same sets also contain a deteriorative cycle in the case of the additive ε-dominance relation. We have $P_2 \prec_{\varepsilon+} P_3$ as

$$
\begin{aligned}
I_{\varepsilon+}(P_3, P_2) &= \max\{\min\{\max\{-3, 5\}, \max\{4, -6\}\}, \\
&\qquad \min\{\max\{5, 4\}, \max\{12, -7\}\}\} \\
&= \max\{4, 5\} = 5 < 6 = \max\{-4, 6\} \\
&= \max\{\min\{\max\{3, -5\}, \max\{-5, -4\}\}, \\
&\qquad \min\{\max\{-4, 6\}, \max\{-12, 7\}\}\} \\
&= I_{\varepsilon+}(P_2, P_3),
\end{aligned}
$$

and $P_3 \prec_{\varepsilon*} P_4$ as

$$
\begin{aligned}
I_{\varepsilon+}(P_4, P_3) &= \max\{\min\{\max\{7, -11\}, \max\{3, -5\}\}, \\
&\qquad \min\{\max\{0, 0\}, \max\{-4, 6\}\}\} \\
&= \max\{3, 0\} = 3 < 4 = \max\{0, 4\} \\
&= \max\{\min\{\max\{-7, 11\}, \max\{0, 0\}\}, \\
&\qquad \min\{\max\{-3, 5\}, \max\{4, -6\}\}\} \\
&= I_{\varepsilon+}(P_3, P_4),
\end{aligned}
$$

and $P_4 \prec_{\varepsilon*} P_1$ as

$$
\begin{aligned}
I_{\varepsilon+}(P_1, P_4) &= \max\{\min\{\max\{-7, 11\}, \max\{-12, 7\}\}, \\
&\qquad \min\{\max\{-3, 5\}, \max\{-8, 1\}\}\} \\
&= \max\{7, 1\} = 7 < 8 = \max\{3, 8\} \\
&= \max\{\min\{\max\{7, -11\}, \max\{3, -5\}\}, \\
&\qquad \min\{\max\{12, -7\}, \max\{8, -1\}\}\} \\
&= I_{\varepsilon+}(P_4, P_1).
\end{aligned}
$$

This shows $P_1 \prec_{dom} P_2 \prec_{\varepsilon+} P_3 \prec_{\varepsilon+} P_4 \prec_{\varepsilon+} P_1$ and finishes the proof.   □

Algorithm 5: Framework for RLS

---

1 Choose x uniformly at random from $\{0, 1\}^n$.;

2 **repeat**

3 $\quad$ Flip one bit chosen uniformly at random from x to obtain an offspring x';

4 $\quad$ **if** selection favors x' over x **then** $x := x'$;

5 **until** stopped;

---

7. Impact on the runtime behavior

In this section, we point out the impact of deteriorative cycles by rigorous runtime analysis. We study simple randomized search heuristics and show how they significantly differ in their runtime behavior when incurring or avoiding deteriorative cycles.

We consider different variants of randomized local search, which work with one single solution at a time. They differ from each other in the way they accept new search points. The framework of the algorithm is shown in Algorithm 5 and motivated by studies carried out in [13].

Our algorithms differ from one another with respect to the selection mechanism, i. e., when to favor x' over x. We study the runtime behavior of different randomized local search variants in the asymptotic sense and measure runtime by the number of iterations of the repeat loop until an algorithm has produced an optimal solution for the first time. As we are interested in the asymptotic behavior, we assume that the problem size is large enough. For our analysis, it suffices to assume that $n \geq 2$ holds.

The first selection methods are taken from Giel and Lehre [13] and take into account only the dominance relation between search points.

- $\text{RLS}_{weakest}$: Favor x' over x iff $(x' \succeq_{dom} x) \lor (x' \parallel_{dom} x)$
- RLS_{weak}: Favor x' over x iff $(x' \succeq_{dom} x)$
- RLS_{strong}: Favor x' over x iff $(x' \succ_{dom} x)$.

In addition, we examine the approach of using the hypervolume indicator for a fixed reference point r. We already know that this approach avoids the problem of deteriorative cycles. We choose the reference point $r = (0, 0)$ and consider bi-objective problems where both objectives have to be maximized. The hypervolume $\text{HYP}(x)$ of a solution x is given by $\text{HYP}(x) = f_1(x) \cdot f_2(x)$ when considering a problem $f = (f_1, f_2)$ consisting of two objectives that have to be maximized. Selecting for our algorithm called RLS_{HYP} is done in the following way.

- RLS_{HYP}: Favor x' over x iff $\text{HYP}(x') \geq \text{HYP}(x)$.

To point out the differences of the introduced algorithms with respect to the runtime behavior, we investigate a function motivated by the function Short-Path with constant fitness (SPC) introduced by Jansen and Wegener [16]. To be compliant with the definitions in this paper, we consider a problem where both objectives have to be maximized.

Let

$$PL = \{1^i 0^{n-i} \mid 0 \leq i \leq n - 1\}$$

and

$$LO(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$$

be the number of leading ones in a search point x. The function Multi-Objective Short-Path with incomparable fitness (MOSPI) is defined as

$$MOSPI(x) = \begin{cases} (|x|_0, |x|_0) & \text{if } x \notin PL \cup \{1^n\} \\ (n + LO(x), n + LO(x)) & \text{if } x \in PL \text{ and } LO(x) \text{ is even} \\ (n + LO(x) + 2, n + LO(x) - 2) & \text{if } x \in PL \text{ and } LO(x) \text{ is odd} \\ (3n, 3n) & \text{if } x = 1^n. \end{cases}$$

MOSPI has the following structure, which is the same as the one of the function SPC when considering only direct Hamming neighbors. As long as the current solution of the algorithm is not in $PL \cup \{1^n\}$, it is always beneficial to maximize the number of zeros. Such steps are strict improvements according to the dominance relation and therefore accepted by each of our algorithms. After having obtained a solution of $PL \cup \{1^n\}$ which is not optimal, $\text{RLS}_{weakest}$ may switch between incomparable solutions having Hamming distance 1, whereas RLS_{weak} and RLS_{strong} do not accept incomparable solutions. The optimal solution, which is the search point 1^n, can only be obtained by switching between incomparable solutions of PL.

This leads to the following result for RLS_{weak} and RLS_{strong} which shows that they are, with high probability, not able to achieve an optimal solution. For the proof we can follow the ideas of Jansen and Wegener [16] for the analysis of $(1+1)\,\text{EA}^*$ (for the results of RLS_{weak} and RLS_{strong}) and $(1+1)\,\text{EA}$ (for the results of $\text{RLS}_{weakest}$) on the function SPC. We will state the theorems and present the main ideas of [16] leading to the runtime results of the mentioned algorithms.

Theorem 7.1. *With probability* $1 - e^{-\Omega(n)}$ *the time until* RLS_{weak} *and* RLS_{strong} *have obtained an optimal solution of MOSPI is infinite.*

Proof. The initial solution has at most $2n/3$ 1-bits with probability $1 - e^{-\Omega(n)}$ using Chernoff bounds [22]. After this, RLS_{weak} and RLS_{strong} can only accept a solution with a larger number of ones when producing a solution of $PL \cup \{1^n\}$ for the first time. This solution has with probability $1 - e^{-\Omega(n)}$ a Hamming distance greater than 1 to the optimal solution 1^n and therefore both algorithms are not able to achieve the optimum. □

$RLS_{weakest}$ has the ability to switch between incomparable solutions, which enables the algorithm to perform a random walk on PL. This leads to a runtime bound of $\Theta(n^3)$ for obtaining an optimal solution.

Theorem 7.2. *The expected time until* $RLS_{weakest}$ *has obtained an optimal solution of MOSPI is* $\Theta(n^3)$.

Proof. A solution of $PL \cup \{1^n\}$ is produced after an expected number of $O(n \log n)$ steps, which can be shown by the method of fitness-based partitions (see e. g. [23]). Afterwards the algorithm increases and decreases with equal probability the number of ones. This leads to a random walk on PL whereas the number of search points of $PL \cup \{1^n\}$ is $n + 1$. The expected waiting time for one step is $\Theta(n)$ and the expected number of steps to reach the optimal solution is $\Theta(n^2)$, following random walk arguments (see proof of Theorem 1 in [8]), which leads to the claimed bound of $\Theta(n^3)$. □

The random walk slows down the process, as it has to cope with deteriorative cycles. Being able to avoid these cycles and staying compliant with the dominance relation enables the algorithm using the hypervolume indicator to reach the optimal solution more quickly.

Theorem 7.3. *The expected time until* RLS_{HYP} *has obtained an optimal solution of MOSPI is* $O(n^2)$.

Proof. Similar to RLS_{weak}, RLS_{HYP} reaches a solution of $PL \cup \{1^n\}$ after an expected number of $O(n \log n)$ steps. We consider the value of the hypervolume of a point of PL. If $LO(x)$ is even, the hypervolume is $(n + LO(x))^2$. If LO(x) is odd the hypervolume is $(n + LO(x) + 2) \cdot (n + LO(x) - 2) = (n + LO(x))^2 - 4$.

Assume that $x \in PL$ and let x' be the solution of $PL \cup \{1^n\}$ with $LO(x') = LO(x) + 1$. We claim that $HYP(x') > HYP(x)$.

If LO(x) is even, then

$$
\begin{aligned}
HYP(x) = (n + LO(x))^2 &< (n + LO(x) + 1)^2 - 4 \\
&= (n + LO(x))^2 + 2(n + LO(x)) + 1 - 4 \\
&\leq HYP(x')
\end{aligned}
$$

as long as $n \geq 2$.

If LO(x) is odd, then

$$
\begin{aligned}
HYP(x) = (n + LO(x))^2 - 4 &< (n + LO(x) + 1)^2 \\
&= (n + LO(x))^2 + 2(n + LO(x)) + 1 \\
&\leq HYP(x').
\end{aligned}
$$

Furthermore, observe that if $x \in PL \cup \{1^n\}$ then $HYP(x) \geq n^2$ which is strictly greater then $HYP(x') \leq (n - 1)^2$ for $x' \notin PL \cup \{1^n\}$. This implies that if it has reached a solution $x \in PL$, then it only accepts a solution x' of $PL \cup \{1^n\}$ with $LO(x') \geq LO(x)$. The probability of producing a solution $x' \in PL \cup \{1^n\}$ with $LO(x') > LO(x)$ is $\Omega(1/n)$, as the leftmost 0-bit can be flipped as long as an optimal solution has not been obtained. The expected waiting time for such a step is therefore $O(n)$ and the number of improvements necessary to achieve the optimal solution is $O(n)$ as well. This shows an upper bound of $O(n^2)$ on the expected time until RLS_{HYP} has obtained an optimal solution of MOSPI. □

8. Conclusions

Evolutionary algorithms for multi-objective optimization search for a set of search points that is minimal with respect to the Pareto dominance relation on sets. This optimization goal has been made explicit recently in Zitzler et al. [31]. With this paper, we have contributed to the theoretical understanding of this optimization process by investigating the underlying relation that an evolutionary algorithm uses for optimization. First, we have shown how to choose a minimal set among the parents and children to build the next parent population. Our algorithms are similar to the method used in NSGA-II and SPEA2 and allow the incorporation of preferences into the computation of a minimal set.

Later on, we pointed out that algorithms that are solely based on the Pareto dominance relation may encounter deteriorative cycles if they can move between incomparable sets. This is due to the fact that the Pareto dominance relation is not a total relation. We have examined how such cycles can be avoided by using indicator functions on incomparable sets. Our studies show that if the total relation on which an algorithm works is Pareto compliant and transitive, then the relation does not contain a deteriorative cycle. Investigating the binary ε-indicator, which is Pareto compliant, we have shown that it is transitive only for very restricted cases and may lead to deteriorative cycles in general. Unary indicators are in a natural way transitive and therefore each unary indicator that is Pareto compliant fulfills our conditions. An indicator matching

the desired properties is the hypervolume indicator when a fixed reference point is used during the run of the algorithm. Therefore, our studies give a further justification for using this indicator that has become very popular in evolutionary multi-objective optimization. It remains an open problem to construct unary indicators that are Pareto compliant and cheaper to compute than the hypervolume indicators. This would help to guide the search with respect to the Pareto dominance relation on sets in a more efficient way.

Acknowledgments

References

[1] A. Auger, B. Doerr (Eds.), Theory of Randomized Search Heuristics: Foundations and Recent Developments, World Scientific, 2011.
[2] R. Berghammer, T. Friedrich, F. Neumann, Set-based multi-objective optimization, indicators, and deteriorative cycles, in: Proc. 12th annual Conference on Genetic and Evolutionary Computation, GECCO'10, ACM Press, 2010, pp. 495–502.
[3] N. Beume, M. Laumanns, G. Rudolph, Convergence rates of $(1+1)$ evolutionary multiobjective optimization algorithms, in: Proc. 11th International Conference Parallel Problem Solving from Nature, PPSN XI, in: Lecture Notes in Computer Science, vol. 6238, Springer, 2010, pp. 597–606.
[4] N. Beume, B. Naujoks, M.T.M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181 (2007) 1653–1669.
[5] K. Bringmann, T. Friedrich, Approximating the volume of unions and intersections of high-dimensional geometric objects, Computational Geometry: Theory and Applications 43 (2010) 601–610.
[6] K. Bringmann, T. Friedrich, The maximum hypervolume set yields near-optimal approximation, in: Proc. 12th annual Conference on Genetic and Evolutionary Computation, GECCO'10, ACM Press, 2010, pp. 511–518.
[7] K. Bringmann, T. Friedrich, Tight bounds for the approximation ratio of the hypervolume indicator, in: Proc. 11th International Conference Parallel Problem Solving from Nature, PPSN XI, in: LNCS, vol. 6238, Springer, 2010, pp. 607–616.
[8] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, E. Zitzler, On the effects of adding objectives to plateau functions, IEEE Trans. Evolutionary Computation 13 (3) (2009) 591–603.
[9] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, Chichester, UK, 2001.
[10] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolutionary Computation 6 (2) (2002) 182–197.
[11] S.G. Ficici, Monotonic solution concepts in coevolution, in: H.-G. Beyer, U.-M. O'Reilly (Eds.), GECCO, ACM, 2005, pp. 499–506.
[12] T. Friedrich, N. Hebbinghaus, F. Neumann, Plateaus can be harder in multi-objective optimization, Theoretical Computer Science 411 (6) (2010) 854–864.
[13] O. Giel, P.K. Lehre, On the effect of populations in evolutionary multi-objective optimisation, Evolutionary Computation 18 (3) (2010) 335–356.
[14] T. Hanne, On the convergence of multiobjective evolutionary algorithms, European Journal of Operational Research 117 (3) (1999) 553–564.
[15] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation 15 (2007) 1–28.
[16] T. Jansen, I. Wegener, Evolutionary algorithms – How to cope with plateaus of constant fitness and when to reject strings of the same fitness, IEEE Trans. Evolutionary Computation 5 (6) (2001) 589–599.
[17] L. Judt, O. Mersmann, B. Naujoks, Non-monotonicity of obtained hypervolume in 1-greedy S-metric selection, in: Proc. 21st International Conference on Multiple Criteria Decision Making, MCDM'11, 2011 (in press).
[18] J. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Feb. 2006.
[19] J.D. Knowles, D. Corne, Approximating the nondominated front using the pareto archived evolution strategy, Evolutionary Computation 8 (2) (2000) 149–172.
[20] J.D. Knowles, D. Corne, Properties of an adaptive archiving algorithm for storing nondominated vectors, IEEE Trans. Evolutionary Computation 7 (2003) 100–116.
[21] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary Computation 10 (3) (2003) 263–282.
[22] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
[23] F. Neumann, C. Witt, Bioinspired Computation in Combinatorial Optimization—Algorithms and Their Computational Complexity, Springer, 2010.
[24] E. Popovici, K.A.D. Jong, Monotonicity versus performance in co-optimization, in: I.I. Garibay, T. Jansen, R.P. Wiegand, A.S. Wu (Eds.), FOGA, ACM, 2009, pp. 151–170.
[25] G. Rudolph, Evolutionary search for minimal elements in partially ordered finite sets, in: Proc. 7th International Conference on Evolutionary Programming, EP'98, in: Lecture Notes in Computer Science, vol. 1447, Springer, 1998, pp. 345–353.
[26] O. Schütze, M. Laumanns, C.A.C. Coello, M. Dellnitz, E.-G. Talbi, Convergence of stochastic search algorithms to finite size pareto set approximations, Journal of Global Optimization 41 (4) (2008) 559–577.
[27] O. Schütze, M. Laumanns, E. Tantar, C.A.C. Coello, E.-G. Talbi, Computing gap free pareto front approximations with stochastic search algorithms, Evolutionary Computation 18 (1) (2010) 65–96.
[28] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration, in: Proc. Fourth International Conference on Evolutionary Multi-Criterion Optimization, EMO'07, in: LNCS, vol. 4403, Springer, 2007, pp. 862–876.
[29] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Proc. of EUROGEN 2001, International Center for Numerical Methods in Engineering, CIMNE, 2002, pp. 95–100.
[30] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evolutionary Computation 3 (1999) 257–271.
[31] E. Zitzler, L. Thiele, J. Bader, On set-based multiobjective optimization, IEEE Trans. Evolutionary Computation 14 (2010) 58–79.
[32] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V. Grunert da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evolutionary Computation 7 (2003) 117–132.