On the Convergence and Diversity-Preservation Properties of Multi-Objective Evolutionary Algorithms

Marco Laumanns[1], Lothar Thiele[1], Kalyanmoy Deb[2], and Eckart Zitzler[1]

[1]Computer Engineering and Networks Laboratory (TIK)
Department of Electrical Engineering
Swiss Federal Institute of Technology (ETH) Zurich
CH-8092 Zürich, Switzerland
E-mail: {laumanns,thiele,zitzler}@tik.ee.ethz.ch

[2]Kanpur Genetic Algorithms Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
E-mail: deb@iitk.ac.in

TIK-Report No. 108
Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich
Gloriastrasse 35, ETH-Zentrum, CH–8092 Zürich, Switzerland

June 13, 2001

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. Many studies have depicted different ways evolutionary algorithms can progress towards the true Pareto-optimal solutions with a widely spread distribution of solutions. 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 and then suggest a class of archive-based MOEAs which can have both properties of converging to the true Pareto-optimal front and maintain a spread among obtained solutions. 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: Convergence, multi-objective optimization, preservation of diversity, evolutionary algorithms

1 Introduction

After the doctoral study of Dave Schaffer on vector evaluated genetic algorithm (VEGA) in 1984 [Sch84], Goldberg's suggestion of the use of non-dominated sorting along with a niching mechanism [Gol89] generated an overwhelming interest on multi-objective evolutionary algorithms (MOEAs). Initial MOEAs – MOGA [FF93], NSGA [SD94], NPGA [HNG94] – 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, 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 better algorithms – SPEA [ZT99], PAES [KC00], NSGA-II [DAPM00], and others. With the development of better algorithms, multi-objective evolutionary algorithms have also been used in a number of interesting application case studies [ZDT$^+$01].

However, what severely lack are studies related to theoretical convergence analysis with guaranteed spread of solutions. In this regard, Rudolph [Rud94, Rud01] suggested a series of algorithms, all of which guarantee converging on the true Pareto-optimal set. However, all these algorithms lack 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-optimal set. Deb [Deb01] suggested a steady-state MOEA which attempts to maintain spread while attempting to converge to the true Pareto-optimal front. But there is no proof for its convergence properties.

In this paper, we propose a class of MOEAs, which in addition to progressing towards the true Pareto-optimal set also maintain a spread among solutions. The algorithm maintains a finite-sized archive of non-dominated solutions, which gets iteratively updated in the presence of every new solution. The non-domination check is performed with a new ϵ-dominance operator, which ensures that no two neighboring solutions (within a ϵ-neighborhood) are non-dominated to each other. The use of ϵ-dominance also makes the algorithms practical by allowing a decision-maker to control the difference in objective function values (by choosing an appropriate ϵ) between two consecutive Pareto-optimal solutions. The baseline algorithm suggested here is a generic convergent MOEA with a guaranteed spread of solutions. It also results in the Rudolph's convergent algorithm as a special case. To make the baseline algorithm more useful in practice, we have also suggested a number of modifications.

In the remainder of the paper, we state the structure of archive-based algorithm for multi-objective optimization. Thereafter, we briefly review the existing MOEAs and discuss why they do not have the theoretical convergent as well as the diversity-preservation properties. Then, we define ϵ-dominance and the corresponding ϵ-Pareto-optimal set. Finally, we suggest a class of algorithms which have both properties by guaranteeing convergence to a diverse set of ϵ-Pareto-optimal solutions. The proposed convergent MOEAs are interesting and should make MOEAs more useful and attractive to both theoreticians and practitioners in the coming years.

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 algorithms which are characterized by the generation of a sequence of search points and a finite memory called population or archive.

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.

Algorithm 1 Iterative search algorithm

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

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 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) and the *update* operator with selection. 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.

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. In addition, the value of presenting such a large set of solutions to a decision maker is doubtful in the

context of desicion 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.

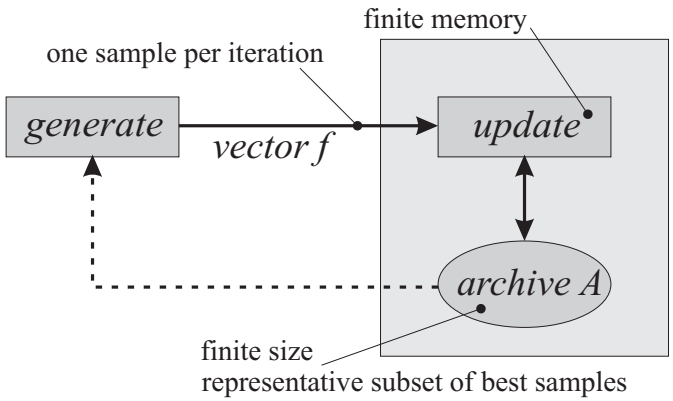

Figure 1: Representation of the generic search algorithm 1.

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 4.1. 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.

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

3 Existing Multi-Objective Algorithms and Their Limitations

Here, we discuss a number of archive-based update functions which are suggested in the context of multi-objective evolutionary algorithms (MOEAs).

3.1 Rudolph and Agapie's Elitst GAs

Rudolph and Agapie [RA00] suggested an archive-based MOEA (they called AR-1), which maintains a fixed-size archive consisting of non-dominated solutions found till the current generation. First, the non-dominated solutions of the child population is selected. Then, each such child solution is checked against the current archive members. If the child solution dominates any archive member, it is removed from the archive and the child solution is added to the archive. On the other hand, if the child solution is non-dominated with all archive members, it is kept in a temporary population for a possibility of its inclusion in the archive. When all non-dominated child solutions are checked for their dominance with respect to the archive, only that many temporary population members are included in the archive as are necessary to fill the archive to its pre-specified fixed size. The investigators have shown that this algorithm with variation operators having a positive-definiteness of its transition probability matrix has a guaranteed convergence property to the true 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 algorithm guarantees convergence to the true Pareto-optimal front, it does not guarantee a good distribution of Pareto-optimal solutions.

In their AR-2 algorithm, the investigators have suggested creating the child population from the non-dominated solutions of the parent population, instead of the complete parent population as in AR-1. This algorithm also does not guarantee finding a good distribution of solutions, because of the identical reasons given above for AR-1.

3.2 Pareto-Archived Evolution Strategy (PAES)

Knowles and Corne [KC00] suggested a simple MOEA using a single parent, single child evolutionary algorithm, similar to (1+1)-evolution strategy. In their Pareto-archived ES (PAES), the child is compared with respect to the parent. If the child dominates the parent, the child is accepted as the next parent and the iteration continues. On the other hand, if the parent dominates the child, the child is discarded and a new mutated solution (a new

child) is found. However, if the child and the parent do not dominate each other, the choice between the child and the parent is made by comparing them with an archive of best solutions found so far. The child is compared with the archive to check if it dominates any member of the archive. If yes, the child is accepted as the new parent and all the dominated solutions are eliminated from the archive. If the child does not dominate any member of the archive, both parent and child are checked for their *nearness* with the solutions of the archive. If the child resides in a least crowded region in the parameter space among the members of the archive, it is accepted as a parent and a copy is added to the archive. If the child and the parent have the same nearness count, one is chosen at random. Crowding is maintained by deterministically dividing the entire search space in d^n subspaces (where d is the depth parameter and n is the number of decision variables) and by updating the subspaces dynamically. Authors have later proposed an improved method such as Pareto-envelope based selection algorithm or PESA, which uses the subpopulation stored in each grid location to control the selection pressure and diversity of population members [CKO00].

The above algorithm clearly prefers non-dominated solutions, an essential property for an MOEA to progress towards the true Pareto-optimal front. The diversity is also maintained by the deterministic crowding approach. However, the algorithm lacks a convergence proof, simply because a non-Pareto-optimal yet non-dominated child can replace a parent which is already a Pareto-optimal solution. Such a temporary digress from the Pareto-optimal set can be overcome by re-discovering a dominant Pareto-optimal parent solution later, but the probability of such occurrences reduces as the child moves closer to the true Pareto-optimal front.

3.3 Strength Pareto Evolutionary Algorithm (SPEA)

Zitzler and Thiele [ZT99] have suggested an elitist MOEA with the concept of non-domination. They suggested maintaining an external population at every generation storing a set of non-dominated solutions discovered so far beginning from the initial population. This external population participates in genetic operations. The fitness of each individual in the current population and in the external population is decided based on the number of dominated solutions. Specifically, the following procedure is adopted. A combined population with the external and the current population is first constructed. All non-dominated solutions in the combined population are assigned a fitness based on the number of solutions they dominate. To maintain diversity and in the context of minimizing the fitness function, they assigned more fitness to a non-dominated solution having more dominated solutions in the combined population. On the other hand, more fitness is

also assigned to solutions dominated by more solutions in the combined population. Care is taken to assign no non-dominated solution a fitness worse than that of the best dominated solution. This assignment of fitness makes sure that the search is directed towards the non-dominated solutions and simultaneously diversity among dominated and non-dominated solutions are maintained. A clustering mechanism groups all currently non-dominated solutions into a pre-defined number of clusters and picks a representative solution from each cluster, thereby ensuring diversity among the external population members.

Here too, the fitness assignment based on non-domination ensures progress towards the true Pareto-optimal set. The use of clustering among external population members also guarantees spread among them. However, the algorithm also lacks a convergence proof, simply because during the clustering procedure an existent Pareto-optimal external population member may get replaced by a non-Pareto-optimal yet less-crowded non-dominated solution. As argued above, although this temporary digress may be corrected by re-discovering a Pareto-optimal solution later, the probability of such occurrences reduce as the generations proceed.

3.4 Elitist Non-Dominated Sorting GA (NSGA-II)

NSGA-II [DAPM00] is a modified NSGA, which uses (i) a fast non-dominated sorting approach, (ii) an elitist approach, and (iii) an efficient crowding approach. By using a better book-keeping strategy, the non-dominated sorting approach is reduced to $O(MN^2)$, where M is the number of objectives and the N is the population size. The elitist approach is also simple. After the children population (of size N) is formed from the parent population, both these populations are combined and a non-dominated sorting is performed. Thereafter, solutions from better non-dominated sets are accepted, one set at a time till the population is filled. In the event of an inadequate available population slots to accommodate all solutions of a non-dominated set, a crowding strategy is used to identify solutions which resides in a less-crowded area. This way, better solutions from the previous iteration get emphasized and a better spread of solutions is maintained by allowing less-crowded solutions to remain in the population.

Early on, when many non-dominated sets prevail the population, NSGA-II provides a search direction towards the Pareto-optimal region by selecting solutions lying on better non-dominated regions. Later, when most solutions reside on or near the Pareto-optimal front, not much selective pressure comes from domination checks, rather selection pressure builds up for lone solutions. This dual-procedure allows both progress towards the Pareto-optimal set and spread of solutions. However, NSGA-II also does not have

the true convergence property. When more than N population members of the combined population belong to the non-dominated set, only those that are maximally apart from their neighbors are chosen. This way, like PAES and SPEA, an existent Pareto-optimal solution may get replaced by a non-Pareto-optimal yet dominated solution during the crowding selection operation.

3.5 Limitations

Thus, it is clear from the above discussion that the above elitist MOEAs can do well in progressing closer to the Pareto-optimal set with a good distribution of solutions. But none of them guarantee true convergence to the Pareto-optimal set. The diversity-preservation operator used in each of the above algorithms are only 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 Pareto-optimal. The diversity-preservation operator always emphasizes the less crowded solutions of the non-dominated solutions. A part of the difficulty arises because of the fact that the update functions of these MOEAs are designed to emphasize a non-dominated set of solutions. When some members of a non-dominated set are Pareto-optimal, it is not necessary that all other members are also Pareto-optimal. The update functions of the above MOEAs emphasizes all non-dominated solutions uniformly, while the diversity-preservation operator treats all such non-dominated solutions equally important to be replaced. In the absence of any knowledge of the true Pareto-optimal front, the above MOEAs rightfully perform the above two tasks in a unbiased manner. Unfortunately, as discussed above, the absence of such knowledge, such unbiased operations also make the above MOEAs theoretically non-convergent.

4 Algorithms for Convergence and Diversity

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

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 [Deb01] 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. $\exists j \in \{1, \ldots, m\} : f_j > g_j$

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

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

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

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

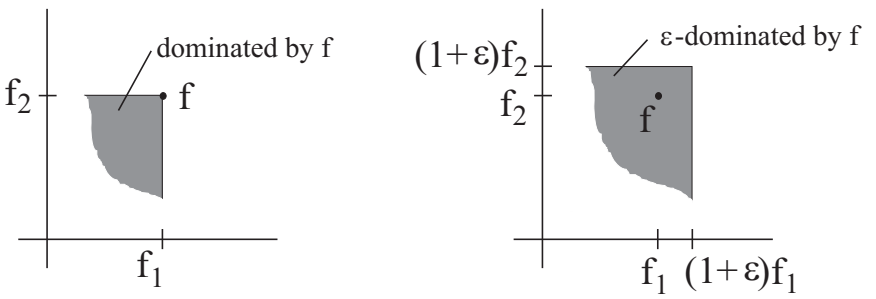

Figure 2: Graphs visualizing the concepts of dominance and ϵ-dominance.

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 size of 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 $F_\epsilon \subseteq F$ is called an ϵ-approximate Pareto set, 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. The concept of approximate Pareto sets has been described in detail in [PY00, EKP01]. It has also been shown that under certain assumptions there is always an approximate Pareto set which is polynomial in size, dependent on the approximation factor ϵ. In particular, by dividing the objective space into boxes, 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

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

vectors.

Note that this concept 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.

$$|F_\epsilon| = O\left[\prod_{j=1}^{m-1} \frac{2K}{\epsilon_i}\right] \tag{7}$$

where $-K \leq f_i \leq K$, $K \geq \epsilon_i$ 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.

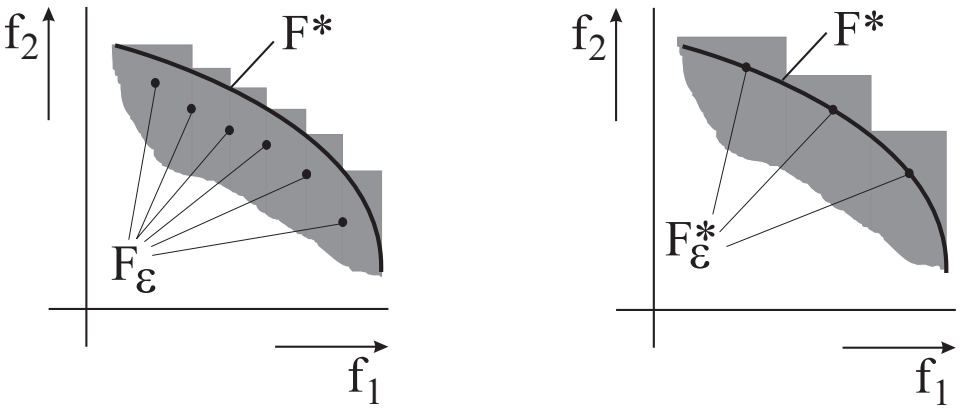

Figure 3: Graphs visualizing the concepts of Pareto set, ϵ-approximate Pareto set and ϵ-Pareto set.

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.

4.2 Generating 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.

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'

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 according to Eq. (5), i.e.

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

Proof.

1. Suppose the algorithm is not correct, i.e. $A^{(t)} \notin P_\epsilon(F^{(t)})$ for some t. According to Def. 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. Since the dominance relation is transitive, and since it implies 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 $f' \in A^{(\tau)}$ which ϵ-dominates f and – with the same argument as before – can only be replaced by accepting elements which 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 [RA00] are 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 non-dominated vectors. Of course, according to the previous corollary, the size of this set might grow to infinity as $t \to \infty$.

4.3 Generating an ϵ-Pareto set

In a next step we would like to guarantee – in addition to a minimal lateral distance of 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 divided 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 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.

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: $\quad A' := A \cup \{f\} \setminus D$
5: **else if** $\exists f' : (box(f') = box(f) \wedge f \succ f')$ **then**
6: $\quad A' := A \cup \{f\} \setminus \{f'\}$
7: **else if** $\nexists f' : box(f') = box(f) \vee box(f') \succ box(f)$ **then**
8: $\quad A' := A \cup \{f\}$
9: **else**
10: $\quad A' := A$
11: **end if**
12: **Output:** A'

Algorithm 4 function *box*

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

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 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 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 either dominates f or whose box value dominates that of f. Since both relations are transitive, and since they both imply ϵ-dominance, there will always be 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. 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 generaliy – 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, successively update a finite subset of Pareto vectors of all vectors generated so far and ϵ-dominates all vectors generated so far.

5 Possible Extensions

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

5.1 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 solution presented above will get modified accordingly.

5.2 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 ob-

tained by the proposed algorithms. Although a 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, the 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.

5.3 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 of not importance to a decision-maker, the above algorithms can be used to search along preferred regions. The concept of ϵ-dominance will then allow pre-specified precisions to exist among the obtained preferred Pareto-optimal vectors.

5.4 Fixed archive size by dynamic adaptation

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 hypergrid dividing the objective space is adapted to the current ranges given by the non-dominated vectors. However, PAES does not guarantee convergence.

Here two modified version 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 exceed a predetermined maximum size.

5.4.1 Generating 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 non-dominating vectors have been found, an initial ϵ is calculated with Eq. 5 as

$$\epsilon = K \frac{1}{a^{1/m}} \tag{8}$$

where a is the maximum archive size and K is set to the maximum ratio of the components

$$K := \max \left\{ \delta f_i | \delta f_i = \left(\frac{g_i^{(1)}}{g_i^{(2)}} \right)^{sgn(g_i^{(1)} - g_i^{(2)})} \right\} \tag{9}$$

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 formula. Using 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.

Agapie's and Rudolph's algorithms AR-1 and AR-2 could be seen as a special cases of this algorithm for $\epsilon = 0$, which means that the guaranteed minimal distance of vectors is also zero, hence not guaranteeing to maintain an ϵ-approximate Pareto set.

5.4.2 Generating an ϵ-Pareto set

In Algorithm 3 a simple modification would be to start with a minimal ϵ using a first pair of mutually non-dominated 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 in the 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 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 make a complicated book-keeping of several different ϵ values in each dimension necessary.

5.4.3 A bi-start 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: At 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 and hence ensuring that the size of the solution set does not exceed the limit a_{max}.

Acknowledgments

The work has been supported by the Swiss Science Foundation.

References

[CKO00] D. Corne, J. Knowles, and M. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature VI (PPSN-VI)*, pages 839–848, 2000.

[DAPM00] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. 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)*, pages 849–858, 2000.

[Deb01] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[EKP01] Thomas Erlebach, Hans Kellerer, and Ulrich Pferschy. Approximating multi-objective knapsack problems. In *Proceedings of the Seventh International Workshop on Algorithms and Data Structures (WADS 2001)*, 2001. to appear.

[FF93] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.

[Gol89] D. E. Goldberg. *Genetic Algorithms for Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

[HNG94] J. Horn, N. Nafploitis, and D.E. Goldberg. A niched Pareto genetic algorithm for multi-objective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, 1994.

[KC00] Joshua D. Knowles and David W. Corne. Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation Journal*, 8(2):149–172, 2000.

[PY00] C. H. Papadimitriou and M. Yannakakis. The complexity of tradeoffs, and optimal access of web sources. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science FOCS'00*, 2000.

[RA00] G. Rudolph and A. Agapie. Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC2000)*, pages 1010–1016, 2000.

[Rud94] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Network*, 5(1):96–101, 1994.

[Rud01] G. Rudolph. 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)*, pages 818–822, 2001.

[Sch84] J. D. Schaffer. *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms.* PhD thesis, Nashville, TN: Vanderbilt University, 1984.

[SD94] N. Srinivas and K. Deb. Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal*, 2(3):221–248, 1994.

[ZDT+01] E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, editors. *Evolutionary Multi-Criterion Optimization (Lecture Notes in Computer Science 1993).* Heidelberg: Springer, 2001.

[ZT99] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.