Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization

Marco Laumanns, Lothar Thiele, Eckart Zitzler
Computer Engineering and Networks Laboratory
Swiss Federal Institute of Technology (ETH) Zurich
CH-8092 Zurich, Switzerland
{laumanns,thiele,zitzler}@tik.ee.ethz.ch

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

Abstract

Over the past few years, the research on evolutionary algorithms has demonstrated their niche in solving multi-objective optimization problems, where the goal is to find a number of Pareto-optimal solutions in a single simulation run. However, none of the multi-objective 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. A new archiving strategy is proposed that maintains a subset of the generated solutions. It guarantees convergence and diversity according to well-defined criteria, i.e. ϵ-dominance and ϵ-Pareto optimality.

1 Introduction

After the doctoral study of Schaffer (1984) on the vector evaluated genetic algorithm (VEGA), Goldberg's suggestion of the use of non-dominated sorting along with a niching mechanism (1989) generated an overwhelming interest on multi-objective evolutionary algorithms (MOEAs). Initial MOEAs – MOGA (Fonseca and Fleming 1993), NSGA (Srinivas and Deb 1994), 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 was 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, in order to ensure convergence to the true Pareto-optimal solutions, an elite-preservation operator was absent in those algorithms. Thus, the latter MOEAs mainly concentrated on how elitism could be introduced in

an MOEA. This resulted in a number of advanced algorithms – SPEA (Zitzler and Thiele 1999), PAES (Knowles and Corne 2000), NSGA-II (Deb et al. 2000), and others. With the development of better algorithms, multi-objective evolutionary algorithms have also been used in a number of 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 (1998, 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 (and is dependent on the fitness landscape and genetic operators used) even in the case of single-objective evolutionary algorithms, the first task is as important as the task of converging to the true Pareto set. Deb (2001) suggested a steady-state MOEA that attempts to maintain spread while attempting to converge 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 monotony conditions. Convergence then is proven as a local optimum of the quality function will be reached, but how this optimum relates to the actual distribution of the solutions is unclear and the computational overhead is enormous. 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.

Algorithm 1 Iterative search procedure

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

Figure 1: Representation of the generic search algorithm 1.

In this paper, we propose an archiving/selection strategy that guarantees at the same time progress towards the Pareto-optimal set and a covering of the whole range of the non-dominated solutions. The algorithm maintains a finite-sized archive of non-dominated solutions which gets 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.

In the remainder of the paper, we state the general structure of an iterative archive-based search procedure which is usually used for multi-objective optimization. In section 3 we formally define our concepts of ϵ-dominance and ϵ-Pareto optimality. We present the new archiving algorithm and prove the required convergence and distribution properties. The simulation results in section 4 illustrate its practical relevance in contrast to existing algorithms which either fail with respect to convergence or to distribution behavior.

2 Structure of an Iterative Multi-Objective 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 which are 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 multi-objective 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
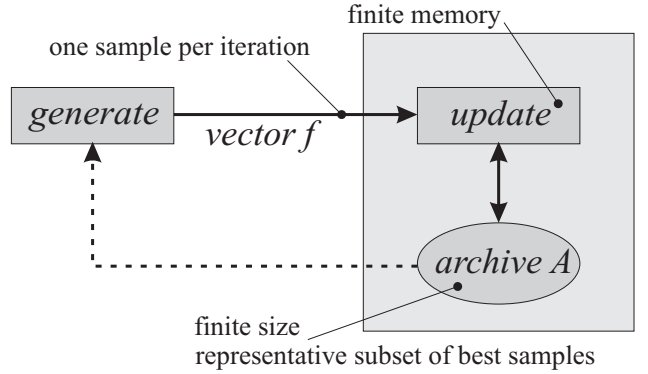
samples by n-dimensional real vectors f where each coordinate represents one of the objective values. Additional information about the corresponding decision values could be associated to f, but will be of no concern in this paper.

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, namely $A^{(t)}$. 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 the 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 easily be viewed as an evolutionary algorithm when the $generate$ operator is associated with variation (recombination and mutation). However, we would like to point out that all following investigations are equally valid for any kind of iterative process which can be described as Algorithm 1 and used for approximating the Pareto set of multi-objective optimization problems, e.g. simulated annealing or 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, as 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 solution set preference information could be used to steer the process to certain parts of the search space.

The paper solely deals with the function *update*, i.e. with an

appropriate generation of the archive. Because of the reasons described above, the corresponding algorithm should have the following properties, see also Fig. 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 3.1. But according to the common notion of optimality in multi-objective 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. Such an algorithm in will be constructed in section 3.3.

3 Algorithms for Convergence and Diversity

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

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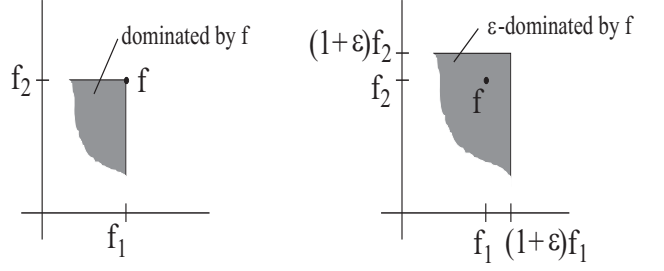
2. *$\exists j \in \{1, \ldots, m\} : f_j > g_j$*

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 $g \in F$ which are not dominated by any vector $f \in F$, i.e.

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

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 $g \in F \setminus F^*$ 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^* which *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 Fig. 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.

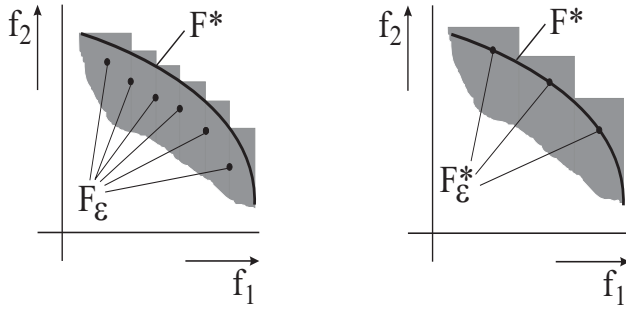$$\forall g \in F : \exists f \in F_\epsilon \quad \text{such that} \quad f \succ_\epsilon g. \tag{4}$$

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

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 Eqn. 7 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) have 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 non-dominated 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 $i \in \{1, \ldots, m\}$,

---
[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 the field of evolutionary computation.

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 Alg. 2 in section 3.3.

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 $P_\epsilon^*(F)$.

The above defined concepts are visualized in Fig. 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, an ϵ-Pareto set is more attractive as it consists of Pareto vectors only.

3.2 Convergence and Diversity

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.

Algorithm 2 *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** $\not\exists f' : box(f') = box(f) \vee box(f') \succ box(f)$ **then**
8:    $A' := A \cup \{f\}$
9: **else**
10:   $A' := A$
11: **end if**
12: **Output:** A'
___

Algorithm 3 function *box*
___
1: **Input:** f
2: **for all** $i \in \{1, \ldots, m\}$ **do**
3:    $b_i := \lfloor \frac{\log f_i}{\log (1+\epsilon)} \rfloor$
4: **end for**
5: $b := (b_1, \ldots, b_m)$
6: **Output:** b {box index vector}
___

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).

3.3 Maintaining an ϵ-Pareto Set

The Algorithm 2 has a two level concept. On the coarse level, the search space is discretized by a division into boxes (see Algorithm 3), where each vector uniquely belongs to one box. Using a generalized dominance relation on these boxes, the algorithm always maintains a set of non-dominated 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's 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.

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-Pareto set of $F^{(t)}$ with bounded size according to Eq. (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 Def. 5 this occurs only if some $f = f^{(\tau)}, \tau \leq t$ is (1) not ϵ-dominated by any member of $A^{(t)}$ and not in $A^{(t)}$ or (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 which 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 which 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 which 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.

$\square$

As a result, Algorithm 2 uses a finite memory, successively updates a finite subset of vectors that ϵ-dominate all vectors generated so far. It can 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 algorithm itself and are not required for the convergence proof (claim 1 of Theorem 1).

They are only utilized to prove the relation between ϵ and the size of the archive given in the second claim.

4 Simulations

This section presents some simulation results to demonstrate the behavior of the proposed algorithm for two example multi-objective optimization problems (MOPs). We use instances of the iterative search procedure (specified in Alg. 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.

4.1 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 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 Alg. 2 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: Non-dominated points – even those which belong to the "real" Pareto set – can get lost, and on the long run might even be replaced by dominated solutions. This is certainly not desirable, and algorithms relying on these strategies cannot be claimed to be convergent, even if the generator would be able to produce all elements of the Pareto set F^{*2}. In contrast, Alg. 2 is able to maintain an ϵ-Pareto set of the generated solutions over time.

---

[2]In our experiments almost all Pareto-optimal points have been produced by the generator within $t = 10,000,000$ iterations.
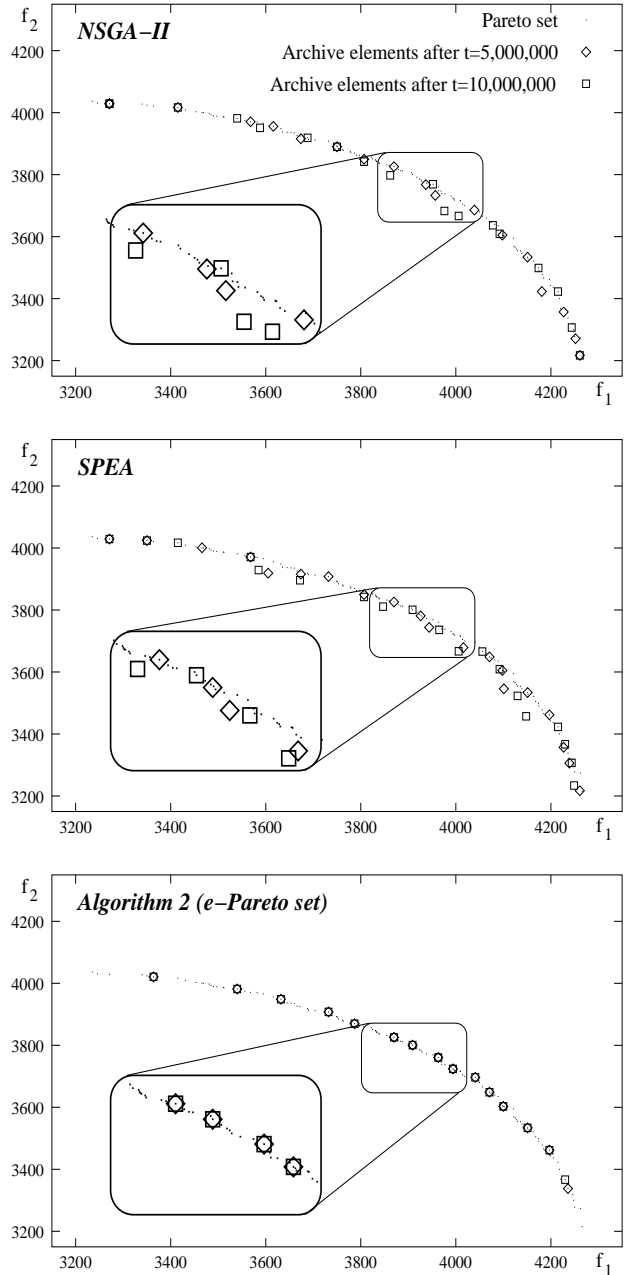

Figure 4: Objective space of the knapsack problem, the dots show the elements of the Pareto set F^*. The different figures correspond to different instances of the *update* operator in Alg. 1: NSGA-II (upper left), SPEA (upper right), and Alg. 2 (lower row). In each figure the archive set $A^{(t)}$ is shown, for $t = 5,000,000$ (with diamonds) and for $t = 10,000,000$ (with boxes). A subset of the samples is highlighted so visualize the negative effect of losing Pareto-optimal solutions in many current archiving/selection schemes.
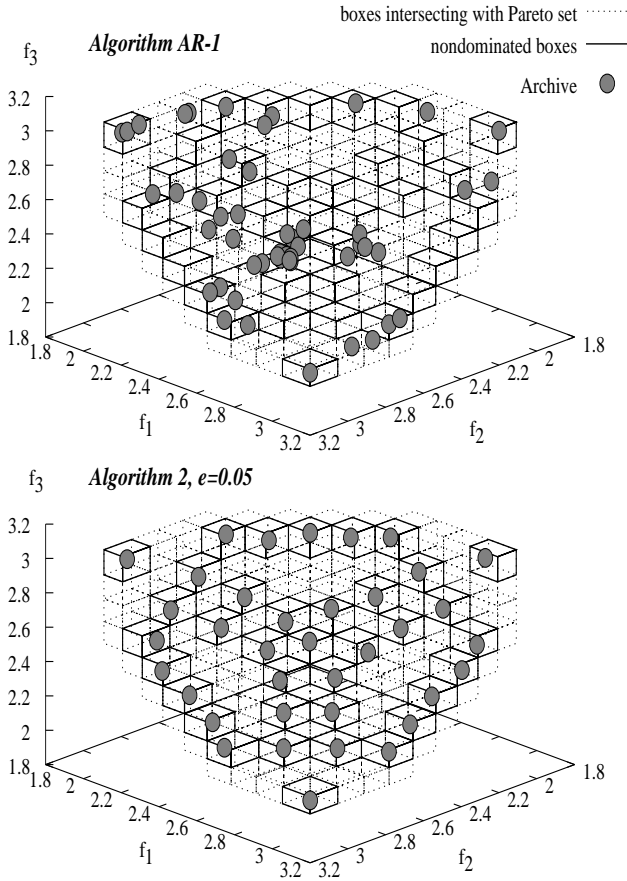
Figure 5: Objective space of MOP (8). The discretization into boxes according to Alg. 3 is indicated by showing all boxes that intersect with the Pareto set F^* in dashed lines. The non-dominated boxes are drawn in bold lines. The circles correspond to the output A of different instances of the iterative search algorithm Alg. 1. For the upper figure an update function according to AR-1 was used, for the lower figure the function according to Alg. 2.

4.2 Distribution Behavior

In order to test for the *distribution* behavior only candidates are taken into account which fulfill the requirements for convergence: Rudolph's and Agapie's algorithm AR-I (Rudolph and Agapie 2000) and Alg. 2. As a test case the following continuous three-dimensional three-objective problem is used:

$$
\begin{aligned}
\text{Max} \quad & f_1(x) = 3 - (1 + x_3)\cos(x_1\pi/2)\cos(x_2\pi/2), \\
\text{Max} \quad & f_2(x) = 3 - (1 + x_3)\cos(x_1\pi/2)\sin(x_2\pi/2), \\
\text{Max} \quad & f_3(x) = 3 - (1 + x_3)\cos(x_1\pi/2)\sin(x_1\pi/2), \\
& 0 \le x_i \le 1, \quad \text{for } i = 1, 2, 3,
\end{aligned}
\tag{8}
$$

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 with SBX (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 Alg. 3 with $\epsilon = 0.05$) is plotted instead of the actual Pareto set. As the multiplicative ϵ is used, it can be seen that the box sizes vary and reflect the relative deviations from different parts of the Pareto set. From all boxes intersecting with the Pareto set the non-dominated ones are highlighted. For an ϵ-approximate Pareto set it is now sufficient to have exactly one solution in each of those non-dominated boxes. This condition is fulfilled by the algorithm using the update strategy Alg. 2, leading to an almost symmetric distribution covering all regions. The strategy from AR-1, which does not discriminate among non-dominated 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 2, 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 3.2. They are not intended to give a uniform distribution on a (hypothetical) surface that might even not exist as in the discrete case.

4.3 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's 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.

- The algorithm proposed in this paper fulfills both the convergence criterion and the desired distribution control as it always maintains an ϵ-Pareto set of the generated solutions.

5 Possible Extensions

The above baseline algorithms can be extended in several interesting and useful ways. In the following we discuss two examples.

5.1 Steering Search by Defining Ranges of Non-acceptance

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

5.2 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 size. A concept like this is implemented in PAES, where the hyper-grid dividing the objective space is adapted to the current ranges given by the non-dominated vectors. However, PAES does not guarantee convergence.

The idea is to start with a minimal ϵ, which is systematically increased every time the number of archived vectors exceeds a predetermined maximum. In Algorithm 2, a simple modification would be to start with a minimal ϵ using a first pair of mutually non-dominated vectors. Afterwards, the increase of ϵ is taken care of by joining boxes and discarding all but the oldest element from the new 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 boundaries, 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 which 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 involve a complicated book-keeping of several different ϵ values in each dimension.

6 Conclusions

In this study we have addressed the problem of simultaneously achieving convergence and distribution quality when approximating Pareto sets of multi-objective optimization problems. It was shown that none of the existing multi-objective evolutionary algorithms is able to accomplish both tasks.

We proposed the ϵ-(approximate) Pareto set as a solution concept for evolutionary multi-objective optimization that

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

We constructed the first archive updating strategy that

- can be used in any iterative search process and
- allows 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 statements 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 (non-vanishing 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 all nice features as 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 maintaining an ϵ-Pareto set the user can be sure to have in addition to a representative, well distributed approximation also a true elitist algorithm in the sense that no better solution had been found and subsequently lost during the run.

Interesting behaviors occur 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. by Laumanns et al. (2000, 2001) using a more general model and a parameter called *elitism intensity*. Now, also the theoretical foundation is given so that the archived members are really guaranteed to be the best solutions found.

Acknowledgments

References

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

Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI)*, pp. 849–858.

Erlebach, T., H. Kellerer, and U. Pferschy (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 Vol. 2125, Berlin. Springer.

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

Fonseca, C. M. and P. J. Fleming (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423.

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

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

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

Horn, J., N. Nafploitis, and D. Goldberg (1994). A niched Pareto genetic algorithm for multi-objective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp. 82–87.

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

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

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

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

Papadimitriou, C. H. and M. Yannakakis (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*, pp. 86–92.

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

Rudolph, G. (1998). On a multi-objective evolutionary algorithm and its convergence to the pareto set. In *IEEE Int'l Conf. on Evolutionary Computation (ICEC'98)*, Piscataway, pp. 511–516. IEEE Press.

Rudolph, G. (2001). Evolutionary search under partially ordered fitness sets. In *Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems (ISI 2001)*, pp. 818–822.

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

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

Schaffer, J. D. (1984). *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*. Ph. D. thesis, Nashville, TN: Vanderbilt University.

Srinivas, N. and K. Deb (1994). Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal 2*(3), 221–248.

Zitzler, E., K. Deb, L. Thiele, C. A. C. Coello, and D. Corne (Eds.) (2001, March). *Evolutionary Multi-Criterion Optimization (EMO 2001)*, Lecture Notes in Computer Science Vol. 1993, Berlin, Germany. Springer.

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