Combining Convergence and Diversity in Evolutionary Multiobjective Optimization

Marco Laumanns laumanns@tik.ee.ethz.ch
Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich, 8092 Zurich, Switzerland

Lothar Thiele thiele@tik.ee.ethz.ch
Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich, 8092 Zurich, Switzerland

Kalyanmoy Deb deb@iitk.ac.in
Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur, PIN 208 016, India

Eckart Zitzler zitzler@tik.ee.ethz.ch
Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich, 8092 Zurich, Switzerland

Abstract

Over the past few years, the research on evolutionary algorithms has demonstrated their niche in solving multiobjective optimization problems, where the goal is to find a number of Pareto-optimal solutions in a single simulation run. Many studies have depicted different ways evolutionary algorithms can progress towards the Pareto-optimal set with a widely spread distribution of solutions. However, none of the multiobjective evolutionary algorithms (MOEAs) has a proof of convergence to the true Pareto-optimal solutions with a wide diversity among the solutions. In this paper, we discuss why a number of earlier MOEAs do not have such properties. Based on the concept of ϵ-dominance, new archiving strategies are proposed that overcome this fundamental problem and provably lead to MOEAs that have both the desired convergence and distribution properties. A number of modifications to the baseline algorithm are also suggested. The concept of ϵ-dominance introduced in this paper is practical and should make the proposed algorithms useful to researchers and practitioners alike.

Keywords

Evolutionary algorithms, multiobjective optimization, convergence, preservation of diversity, ϵ-approximation, elitism, archiving.

1 Introduction

After the doctoral study of Schaffer (1984) on the *vector evaluated genetic algorithm* (VEGA), Goldberg's (1989) suggestion of the use of nondominated sorting along with a niching mechanism generated an overwhelming interest in *multiobjective evolutionary algorithms* (MOEAs). Initial MOEAs – *Multiobjective Genetic Algorithm* (MOGA) (Fonseca and Fleming, 1993), *Nondominated Sorting Genetic Algorithm* (NSGA) (Srinivas and Deb, 1994), and *Niched Pareto Genetic Algorithm* (NPGA) (Horn et al., 1994) – used Goldberg's suggestion in a straightforward manner: (i) the fitness of a solution was assigned using the extent of its domination in the population, and (ii) the diversity among solutions were preserved using a niching strategy. The above three studies have shown that different ways of implementing the above two tasks can all result in successful MOEAs. However, these algorithms could not ensure convergence to the true Pareto-optimal set since an operator for elite preservation was missing. Thus, the latter MOEAs mainly concentrated on how elitism could be introduced in an MOEA. This resulted in a number of advanced algorithms described in Section 3.2: SPEA, PAES, and NSGA-II. With the development of better algorithms, multiobjective evolutionary algorithms have also been used in a number of interesting application case studies (Zitzler et al., 2001).

What is severely lacking are studies related to theoretical convergence analysis with guaranteed spread of solutions. In this regard, Rudolph (1998b, 2001) and Rudolph and Agapie (2000) suggested a series of algorithms, all of which guarantee convergence, but do not address the following two aspects:

1. The convergent algorithms do not guarantee maintaining a spread of solutions.
2. The algorithms do not specify any time complexity for their convergence to the true Pareto-optimal set.

Although the second task is difficult to achieve even for simple objective functions (see Laumanns et al. (2002)) and also in the single-objective case, the first task is as important as the task of converging to the true Pareto-optimal set. Deb (2001) suggested a steady-state MOEA that attempts to maintain spread while converging to the true Pareto-optimal front. But there is no proof for its convergence properties. Knowles (2002) has analyzed two further possibilities: metric-based archiving and adaptive grid archiving. The metric-based strategy requires a function that assigns a scalar value to each possible approximation set reflecting its quality and fulfilling certain monotonicity conditions. Convergence is then defined as the achievement of a local optimum of the quality function. The adaptive grid archiving strategy implemented in PAES provably maintains solutions in some "critical" regions of the Pareto set once they have been found, but convergence can only be guaranteed for the solutions at the extremes of the Pareto set.

In this paper, we propose archiving/selection strategies that guarantee both progress towards the Pareto-optimal set and covering the whole range of nondominated solutions. The algorithms maintain a finite-sized archive of nondominated solutions that is iteratively updated in the presence of a new solution based on the concept of ϵ-dominance. The use of ϵ-dominance also makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ϵ value. The archiving algorithms suggested here are generic and enable convergence with a guaranteed spread of solutions.

In the remainder of the paper, we state the general structure of an iterative archive-based search procedure, which is usually used for multiobjective optimization. Thereafter, we briefly review the existing MOEAs and discuss why they do not have theoretical convergence as well as diversity-preservation properties at the same time. In Section 4 we formally define our concepts of ϵ-dominance and the corresponding ϵ-Pareto-optimal set as well as the new selection algorithms. Section 5 presents some simulation results to demonstrate the behavior of the new algorithms and to highlight the important differences to the existing approaches. In Section 6, various practically relevant extensions to the new approach are outlined and discussed. The proposed convergent MOEAs are interesting and should make MOEAs more useful and attractive to both theoreticians and practitioners.

Algorithm 1 Iterative search procedure

1: $t := 0$
2: $A^{(0)} := \emptyset$
3: **while** $terminate(A^{(t)}, t) = false$ **do**
4: $\quad t := t + 1$
5: $\quad f^{(t)} := generate()$ {generates new search point}
6: $\quad A^{(t)} := update(A^{(t-1)}, f^{(t)})$ {updates archive}
7: **end while**
8: **Output:** $A^{(t)}$

2 Structure of an Iterative Multiobjective Search Algorithm

The purpose of this section is to informally describe the problem we are dealing with. To this end, let us first give a template for a large class of iterative search procedures characterized by the generation of a sequence of search points and a finite memory.

The purpose of such algorithms is to find or approximate the Pareto set of the image set F of a vector valued function $h : X \to F$ defined over some domain X. In the context of multiobjective optimization, h, F, and X are often called the multi-valued objective function, the objective space, and the decision space, respectively.

An abstract description of a generic iterative search algorithm is given in Algorithm 1. The integer t denotes the iteration count, the n-dimensional vector $f^{(t)} \in F$ is the sample generated at iteration t, and the set $A^{(t)}$ will be called the archive at iteration t and should contain a representative subset of the samples in the objective space F generated so far. To simplify the notation, we represent samples by m-dimensional real vectors f where each coordinate represents one of the objective values. Additional information about the corresponding decision values could be associated to f but will be of no concern in this paper.

The purpose of the function *generate* is to generate a new solution in each iteration t, possibly using the contents of the old archive set $A^{(t-1)}$. The function $update$ gets the new solution $f^{(t)}$ and the old archive set $A^{(t-1)}$ and determines the updated one $A^{(t)}$. In general, the purpose of this sample storage is to gather useful information about the underlying search problem during the run. Its use is usually two-fold: On one hand, it is used to store the best solutions found so far; on the other hand, the search operator exploits this information to steer the search to promising regions.

This algorithm could be viewed as an evolutionary algorithm when the *generate* operator is associated with variation (recombination and mutation). However, we point out that all following investigations are equally valid for any kind of iterative process that can be described as Algorithm 1 and used for approximating the Pareto set of multiobjective optimization problems (e.g., simulated annealing, tabu search).

There are several reasons why the archive $A^{(t)}$ should be of constant size, independent of the number of iterations t. At first, the computation time grows with the number of archived solutions, for example, the function *generate* may use it for guiding the search, or it may simply be impossible to store all solutions as the physical memory is always finite. In addition, the value of presenting such a large set of solutions to a decision maker is doubtful in the context of decision support, instead one should provide him with a set of the best *representative* samples. Finally, in limiting the size of solution set, preference information could be used to steer the process to certain parts of the search space.

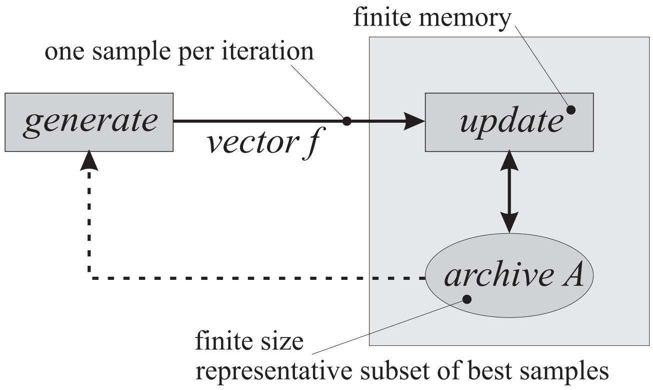

Figure 1: Representation of the generic search Algorithm 1.

This paper solely deals with the function *update*, i.e., with an appropriate handling of the archive. Because of the reasons described above, the corresponding algorithm should have the following properties (see also Figure 1):

- The algorithm is provided with one sample $f^{(t)}$ at each iteration, i.e., one at a time.
- It operates with finite memory. In particular, it cannot store all the samples submitted until iteration t.
- The algorithm should maintain a set $A^{(t)}$ of a limited size, which is independent of the iteration count t. The set should contain a representative subset of the best samples $f^{(1)}, ..., f^{(t)}$ received so far.

A clear definition of the term *representative subset* of the *best samples* will be given in Section 4.1. But according to the common notion of optimality in multiobjective optimization and the above discussion, it should be apparent that the archive $A^{(t)}$ should contain a subset of all Pareto vectors of the samples generated until iteration t. In addition, these selected Pareto vectors should represent the diversity of all Pareto vectors generated so far.

We will construct such an algorithm in Sections 4.2 and 4.3. Beforehand, existing approaches will be described.

3 Existing Multiobjective Algorithms and Their Limitations

Here, we discuss a number of archiving strategies that are suggested in the context of MOEAs. They can be broadly categorized into two categories depending on whether their focus lies on convergence or distribution quality.

3.1 Algorithms for Guaranteed Convergence

Theoretic work on convergence in evolutionary multiobjective optimization is mainly due to Rudolph (1998a, 1998b, 2001), Rudolph and Agapie (2000), and Hanne (1999, 2001). The corresponding concepts and their algorithms are described in the following.

Efficiency Preservation and the Problem of Deterioration Hanne (1999) suggested and implemented (2001) a selection strategy for MOEAs based on the concept of "(negative) efficiency preservation" as a multiobjective generalization of the "plus" (elitist) selection in evolution strategies. He defines efficiency preservation as the property of only accepting new solutions that dominate at least one of the current solutions. Negative efficiency preservation is given when a solution is discarded only if a dominating solution is accepted in return. Both properties are mutually independent, and sufficient to preclude the problem of *deterioration*. Deterioration occurs, when elements of a solution set at a given time are dominated by a solution set the algorithm maintained some time before. This can happen using the standard Pareto-based selection schemes even under elitism, as well as with virtually all archiving schemes used in the advanced state-of-the-art MOEAs, as will be described shortly.

In Hanne (1999), a convergence proof for a $(\mu + \lambda)$-MOEA with Gaussian mutation distributions over a compact real search space has been enabled by the application of a (negative) efficiency preservation selection scheme. A disadvantage of this approach is that no assumptions can be given as to the distribution of solutions, since with both efficiency and negative efficiency preservation, arbitrary regions of the objective space, and hence of the Pareto set, can become unreachable.

Rudolph's and Agapie's Elitist MOEAs Based on Rudolph (1998a), Rudolph and Agapie (2000) suggested MOEAs with a fixed-size archive, where a sophisticated selection process precludes the problem of deterioration. They have shown that these algorithms with variation operators having a positive transition probability matrix guarantee convergence to the Pareto-optimal set. However, when all archive members are Pareto-optimal, the algorithm does not allow any new Pareto-optimal solution to enter the archive. Thus, although the algorithms guarantee convergence to the true Pareto-optimal front, they do not guarantee a good distribution of Pareto-optimal solutions.

3.2 Elitist MOEAs with Focus on the Distribution Quality

Recently a number of elitist MOEAs have been proposed that especially address the diversity of the archived solutions by different mechanisms.

Pareto-Archived Evolution Strategy (PAES) Knowles and Corne (2000) suggested a simple elitist MOEA using a single parent, single child $(1+1)$-evolutionary algorithm called PAES. If a new solution is not dominated by any archive member, it is included in the archive, deleting in turn all members that it dominates. If the archive would exceed its maximum size, the acceptance of new solutions is decided by a histogram-like density measure over a hyper-grid division of the objective space. This archiving strategy is similar to the one proposed by Kursawe (1990, 1991), who already used an adaptive distance measure to maintain a good spread of nondominated solutions in a fixed-size archive.

Strength Pareto Evolutionary Algorithm (SPEA) Zitzler and Thiele (1999) have suggested an elitist MOEA using the concept of nondomination and a secondary population of nondominated points. After every generation, the secondary population is updated with the nondominated offspring, while all dominated elements are discarded. If this archive exceeds its maximum size, a clustering mechanism groups all currently nondominated solutions into a predefined number of clusters and picks a representative solution from each cluster, thereby ensuring diversity among the external population members.

Elitist Non-Dominated Sorting GA (NSGA-II) In NSGA-II (Deb et al., 2000), the parent and offspring population (each of size N) are combined and evaluated using (i) a fast nondominated sorting approach, (ii) an elitist approach, and (iii) an efficient crowding approach. When more than N population members of the combined population belong to the nondominated set, only those that are maximally apart from their neighbors according to the crowding measure are chosen.

This way, like PAES and SPEA, an existing nondominated solution may get replaced by another, since selection is then based only on the specific diversity or density measure or on the clustering procedure. In a succession of these steps, deterioration possibly occurs, thus convergence can no longer be guaranteed for any of these algorithms.

3.3 Limitations

It is clear from the above discussion that the above elitist MOEAs cannot achieve both tasks simultaneously, either they enable convergence or they focus on a good distribution of solutions. The convergence criterion can easily be fulfilled by dominance preservation, however, a pure implementation of this approach leaves the distribution aspect unsolved. All algorithms focusing on a good distribution are in danger of deterioration though. The diversity-preservation operator used in each of the above algorithms is primarily geared to maintain spread among solutions. While doing so, the algorithm has no way of *knowing* which solutions are already Pareto-optimal and which are not. The diversity-preservation operator always emphasizes the less crowded regions of the nondominated solutions.

4 Algorithms for Convergence and Diversity

Before we present the update functions for finding a diverse set of Pareto-optimal solutions, we define some terminology.

4.1 Concept of Pareto Set Approximation

In this section, we define relevant concepts of dominance and (approximate) Pareto sets. Without loss of generality, we assume a normalized and positive objective space in the following for notational convenience. The algorithms presented in this paper assume that all objectives are to be maximized. However, either by using the *duality* principle (Deb, 2001) or by simple modifications to the domination definitions, these algorithms can be used to handle minimization or combined minimization and maximization problems.

Objective vectors are compared according to the dominance relation defined below and displayed in Figure 2 (left).

DEFINITION 1 (Dominance Relation): *Let $f, g \in \mathbb{R}^m$. Then f is said to dominate g (denoted as $f \succ g$) iff*

1. $\forall i \in \{1, \ldots, m\} : f_i \geq g_i$
2. $\exists j \in \{1, \ldots, m\} : f_j > g_j$

Based on the concept of dominance, the Pareto set can be defined as follows.

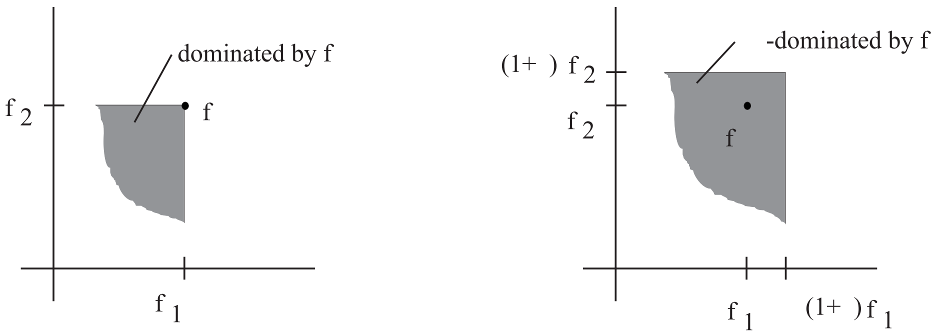

Figure 2: Graphs visualizing the concepts of dominance (left) and ϵ-dominance (right).

DEFINITION 2 (Pareto Set): *Let $F \subseteq \mathbb{R}^m$ be a set of vectors. Then the Pareto set F^* of F is defined as follows: F^* contains all vectors $g \in F$ that are not dominated by any vector $f \in F$, i.e.,*

$$F^* := \{g \in F | \nexists f \in F : f \succ g\} \tag{1}$$

Vectors in F^* are called Pareto vectors of F. The set of all Pareto sets of F is denoted as $P^*(F)$.

From the above definition we can easily deduce that any vector $g \in F \backslash F^*$ is dominated by at least one $f \in F^*$, i.e.,

$$\forall g \in F \setminus F^* : \quad \exists f \in F^* \quad \text{such that} \quad f \succ g. \tag{2}$$

Moreover, for a given set F, the set F^* is unique. Therefore, we have $P^*(F) = \{F^*\}$. For many sets F, the Pareto set F^* is of substantial size. Thus, the numerical determination of F^* is prohibitive, and F^* as a result of an optimization is questionable. Moreover, it is not clear at all what a decision maker can do with such a large result of an optimization run. What would be more desirable is an approximation of F^* that *approximately* dominates all elements of F and is of (polynomially) bounded size. This set can then be used by a decision maker to determine interesting regions of the decision and objective space, which can be explored in further optimization runs. Next, we define a generalization of the dominance relation as visualized in Figure 2 (right).

DEFINITION 3 (ϵ-Dominance): *Let $f, g \in \mathbb{R}^{+m}$. Then f is said to ϵ-dominate g for some $\epsilon > 0$, denoted as $f \succ_\epsilon g$, iff for all $i \in \{1, \ldots, m\}$*

$$(1 + \epsilon) \cdot f_i \geq g_i. \tag{3}$$

DEFINITION 4 (ϵ-approximate Pareto Set): *Let $F \subseteq \mathbb{R}^{+m}$ be a set of vectors and $\epsilon > 0$. Then a set F_ϵ is called an ϵ-approximate Pareto set of F, if any vector $g \in F$ is ϵ-dominated by at least one vector $f \in F_\epsilon$, i.e.,*

$$\forall g \in F : \exists f \in F_\epsilon \quad \textit{such that} \quad f \succ_\epsilon g. \tag{4}$$

The set of all ϵ-approximate Pareto sets of F is denoted as $P_\epsilon(F)$.

Of course, the set F_ϵ is not unique. Many different concepts for ϵ-efficiency[1] and the corresponding Pareto set approximations exist in the operations research literature, a survey is given by Helbig and Pateva (1994). As most of the concepts deal with infinite sets, they are not practical for our purpose of producing and maintaining a representative subset. Nevertheless they are of theoretical interest and have nice properties which can for instance be used in convergence proofs (see Hanne (1999) for an application in MOEAs).

Using discrete ϵ-approximations of the Pareto set was suggested simultaneously by Evtushenko and Potapov (1987), Reuter (1990), and Ruhe and Fruhwirt (1990). As in our approach, each Pareto-optimal point is approximately dominated by some point of the representative set. The first two papers use absolute deviation (additive ϵ, see below) and the third relative deviation (multiplicative ϵ as above), but they are not concerned with the size of the representative set in the general case.

Recently, Papadimitriou and Yannakakis (2000) and Erlebach et al. (2001) pointed out that under certain assumptions, there is always an approximate Pareto set whose size is polynomial in the length of the encoded input. This can be achieved by placing a hyper-grid in the objective space using the coordinates $1, (1+\epsilon), (1+\epsilon)^2, \ldots$ for each objective. As it suffices to have one representative solution in each grid cell and to have only nondominated cells occupied, it can be seen that for any finite ϵ and any set F with bounded vectors f, i.e., $1 \leq f_i \leq K$ for all $i \in \{1, \ldots, m\}$, there exists a set F_ϵ containing

$$|F_\epsilon| \leq \left(\frac{\log K}{\log(1+\epsilon)}\right)^{m-1} \tag{5}$$

vectors. A proof will be given in connection with Algorithm 3 in Section 4.3.

Note that the concept of approximation can also be used if other similar definitions of ϵ-dominance are used, e.g., the following additive approximation

$$\epsilon_i + f_i \geq g_i \quad \forall i \in \{1, \ldots, m\} \tag{6}$$

where ϵ_i are constants, separately defined for each coordinate. In this case there exist ϵ-approximate Pareto sets whose size can be bounded as follows:

$$|F_\epsilon| \leq \prod_{j=1}^{m-1} \frac{K-1}{\epsilon_i} \tag{7}$$

where $1 \leq f_i \leq K$, $K \geq \epsilon_i$ for all $i \in \{1, \ldots, m\}$. A further refinement of the concept of ϵ-approximate Pareto sets leads to the following definition.

DEFINITION 5 (ϵ-Pareto Set): *Let $F \subseteq \mathbb{R}^{+m}$ be a set of vectors and $\epsilon > 0$. Then a set $F_\epsilon^* \subseteq F$ is called an ϵ-Pareto set of F if*

1. *F_ϵ^* is an ϵ-approximate Pareto set of F, i.e., $F_\epsilon^* \in P_\epsilon(F)$, and*
2. *F_ϵ^* contains Pareto points of F only, i.e., $F_\epsilon^* \subseteq F^*$.*

The set of all ϵ-Pareto sets of F is denoted as $P_\epsilon^*(F)$.

[1] The terms "efficient" and "Pareto-optimal" can be used synonymously. While the former appears to be more frequent in operations research literature, we generally use the latter as it is more common in the field of evolutionary computation.

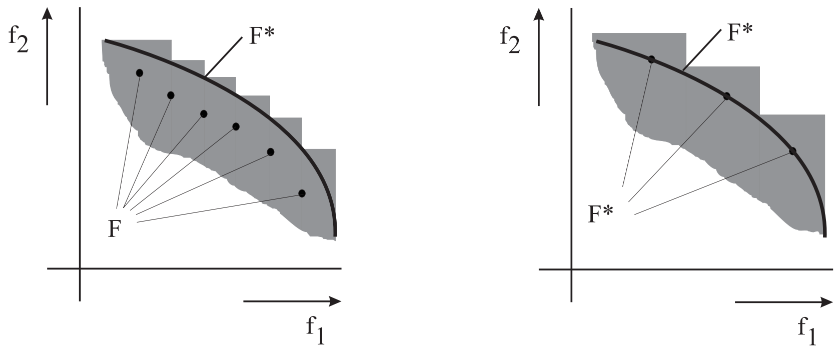

Figure 3: Graphs visualizing the concepts of ϵ-approximate Pareto set (left) and ϵ-Pareto set (right).

The above defined concepts are visualized in Figure 3. An ϵ-Pareto set F_ϵ^* not only ϵ-dominates all vectors in F, but also consists of Pareto-optimal vectors of F only, therefore we have $P_\epsilon^*(F) \subseteq P_\epsilon(F)$.

Since finding the Pareto set of an arbitrary set F is usually not practical because of its size, one needs to be less ambitious in general. Therefore, the ϵ-approximate Pareto set is a practical solution concept as it not only represents all vectors F but also consists of a smaller number of elements. Of course, a ϵ-Pareto set is more attractive as it consists of Pareto vectors only.

Convergence and diversity can be defined in various ways. Here, we consider the objective space only. According to Definition 3, the ϵ value stands for a relative "tolerance" that we allow for the objective values. In contrast, using Equation (6) we would allow a constant additive (absolute) tolerance.

The choice of the ϵ value is application specific: A decision maker should choose a type and magnitude that suits the (physical) meaning of the objective values best. The ϵ value further determines the maximal size of the archive according to Equations (5) and (7).

4.2 Maintaining an ϵ-Approximate Pareto Set

We first present an *update* function that leads to the maintenance of an ϵ-approximate Pareto set. The idea is that new points are only accepted if they are not ϵ-dominated by any other point of the current archive. If a point is accepted, all dominated points are removed.

THEOREM 1: *Let $F^{(t)} = \bigcup_{j=1}^{t} f^{(j)}, 1 \leq f_i^{(j)} \leq K$, be the set of all vectors created in Algorithm 1 and given to the update function as defined in Algorithm 2. Then $A^{(t)}$ is an ϵ-approximate Pareto set of $F^{(t)}$ with bounded size, i.e.,*

1. $A^{(t)} \in P_\epsilon(F^{(t)})$
2. $|A^{(t)}| \leq \left(\frac{\log K}{\log(1+\epsilon)}\right)^m$

Algorithm 2 *update* function for ϵ-approximate Pareto set

1: **Input:** A, f
2: **if** $\exists f' \in A$ such that $f' \succ_\epsilon f$ **then**
3: $\quad A' := A$
4: **else**
5: $\quad D := \{f' \in A | f \succ f'\}$
6: $\quad A' := A \cup \{f\} \setminus D$
7: **end if**
8: **Output:** A'

PROOF:

1. Suppose the algorithm is not correct, i.e., $A^{(t)} \not\in P_\epsilon(F^{(t)})$ for some t. According to Definition 4 this occurs only if some $f = f^{(\tau)}, \tau \leq t$ is not ϵ-dominated by any member of $A^{(t)}$ and is not in $A^{(t)}$.

   For $f = f^{(\tau)}$ not being in $A^{(t)}$, it can either have been rejected at $t = \tau$ or accepted at $t = \tau$ and removed later on. Removal, however, only takes place when some new f' enters A which dominates f (line 6). Since the dominance relation is transitive, and since it implies ϵ-dominance, there will always be an element in A that ϵ-dominates f, which contradicts the assumption. On the other hand, f will only be rejected if there is another $f' \in A^{(\tau)}$ that ϵ-dominates f (line 2) and – with the same argument as before – can only be replaced by accepting elements that also ϵ-dominate f.

2. Every $f \in A^{(t)}$ defines a hyper-rectangle between f and $(1 + \epsilon) \cdot f$, where no other element of $A^{(t)}$ can exist because dominated elements are always deleted from the set. Furthermore, these areas do not overlap since this would mean that the two corresponding points ϵ-dominate each other, which is precluded by the acceptance criterion. The maximum number of non-overlapping hyper-rectangles in the whole objective space is given by $\left(\frac{\log K}{\log(1+\epsilon)}\right)^m$. □

The algorithms VV and PR of Rudolph and Agapie (2000) can be viewed as special cases of this algorithm for $\epsilon \to 0$. In the limit, the ϵ-dominance becomes the normal dominance relation, and the algorithm will always maintain a set of only nondominated vectors. Of course, according to the previous theorem, the size of this set might grow to infinity as $t \to \infty$.

4.3 Maintaining an ϵ-Pareto Set

In a next step, we would like to guarantee – in addition to a minimal distance between points – that the points in $A^{(t)}$ are Pareto points of all vectors generated so far. The following Algorithm 3 has a two level concept. On the coarse level, the search space is discretized by a division into boxes (see Algorithm 4), where each vector uniquely belongs to one box. Using a generalized dominance relation on these boxes, the algorithm always maintains a set of nondominated boxes, thus guaranteeing the ϵ-approximation property. On the fine level, at most one element is kept in each box. Within a box, each representative vector can only be replaced by a dominating one (similar to Agapie and Rudolph's algorithm), thus guaranteeing convergence.

Now, we can prove the convergence of the above update strategy to the Pareto set while preserving diversity of solution vectors at the same time.

Algorithm 3 *update* function for ϵ-Pareto set

1: **Input:** A, f
2: $D := \{f' \in A | box(f) \succ box(f')\}$
3: **if** $D \neq \emptyset$ **then**
4:     $A' := A \cup \{f\} \setminus D$
5: **else if** $\exists f' : (box(f') = box(f) \wedge f \succ f')$ **then**
6:     $A' := A \cup \{f\} \setminus \{f'\}$
7: **else if** $\nexists f' : box(f') = box(f) \vee box(f') \succ box(f)$ **then**
8:     $A' := A \cup \{f\}$
9: **else**
10:     $A' := A$
11: **end if**
12: **Output:** A'

Algorithm 4 function *box*

Input: f
for all $i \in \{1, \ldots, m\}$ **do**
    $b_i := \lfloor \frac{\log f_i}{\log (1+\epsilon)} \rfloor$
end for
$b := (b_1, \ldots, b_m)$
Output: b {box index vector}

THEOREM 2: *Let $F^{(t)} = \bigcup_{j=1}^{t} f^{(j)}, 1 \leq f_i^{(j)} \leq K$ be the set of all vectors created in Algorithm 1 and given to the update function as defined in Algorithm 3. Then $A^{(t)}$ is an ϵ-Pareto set of $F^{(t)}$ with bounded size according to Equation (5), i.e.,*

1. $A^{(t)} \in P_\epsilon^*(F^{(t)})$
2. $|A^{(t)}| \leq \left(\frac{\log K}{\log (1+\epsilon)}\right)^{(m-1)}$

PROOF:

1. Suppose the algorithm is not correct, i.e., $A^{(t)} \notin P_\epsilon^*(F^{(t)})$ for some t. According to Definition 5, this occurs only if some $f = f^{(\tau)}, \tau \leq t$ is (Case 1) not ϵ-dominated by any member of $A^{(t)}$ and not in $A^{(t)}$ or (Case 2) in $A^{(t)}$ but not in the Pareto set of $F^{(t)}$.

   Case (1): For $f = f^{(\tau)}$ not being in $A^{(t)}$, it can either have been rejected at $t = \tau$ or accepted at $t = \tau$ and removed later on. Removal, however, only takes place when some new f' enters A, which dominates f (line 6) or whose box value dominates that of f (line 4). Since both relations are transitive, and since they both imply ϵ-dominance, there will always be an element in A that ϵ-dominates f, which contradicts the assumption. On the other hand, f will only be rejected if there is another $f' \in A^{(\tau)}$ with the same box value and that is not dominated by f (line 10). This f', in turn, ϵ-dominates f and, with the same argument as before, can only be replaced by accepting elements that also ϵ-dominate f.

   Case (2): Since f is not in the Pareto set of $F^{(t)}$, there exists $f' = f^{(\tau')}, \tau' \neq \tau, f' \in F^{*(t)}$ with $f' \succ f$. This implies $box(f') \succ box(f)$ or $box(f') = box(f)$. Hence, if

$\tau' < \tau$, f would not have been accepted. If $\tau' > \tau$, f would have been removed from A. Thus, $f \notin A^{(t)}$, which contradicts the assumption.

2. The objective space is divided into $\left(\frac{\log K}{\log(1+\epsilon)}\right)^m$ boxes, and from each box at most one point can be in $A^{(t)}$ at the same time. Now consider the $\left(\frac{\log K}{\log(1+\epsilon)}\right)^{(m-1)}$ equivalence classes of boxes where, without loss of generality, in each class the boxes have the same coordinates in all but one dimension. There are $\frac{\log K}{\log(1+\epsilon)}$ different boxes in each class constituting a chain of dominating boxes. Hence, only one point from each of these classes can be a member of $A^{(t)}$ at the same time. □

As a result, Algorithms 2 and 3 use finite memory and successively update a finite subset of vectors that ϵ-dominate all vectors generated so far. For Algorithm 3, it can additionally be guaranteed that the subset contains only elements which are not dominated by any of the generated vectors. Note that specific bounds on the objective values are *not* used in the algorithms themselves and are not required to prove the convergence. They are only utilized to prove the relation between ϵ and the size of the archive given in the second claim.

5 Simulations

This section presents some simulation results to demonstrate the behavior of the proposed algorithms for two example *multiobjective optimization problems* (MOPs). We use instances of the iterative search procedure (specified in Algorithm 1) with a common generator and examine different *update* operators. An isolated assessment of the update strategy of course requires the generator to act independently from the archive set $A^{(t)}$ to guarantee that exactly the same sequence of points is given to the update function for all different strategies. Despite that, the exact implementation of the generator is irrelevant for this study, therefore we use standard MOEAs here and take the points in the sequence of their generation as input for the different update functions.

5.1 Convergence Behavior

At first we are interested in how different update strategies affect the *convergence* of the sequence $(A^{(t)})$. As a test problem, a two-objective knapsack problem with 100 items is taken from Zitzler and Thiele (1999). The low number of decision variables is sufficient to show the anticipated effects, and we found it advantageous for visualization and comparison purposes to be able to compute the complete Pareto set F^* beforehand via Integer Linear Programming.

The points given to the update operator are generated by a standard NSGA-II with population size 100, one-point crossover, and bit-flip mutations (with probability $4/n = 0.04$). Figure 4 shows the output $A^{(t)}$ of sample runs for the different instances after $t = 5,000,000$ and $t = 10,000,000$ iterations (generated objective vectors), using update operators from SPEA, NSGA-II (both with maximum archive size of 20), and Algorithm 3 with $\epsilon = 0.01$.

It is clearly visible that both the archiving (selection) strategies from SPEA and NSGA-II suffer from the problem of partial deterioration: Nondominated points – even those belonging to the "real" Pareto set – can get lost, and in the long run might even be replaced by dominated solutions. This is certainly not desirable, and algorithms relying on these strategies cannot claim to be convergent, even if the generator produces all

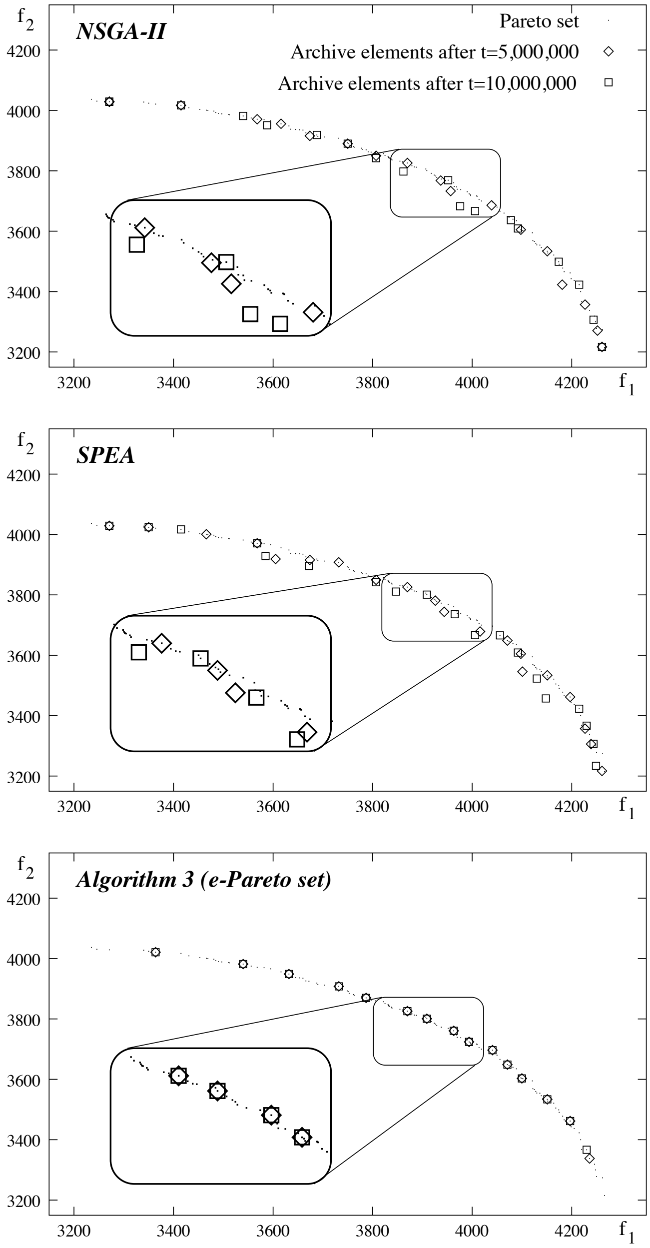

Figure 4: Objective space of the knapsack problem. The dots show the elements of the Pareto set F^*. The different figures correspond to different instances of the *update* operator in Algorithm 1: NSGA-II (top), SPEA (middle), and Algorithm 3 (bottom). In each figure, the archive set $A^{(t)}$ is shown for $t = 5,000,000$ (with diamonds) and for $t = 10,000,000$ (with boxes). A subset of the samples is enlarged to highlight the negative effect of losing Pareto-optimal solutions in many current archiving/selection schemes.

elements of the Pareto set F^*. In contrast, Algorithm 3 is able to maintain an ϵ-Pareto set of the generated solutions over time.

The number of function evaluations in this experiment is certainly extremely high, but necessary to produce all Pareto-optimal points in this test case, especially at the extremes of the Pareto set. It shows that the problem of deterioration does not only occur at the beginning of a run. The nonconvergent algorithms can even be run infinitely long without converging the Pareto set, although all Pareto-optimal points are generated over and over again.

5.2 Distribution Behavior

In order to test for the *distribution* behavior, only candidates are taken into account that fulfill the requirements for convergence: Rudolph and Agapie's algorithm AR-1 and Algorithm 3. As a test case, the following continuous, three-dimensional, three-objective problem is used:

$$\left.\begin{array}{ll}\text{Maximize} & f_1(x) = 3 - (1 + x_3)\cos(x_1\pi/2)\cos(x_2\pi/2),\\ \text{Maximize} & f_2(x) = 3 - (1 + x_3)\cos(x_1\pi/2)\sin(x_2\pi/2),\\ \text{Maximize} & f_3(x) = 3 - (1 + x_3)\cos(x_1\pi/2)\sin(x_1\pi/2),\\ & 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, 3,\end{array}\right\} \tag{8}$$

The Pareto set of this problem is a surface, a quadrant of the hyper-sphere of radius 1 around $(3, 3, 3)$. For the results shown in Figure 5, the real-coded NSGA without fitness sharing, crossover using *Simulated Binary Crossover* (SBX) (Deb and Agrawal, 1995), with distribution index $\eta = 5$, and population size 100 was used to generate the candidate solutions. The distribution quality is judged in terms of the ϵ-dominance concept, therefore a discretization of the objective space into boxes (using Algorithm 4 with $\epsilon = 0.05$) is plotted instead of the actual Pareto set. From all boxes intersecting with the Pareto set, the nondominated ones are highlighted. For an ϵ-approximate Pareto set it is now sufficient to have exactly one solution in each of those nondominated boxes. This condition is fulfilled by the algorithm using the update strategy Algorithm 3, leading to an almost symmetric distribution covering all regions. The strategy from AR-1, which does not discriminate among nondominated points, is sensitive to the sequence of the generated solution and fails to provide an ϵ-approximation of the Pareto set of similar quality even with an allowed archive size of 50.

Looking at the graphs of Algorithm 3, one might have the impression that not all regions of the Pareto set are equally represented by archive members. However, these examples represent *optimal* approximations according to the concepts explained in Section 4.1. They are not intended to give a uniform distribution on a (hypothetical) surface that might even not exist as in the discrete case.

5.3 Results

The simulation results support the claims of the preceding sections. The archive updating strategy plays a crucial role for the convergence and distribution properties. The key results are:

- Rudolph and Agapie's algorithm guarantees convergence, but has no control over the distribution of points.
- The current MOEAs designed for maintaining a good distribution do not fulfill the convergence criterion, as has been demonstrated for SPEA and NSGA-II for a simple test case.

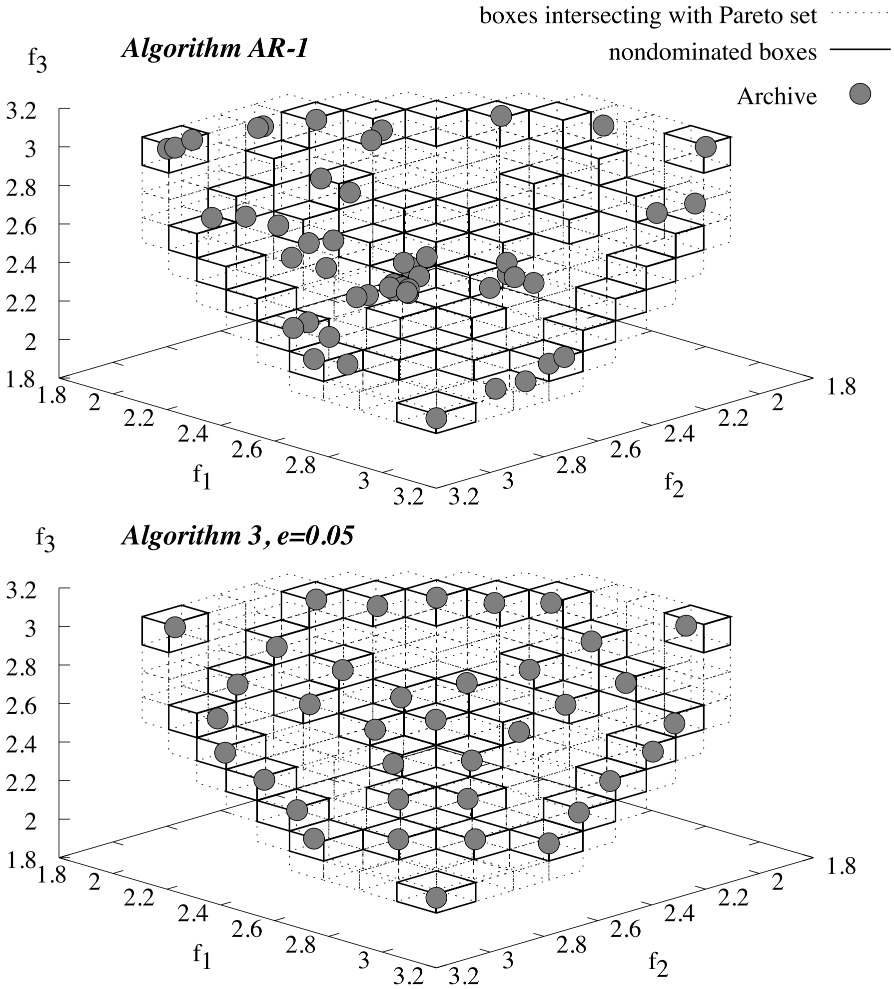

Figure 5: Objective space of MOP (8). The discretization into boxes according to Algorithm 4 is indicated by showing all boxes that intersect with the Pareto set F^* in dashed lines. The nondominated boxes are drawn in bold lines. The circles correspond to the output A of different instances of the iterative search algorithm Algorithm 1. For the upper figure, an update function according to AR-1 was used, for the lower figure, the function according to Algorithm 3.

- The algorithms proposed in this paper fulfill both the convergence criterion and the desired distribution control.

6 Possible Extensions

The above baseline algorithms can be extended in several interesting and useful ways. In the following, we list some of these extensions and variations and discuss them briefly.

6.1 Other Definitions of ϵ-Dominance

The convergent algorithms can also be implemented with a different definition of ϵ-dominance. For example, with the dominance definition given in (6), grids are uniformly sized in the search space. Although the size of the generated Pareto-optimal set will be different from that presented earlier, the algorithms given so far also maintain the convergence and preserve the diversity.

Although an identical ϵ value is suggested in the definition of ϵ-dominance, a different ϵ_i can be used for each coordinate of the objective space. This way, different precisions among the obtained Pareto-optimal vectors can be obtained in different criteria. The upper bound of the number of Pareto-optimal solutions presented above will get modified accordingly.

6.2 Guaranteeing a Minimum Distance Between Obtained Vectors

The ϵ-dominance definition and the diversity preservation through grids allow a diverse and well-convergent set of Pareto-optimal vectors to be obtained by the proposed algorithms. Although diversity among the elements is ensured, the distance between the obtained neighboring Pareto-optimal vectors may not be uniform. It is guaranteed by the proposed algorithms that one box will have only one solution. But in practice, two vectors lying on two neighboring boxes may lie very close to each other. To ensure a good diversity among neighboring vectors, Algorithm 3 may be modified in the following manner. In addition to discouraging two vectors to lie on the same box, the vectors can also be discouraged to lie on the *even* numbered boxes. This way, vectors can only lie on the alternate boxes, thereby ensuring a minimum difference of ϵ in each objective function value between two neighboring Pareto-optimal vectors.

6.3 Steering Search by Defining Ranges of Non-Acceptance

In most multiobjective optimization problems, a decision maker plays an important role. If the complete search space is not of importance to a decision maker, the above algorithms can be used to search along preferred regions. The concept of ϵ-dominance will then allow prespecified precisions to exist among the obtained preferred Pareto-optimal vectors.

6.4 Fixed Archive Size by Dynamic Adaptation of ϵ

Instead of predetermining an approximation accuracy ϵ in advance, one might ask whether the algorithm would be able to dynamically adjust its accuracy to always maintain a set of vectors of a given magnitude. A concept like this is implemented in PAES (see Section 3), where the hyper-grid dividing the objective space is adapted to the current ranges given by the nondominated vectors. However, PAES does not guarantee convergence.

Here two modified versions of the proposed converging algorithms are illustrated. The idea is to start with a minimal ϵ, which is systematically increased every time the number of archived vectors exceeds a predetermined maximum size.

6.4.1 Maintaining an ϵ-Approximate Pareto Set

In order to generate an ϵ-approximate Pareto set with a given upper bound a on its size, Algorithm 2 can be modified. After the first pair $g^{(1)}, g^{(2)}$ of mutually nondominating vectors have been found, an initial ϵ is calculated according to Theorem 1 as

$$\epsilon = K'^{\frac{1}{a^{1/m}}} - 1 \tag{9}$$

where a is the maximum archive size. K' is set to the current maximum relative range of the m objectives

$$K' := \max_{1 \leq i \leq m} \left\{ \frac{u_i}{l_i} \right\} \tag{10}$$

where u_i and l_i, $u_i \geq l_i > 0$ are the current maximum and minimum values of objective function i.

From this onwards, new vectors are accepted according to Algorithm 2, where for each element, a time stamp is recorded. If the archive would exceed its maximum size a_{max}, a larger ϵ must be chosen, again using the new ranges and the above formulas. By this new ϵ, all archive elements are again compared in the order of their time stamps. Whenever one element is ϵ-dominated by an older one, the younger will be removed. This way, the ranges will always be increased in order to cover the whole extent of the current ϵ-approximate Pareto set. However, if the range of the actual Pareto set decreases in the later part of the run, there is no possibility to decrease the ϵ again without, in general, losing the property given by Theorem 1.

For the ϵ-dominance definition given in Equation (6), Equation (9) becomes $\epsilon = \frac{K'}{a^{1/m}} - 1$, and K' is calculated as $K' = \max_{1 \leq i \leq m}\{u_i - l_i\}$.

Agapie's and Rudolph's algorithms AR-1 and AR-2 also implement a fixed-size archive, but with a constant $\epsilon = 0$ during the run. This means that the guaranteed minimal distance of vectors is also zero, hence not guaranteeing to maintain an ϵ-approximate Pareto set.

6.4.2 Maintaining an ϵ-Pareto Set

In Algorithm 3, a simple modification would be to start with a minimal ϵ using a first pair of mutually nondominated vectors as in the previous subsection. Afterwards, the increase of ϵ is taken care of by joining boxes and discarding all but the oldest element from the new, bigger box.

The joining of boxes could be done in several ways, however, for ensuring the convergence property, it is important not to move or translate any of the box limits, in other words, the assignment of the elements to the boxes must stay the same. A simple implementation could join every second box, while it suffices to join only in the dimensions where the ranges have been exceeded by the new element. This will mean that in the worst case an area will be ϵ-dominated that is almost twice the size of the actual Pareto set in each dimension. A more sophisticated approach would join only two boxes at a time, which would eliminate the over-covering, but make a complicated bookkeeping of several different ϵ values in each dimension necessary.

6.4.3 A Bistart Strategy

In cases where the bounds of the Pareto set are much smaller than the bounds on F, both algorithms suffer from their inability to increase the precision again after having reached any level of coarseness. In the worst case, they might end up with only one solution ϵ-dominating the whole Pareto set using a rather large ϵ.

We illustrate how to use our proposed algorithms to maintain an ϵ-approximate Pareto set or an ϵ-Pareto set, respectively, with a maximum predefined cardinality a_{max}. For this, a two-step strategy is followed: First, one of the two dynamic algorithms of the previous section is used to get a first, rough approximation of the Pareto set. From their results the ranges of the Pareto set in the objective space can be determined and used to calculate a fixed ϵ for a second run of the algorithm. Of course, one may use different ϵ_i for the different objectives. In the second run, the only required change to the *update* operator is that it never accepts any vectors outside the ranges determined by the first run, hence ensuring that the size of the solution set does not exceed the limit a_{max}.

7 Conclusions

In this study, we proposed the ϵ-(approximate) Pareto set as a solution concept for evolutionary multiobjective optimization that is

- theoretically attractive as it helps to construct algorithms with the desired convergence and distribution properties, and
- practically important as it works with a solution set with bounded size and predefined resolution.

We constructed the first archive updating strategies that

- can be used in any iterative search process and
- allow for the desired convergence properties, while at the same time,
- guaranteeing an optimal distribution of solutions.

As we have exclusively dealt with the update operator (or the archiving/selection scheme of the corresponding search and optimization algorithms) so far, all statements had to be done with respect to the generated solutions only. In order to make a statement about the convergence to the Pareto set of the whole search space, one has of course to include the generator into the analysis. However, with appropriate assumptions (nonvanishing probability measure for the generation of all search points at any time step), it is clear that the probability of not creating a specific point goes to zero as t goes to infinity. Analogously to Hanne (1999) or Rudolph and Agapie (2000), results on the limit behavior, such as almost sure convergence and stochastic convergence to an ϵ-Pareto set (including features described in this paper), can be derived.

Though the limit behavior might be of mainly theoretical interest, it is of high practical relevance that now the problem of partial deterioration, which was imminent even in the elitist MOEAs, could be solved. Using the proposed archiving strategy to maintain an ϵ-Pareto, set the user can be sure to have, in addition to a representative, well-distributed approximation, a true elitist algorithm in the sense that no better solution had been found and subsequently lost during the run.

Interesting behavior occurs when there are interactions between the archive and the generator. Allowing the archive members to take part in the generating process has empirically been investigated (e.g., Laumanns et al. (2000, 2001)) using a more general model and a parameter called *elitism intensity*. Now, the theoretical foundation is also given so that the archived members are really guaranteed to be the best solutions found.

Acknowledgments

The work has been supported by the Swiss National Science Foundation (SNF) under the ArOMA project 2100-057156.99/1.

References

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK.

Deb, K. and Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148.

Deb, K. et al. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Schoenauer, M. et al., editors, *Parallel Problem Solving from Nature – PPSN VI*, pages 849–858, Springer, Berlin, Germany.

Erlebach, T., Kellerer, H., and Pferschy, U. (2001). Approximating multi-objective knapsack problems. In *Proceedings of the Seventh International Workshop on Algorithms and Data Structures (WADS 2001). Lecture Notes in Computer Science*, 2125:210–221.

Evtushenko, Y. G. and Potapov, M. A. (1987). Methods of numerical solution of multicriterion problem. *Soviet mathematics – doklady*, 34:420–423.

Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, Morgan Kaufmann, San Mateo, California.

Goldberg, D. E. (1989). *Genetic Algorithms for Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts.

Hanne, T. (1999). On the convergence of multiobjective evolutionary algorithms. *European Journal Of Operational Research*, 117(3):553–564.

Hanne, T. (2001). Global multiobjective optimization with evolutionary algorithms: Selection mechanisms and mutation control. In Zitzler, E. et al., editors, *Evolutionary Multi-Criterion Optimization (EMO 2001) Proceedings*. *Lecture Notes in Computer Science*, 1993:197–212.

Helbig, S. and Pateva, D. (1994). On several concepts for ϵ-efficiency. *OR Spektrum*, 16(3):179–186.

Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, Volume 1, pages 82–87, IEEE Press, Piscataway, New Jersey.

Knowles, J. D. (2002). *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. Ph.D. thesis, The University of Reading, Department of Computer Science, Reading, UK.

Knowles, J. D. and Corne, D. W. (2000). Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172.

Kursawe, F. (1990). Evolutionsstrategien für die Vektoroptimierung. Diplomarbeit, Universität Dortmund, Dortmund, Germany (in German).

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In Schwefel, H.-P. and Männer, R., editors, *Parallel Problem Solving from Nature*, pages 193–197, Springer, Berlin, Germany.

Laumanns, M., Zitzler, E., and Thiele, L. (2000). A unified model for multi-objective evolutionary algorithms with elitism. In *Congress on Evolutionary Computation (CEC 2000)*, Volume 1, pages 46–53, IEEE Press, Piscataway, New Jersey.

Laumanns, M., Zitzler, E., and Thiele, L. (2001). On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In Zitzler, E. et al., editors, *Evolutionary Multi-Criterion Optimization (EMO 2001) Proceedings*. *Lecture Notes in Computer Science*, 1993:181–196.

Laumanns, M. et al. (2002). Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving From Nature — PPSN VII*, Springer, Berlin, Germany.

Papadimitriou, C. H. and Yannakakis, M. (2000). The complexity of tradeoffs, and optimal access of web sources. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science – FOCS 2000*, pages 86–92, IEEE Computer Society, Los Alamitos, California.

Reuter, H. (1990). An approximation method for the efficiency set of multiobjective programming problems. *Optimization*, 21:905–911.

Rudolph, G. (1998a). Evolutionary search for minimal elements in partially ordered sets. In Porto, V. W. et al., editors, *Evolutionary Programming VII – Proceedings of the Seventh Annual Conference on Evolutionary Programming (EP-98)*, pages 345–353, Springer, Berlin, Germany.

Rudolph, G. (1998b). On a multi-objective evolutionary algorithm and its convergence to the pareto set. In *IEEE International Conference on Evolutionary Computation (ICEC'98)*, pages 511–516, IEEE Press, Piscataway, New Jersey.

Rudolph, G. (2001). Evolutionary Search under Partially Ordered Fitness Sets. In Sebaaly, M. F., editor, *Proceedings of the International NAISO Congress on Information Science Innovations (ISI 2001)*, pages 818–822, ICSC Academic Press, Sliedrecht, The Netherlands.

Rudolph, G. and Agapie, A. (2000). Convergence properties of some multi-objective evolutionary algorithms. In Zalzala, A. and Eberhart, R., editors, *Congress on Evolutionary Computation (CEC 2000)*, Volume 2, pages 1010-1016, IEEE Press, Piscataway, New Jersey.

Ruhe, G. and Fruhwirt, B. (1990). ϵ-optimality for bicriteria programs and its application to minimum cost flows. *Computing*, 44:21–34.

Schaffer, J. D. (1984). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Ph.D. thesis, Vanderbilt University, Nashville, Tennessee.

Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Zitzler, E. et al., editors, (2001). *Evolutionary Multi-Criterion Optimization (Lecture Notes in Computer Science 1993)*. Springer, Heidelberg, Germany.