SPEA2: Improving the Strength Pareto Evolutionary Algorithm

Eckart Zitzler, Marco Laumanns, and Lothar Thiele
Computer Engineering and Networks Laboratory (TIK)
Department of Electrical Engineering
Swiss Federal Institute of Technology (ETH) Zurich
ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switzerland
`{zitzler,laumanns,thiele}@tik.ee.ethz.ch`

Abstract

The Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999) is a relatively recent technique for finding or approximating the Pareto-optimal set for multiobjective optimization problems. In different studies (Zitzler and Thiele 1999; Zitzler, Deb, and Thiele 2000) SPEA has shown very good performance in comparison to other multiobjective evolutionary algorithms, and therefore it has been a point of reference in various recent investigations, e.g., (Corne, Knowles, and Oates 2000). Furthermore, it has been used in different applications, e.g., (Lahanas, Milickovic, Baltas, and Zamboglou 2001). In this paper, an improved version, namely SPEA2, is proposed, which incorporates in contrast to its predecessor a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. The comparison of SPEA2 with SPEA and two other modern elitist methods, PESA and NSGA-II, on different test problems yields promising results.

1 Introduction

After the first studies on evolutionary multiobjective optimization (EMO) in the mid-1980s, a number of Pareto-based techniques were proposed in 1993 and 1994, e.g., MOGA (Fonseca and Fleming 1993), NPGA (Horn, Nafpliotis, and Goldberg 1994), and NSGA (Srinivas and Deb 1994), which demonstrated the capability of EMO algorithms to approximate the set of optimal trade-offs in a single optimization run. These approaches did not incorporate elitism explicitly, but a few years later the importance

of this concept in multiobjective search was recognized and supported experimentally (Parks and Miller 1998; Zitzler, Deb, and Thiele 2000). A couple of elitist multi-objective evolutionary algorithms were presented at this time, e.g., SPEA (Zitzler and Thiele 1998; Zitzler and Thiele 1999) and PAES (Knowles and Corne 1999). SPEA, an acronym for Strength Pareto Evolutionary Algorithm, was among the first techniques that were extensively compared to several existing evolution-based methods (Zitzler and Thiele 1999; Zitzler, Deb, and Thiele 2000). As it clearly outperformed the (non-elitist) alternative approaches under consideration, it has been used as a point of reference by various researchers, e.g., (Corne, Knowles, and Oates 2000; Jaszkiewicz 2000; Tan, Lee, and Khor 2001). Meanwhile further progress has been made and recently proposed methods, for instance NSGA-II (Deb, Agrawal, Pratap, and Meyarivan 2000) and PESA (Corne, Knowles, and Oates 2000), were shown to outperform SPEA on certain test problems. Furthermore, new insights into the behavior of EMO algorithms improved our knowledge about the basic principles and the main factors of success in EMO (Laumanns, Zitzler, and Thiele 2000; Laumanns, Zitzler, and Thiele 2001).

In this paper, SPEA2 is presented, for which we tried to eliminate the potential weaknesses of its predecessor and to incorporate most recent results in order to design a powerful and up-to-date EMO algorithm. The main differences of SPEA2 in comparison to SPEA are:

- An improved fitness assignment scheme is used, which takes for each individual into account how many individuals it dominates and it is dominated by.

- A nearest neighbor density estimation technique is incorporated which allows a more precise guidance of the search process.

- A new archive truncation methods guarantees the preservation of boundary solutions.

As will be shown in this study, the proposed algorithm provides good performance in terms of convergence and diversity, outperforms SPEA, and compares well to PESA and NSGA-II on various, well-known test problems.

2 Background

2.1 Issues in Evolutionary Multiobjective Optimization

The approximation of the Pareto-optimal set involves itself two (possibly conflicting) objectives: the distance to the optimal front is to be minimized and the diversity of the generated solutions is to be maximized (in terms of objective or parameter values). In this context, there are two fundamental issues when designing a multiobjective evolutionary algorithm: mating selection and environmental selection. The first issue is directly related to the question of how to guide the search towards the Pareto-optimal front. Given a pool of individuals, fitness values have to be assigned on the basis of which individuals for offspring production are selected. The procedure to fill the mating pool is usually randomized. The second issue addresses the question of which

individuals to keep during the evolution process. Due to limited time and storage resources, only a certain fraction of the individuals in a specific generation can be copied to the pool of the next generation. It is common practice to use a deterministic selection here.

In most modern EMO algorithms these two concepts are realized in the following way although the details may be different:

Environmental selection: Besides the population, an archive is maintained which contains a representation of the nondominated front among all solutions considered so far. A member of the archive is only removed if i) a solution has been found that dominates it or ii) the maximum archive size is exceeded and the portion of the front where the archive member is located is overcrowded. Usually, being copied to the archive is the only way how an individual can survive several generations in addition to pure reproduction which may occur by chance. This technique is incorporated in order not to lose certain portions of the current nondominated front due to random effects.

Mating selection: The pool of individuals at each generation is evaluated in a two stage process. First all individuals are compared on the basis of the Pareto dominance relation, which defines a partial order on this multi-set. Basically, the information which individuals each individual dominates, is dominated by or is indifferent to is used to define a ranking on the generation pool. Afterwards, this ranking is refined by the incorporation of density information. Various density estimation techniques are used to measure the size of the niche in which a specific individual is located.

In principle, both selection schemes are completely independent from each other. Thus, the first may be Pareto-based while the second can use the weighting approach to calculate fitness values. However, with many evolutionary methods both concepts are implemented similarly as will be illustrated on the basis of two recent EMO methods.

In PESA (Pareto Envelope-Based Selection Algorithm) (Corne, Knowles, and Oates 2000), for instance, mating selection is only performed on the archive which stores (a subset of) the current nondominated set. A particular density measure which can be classified as a histogram technique, allows to sample the archive members differently according to the degree of crowding. The generated children, which constitute the actual population, are then checked for inclusion into the archive (as described above). Those individuals which did not enter the archive are removed finally before the next generational cycle starts. With this approach, mating and environmental selection are identical regarding the selection criteria (member of the current nondominated front, crowding measure) and only differ with respect to the selection process (randomized versus deterministic). The same holds for another promising algorithm, NSGA-II (Nondominated Sorting Genetic Algorithm) (Deb, Agrawal, Pratap, and Meyarivan 2000). Here, the pool of individuals is first split into different fronts according to the concept of Pareto dominance. Individuals belonging to the first nondominated front are assigned highest rank, those in the second nondominated front the second highest rank and so forth. Within each rank, a specific crowding measure, which represents the sum of distances to the two closest individuals along each objective, is used to define

an order among the individuals. On the basis of this ranking both environmental and mating selection are performed. By combining parent population (which can actually be regarded as the archive) and offspring population and then deleting the worst 50%, the pool of individuals is truncated. Afterwards binary tournaments are carried out on the remaining individuals (the archive members) in order to generate the next offspring population. Note that the archive may not only contain nondominated individuals but also dominated ones in contrast to PESA; with NSGA-II the archive is always filled completely, while with PESA it may be filled only partially.

2.2 The Strength Pareto Evolutionary Algorithm

As SPEA (Strength Pareto Evolutionary Algorithm) (Zitzler and Thiele 1999) forms the basis for SPEA2, we give a brief summary of the algorithm here. For a more detailed description the interested reader is referred to (Zitzler 1999).

SPEA uses a regular population and an archive (external set). Starting with an initial population and an empty archive the following steps are performed per iteration. First, all nondominated population members are copied to the archive; any dominated individuals or duplicates (regarding the objective values) are removed from the archive during this update operation. If the size of the updated archive exceeds a predefined limit, further archive members are deleted by a ==clustering technique== which preserves the characteristics of the nondominated front. Afterwards, fitness values are assigned to both archive and population members:

- Each individual i in the archive is assigned a strength value $S(i) \in [0, 1)$, which at the same time represents its fitness value $F(i)$. $S(i)$ is the number of population members j that are dominated by or equal to i with respect to the objective values, divided by the population size plus one.

- The fitness $F(j)$ of an individual j in the population is calculated by summing the strength values $S(i)$ of all archive members i that dominate or are equal to j, and adding one at the end.

The next step represents the mating selection phase where individuals from the union of population and archive are selected by means of binary tournaments. Please note that fitness is to be minimized here, i.e., each individual in the archive has a higher chance to be selected than any population member. Finally, after recombination and mutation the old population is replaced by the resulting offspring population.

Although SPEA performed well in different comparative studies (Zitzler and Thiele 1999; Zitzler, Deb, and Thiele 2000), there is still room for improvement as recent studies (Corne, Knowles, and Oates 2000; Deb, Agrawal, Pratap, and Meyarivan 2000) have shown. In particular, we have identified the following issues as potential weaknesses of SPEA:

Fitness assignment: Individuals that are dominated by the same archive members have identical fitness values. That means in the case when the archive contains only a single individual, all population members have the same rank independent of whether they dominate each other or not. As a consequence, the selection

4

[Handwritten margin annotations:]

The relation network could be considered as the network of the strong combinations in one individual. The strong combination in a individual is the genes which appear regularly, and that forms a pattern.

the fitness value here is different to the objective value.

limit the S(j) in the [1, 0]

the weaker the individual in the archive, the smaller the evolution pressure on the individual of the population.

does not dominated by each other

why adding one?

pressure is decreased substantially and in this particular case SPEA behaves like a random search algorithm.

Density estimation: If many individuals of the current generation are indifferent, i.e., do not dominate each other, none or very little information can be obtained on the basis of the partial order defined by the dominance relation. In this situation, which is very likely to occur in the presence of more than two objectives, density information has to be used in order to guide the search more effectively. Clustering makes use of this information, but only with regard to the archive and not to the population.

Archive truncation: Although the clustering technique used in SPEA is able to reduce the nondominated set without destroying its characteristics, it may lose outer solutions. However, these solutions should be kept in the archive in order to obtain a good spread of nondominated solutions.

In the next section we will address these issues and describe the improved algorithm, which we call SPEA2, in detail.

3 The SPEA2 Algorithm

SPEA2 was designed to overcome the aforementioned problems. The overall algorithm is as follows:

Algorithm 1 (SPEA2 Main Loop)

Input:	N	*(population size)*
	$\overline{N}$	*(archive size)*
	T	*(maximum number of generations)*
Output:	$\boldsymbol{A}$	*(nondominated set)*

Step 1: **Initialization**: *Generate an initial population $\boldsymbol{P}_0$ and create the empty archive (external set) $\overline{\boldsymbol{P}}_0 = \emptyset$. Set $t = 0$.*

Step 2: **Fitness assignment**: *Calculate fitness values of individuals in $\boldsymbol{P}_t$ and $\overline{\boldsymbol{P}}_t$ (cf. Section 3.1).*

Step 3: **Environmental selection**: *Copy all nondominated individuals in $\boldsymbol{P}_t$ and $\overline{\boldsymbol{P}}_t$ to $\overline{\boldsymbol{P}}_{t+1}$. If size of $\overline{\boldsymbol{P}}_{t+1}$ exceeds $\overline{N}$ then reduce $\overline{\boldsymbol{P}}_{t+1}$ by means of the truncation operator, otherwise if size of $\overline{\boldsymbol{P}}_{t+1}$ is less than $\overline{N}$ then fill $\overline{\boldsymbol{P}}_{t+1}$ with dominated individuals in $\boldsymbol{P}_t$ and $\overline{\boldsymbol{P}}_t$ (cf. Section 3.2).*

Step 4: **Termination**: *If $t \geq T$ or another stopping criterion is satisfied then set $\boldsymbol{A}$ to the set of decision vectors represented by the nondominated individuals in $\overline{\boldsymbol{P}}_{t+1}$. Stop.*

Step 5: **Mating selection**: *Perform binary tournament selection with replacement on $\overline{\boldsymbol{P}}_{t+1}$ in order to fill the mating pool.*
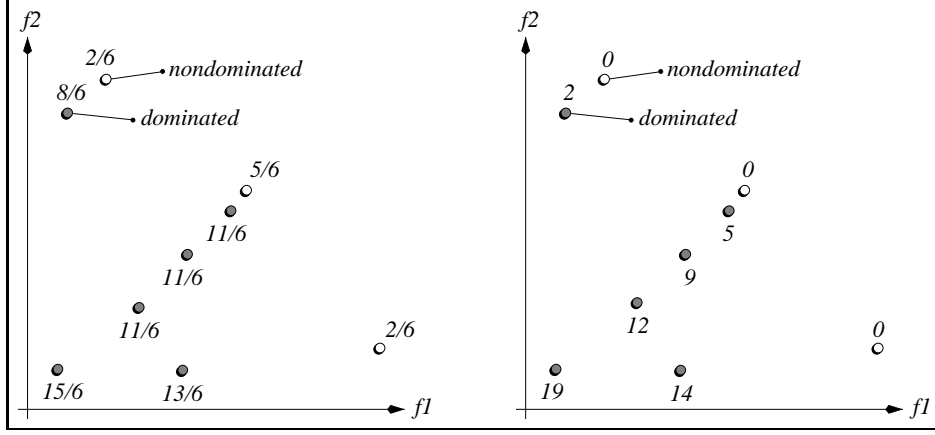
Figure 1: Comparison of fitness assignment schemes in SPEA and SPEA2 for a maximization problem with two objectives f_1 and f_2. On the left, the fitness values for a given population according to the SPEA scheme is shown. On the right, the raw SPEA2 fitness values for the same population are depicted.

Step 6: **Variation**: *Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t = t + 1$) and go to Step 2.*

In contrast to SPEA, SPEA2 uses a fine-grained fitness assignment strategy which incorporates density information as will be described in Section 3.1. Furthermore, the archive size is fixed, i.e., whenever the number of nondominated individuals is less than the predefined archive size, the archive is filled up by dominated individuals; with SPEA, the archive size may vary over time. In addition, the clustering technique, which is invoked when the nondominated front exceeds the archive limit, has been replaced by an alternative truncation method which has similar features but does not loose boundary points. Details on the environmental selection procedure will be given in Section 3.2. Finally, another difference to SPEA is that only members of the archive participate in the mating selection process.

3.1 Fitness Assignment

To avoid the situation that individuals dominated by the same archive members have identical fitness values, with SPEA2 for each individual both dominating and dominated solutions are taken into account. In detail, each individual i in the archive $\overline{P}_t$ *and* the population P_t is assigned a strength value $S(i)$, representing the number of

6

solutions it dominates:[1]

$$S(\boldsymbol{i}) = |\{\boldsymbol{j} \mid \boldsymbol{j} \in \boldsymbol{P}_t + \overline{\boldsymbol{P}}_t \wedge \boldsymbol{i} \succ \boldsymbol{j}\}|$$

where $|\cdot|$ denotes the cardinality of a set, $+$ stands for multiset union and the symbol $\succ$ corresponds to the Pareto dominance relation. On the basis of the S values, the raw fitness $R(\boldsymbol{i})$ of an individual $\boldsymbol{i}$ is calculated:

$$R(\boldsymbol{i}) = \sum_{\boldsymbol{j} \in \boldsymbol{P}_t + \overline{\boldsymbol{P}}_t, \boldsymbol{j} \succ \boldsymbol{i}} S(\boldsymbol{j})$$

That is the raw fitness is determined by the strengths of its dominators in both archive and population, as opposed to SPEA where only archive members are considered in this context. It is important to note that fitness is to be minimized here, i.e., $R(\boldsymbol{i}) = 0$ corresponds to a nondominated individual, while a high $R(\boldsymbol{i})$ value means that $\boldsymbol{i}$ is dominated by many individuals (which in turn dominate many individuals). This scheme is illustrated in Figure 1.

Although the raw fitness assignment provides a sort of niching mechanism based on the concept of Pareto dominance, it may fail when most individuals do not dominate each other. Therefore, additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique used in SPEA2 is an adaptation of the k-th nearest neighbor method (Silverman 1986), where the density at any point is a (decreasing) function of the distance to the k-th nearest data point. Here, we simply take the inverse of the distance to the k-th nearest neighbor as the density estimate. To be more precise, for each individual $\boldsymbol{i}$ the distances (in objective space) to all individuals $\boldsymbol{j}$ in archive and population are calculated and stored in a list. After sorting the list in increasing order, the k-th element gives the distance sought, denoted as $\sigma_{\boldsymbol{i}}^k$. As a common setting, we use k equal to the square root of the sample size (Silverman 1986), thus, $k = \sqrt{N + \overline{N}}$. Afterwards, the density $D(\boldsymbol{i})$ corresponding to $\boldsymbol{i}$ is defined by

$$D(\boldsymbol{i}) = \frac{1}{\sigma_{\boldsymbol{i}}^k + 2}$$

In the denominator, two is added to ensure that its value is greater than zero and that $D(\boldsymbol{i}) < 1$. Finally, adding $D(\boldsymbol{i})$ to the raw fitness value $R(\boldsymbol{i})$ of an individual $\boldsymbol{i}$ yields its fitness $F(\boldsymbol{i})$:

$$F(\boldsymbol{i}) = R(\boldsymbol{i}) + D(\boldsymbol{i})$$

The run-time of the fitness assignment procedure is dominated by the density estimator ($\mathcal{O}(M^2 \log M)$), while the calculation of the S and R values is of complexity $\mathcal{O}(M^2)$, where $M = N + \overline{N}$.

---

[1]This (and the following) formula slightly differs from the one presented in (Bleuler, Brack, Thiele, and Zitzler 2001), where also individuals which have identical objective values contribute to the strength of an individual.

3.2 Environmental Selection

The archive update operation (Step 3 in Algorithm 1) in SPEA2 differs from the one in SPEA in two respects: i) the number of individuals contained in the archive is constant over time, and ii) the truncation method prevents boundary solutions being removed.

During environmental selection, the first step is to copy all nondominated individuals, i.e., those which have a fitness lower than one, from archive and population to the archive of the next generation:

$$\overline{\boldsymbol{P}}_{t+1} = \{\boldsymbol{i} \mid \boldsymbol{i} \in \boldsymbol{P}_t + \overline{\boldsymbol{P}}_t \wedge F(\boldsymbol{i}) < 1\}$$

If the nondominated front fits exactly into the archive ($|\overline{\boldsymbol{P}}_{t+1}| = \overline{N}$) the environmental selection step is completed. Otherwise, there can be two situations: Either the archive is too small ($|\overline{\boldsymbol{P}}_{t+1}| < \overline{N}$) or too large ($|\overline{\boldsymbol{P}}_{t+1}| > \overline{N}$). In the first case, the best $\overline{N} - |\overline{\boldsymbol{P}}_{t+1}|$ dominated individuals in the previous archive and population are copied to the new archive. This can be implemented by sorting the multiset $\boldsymbol{P}_t + \overline{\boldsymbol{P}}_t$ according to the fitness values and copy the first $\overline{N} - |\overline{\boldsymbol{P}}_{t+1}|$ individuals $\boldsymbol{i}$ with $F(\boldsymbol{i}) \geq 1$ from the resulting ordered list to $\overline{\boldsymbol{P}}_{t+1}$. In the second case, when the size of the current nondominated (multi)set exceeds $\overline{N}$, an archive truncation procedure is invoked which iteratively removes individuals from $\overline{\boldsymbol{P}}_{t+1}$ until $|\overline{\boldsymbol{P}}_{t+1}| = \overline{N}$. Here, at each iteration that individual $\boldsymbol{i}$ is chosen for removal for which $\boldsymbol{i} \leq_d \boldsymbol{j}$ for all $\boldsymbol{j} \in \overline{\boldsymbol{P}}_{t+1}$ with

$$\boldsymbol{i} \leq_d \boldsymbol{j} \quad :\Leftrightarrow \quad \begin{aligned} &\forall\, 0 < k < |\overline{\boldsymbol{P}}_{t+1}| \;:\; \sigma_{\boldsymbol{i}}^k = \sigma_{\boldsymbol{j}}^k \quad \vee \\ &\exists\, 0 < k < |\overline{\boldsymbol{P}}_{t+1}| \;:\; \left[\left(\forall\, 0 < l < k \;:\; \sigma_{\boldsymbol{i}}^l = \sigma_{\boldsymbol{j}}^l\right) \wedge \; \sigma_{\boldsymbol{i}}^k < \sigma_{\boldsymbol{j}}^k\right] \end{aligned}$$

where $\sigma_{\boldsymbol{i}}^k$ denotes the distance of $\boldsymbol{i}$ to its k-th nearest neighbor in $\overline{\boldsymbol{P}}_{t+1}$. In other words, the individual which has the minimum distance to another individual is chosen at each stage; if there are several individuals with minimum distance the tie is broken by considering the second smallest distances and so forth. How this truncation technique works is illustrated in Figure 2.

Although, the worst run-time complexity of the truncation operator is $\mathcal{O}(M^3)$ $(M = N + \overline{N})^2$, on average the complexity will be lower ($\mathcal{O}(M^2 \log M)$) as individuals usually differ with regard to the second or third nearest neighbor, and thus the sorting of the distances governs the overall complexity.

4 Experimental Design

The behavior of SPEA2 is compared to SPEA, NSGA-II and PESA on a number of test functions. The algorithms are implemented according to their description in the literature. As the main feature under concern is the fitness assignment and the selection processes, our implementation only differ in these respects, where the other operators (recombination, mutation, sampling) remain identical. For each algorithm we used identical population and archive sizes.

---

[2]Constructing for each individual the list of distances to all other individual takes $\mathcal{O}(M^2)$, sorting all distance lists is of complexity $\mathcal{O}(M^2 \log M)$, choosing an individual for removal can be done in $\mathcal{O}(M^2)$ time, and updating the remaining distance lists after removal of an individual can be done in time $\mathcal{O}(M)$ (or even omitted with appropriate data structures).
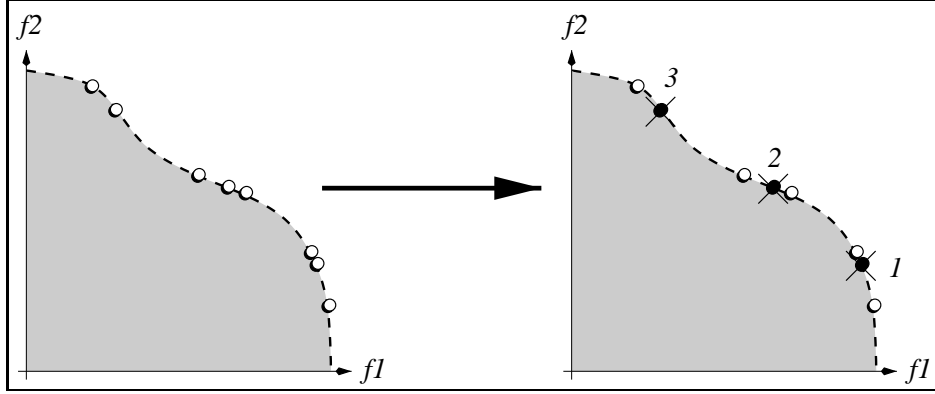
Figure 2: Illustration of the archive truncation method used in SPEA2. On the right, a nondominated set is shown. On the left, it is depicted which solutions are removed in which order by the truncate operator (assuming that $\overline{N} = 5$).

4.1 Test Problems and representation of solutions

The test functions are summarized in Tab. 1, where both combinatorial and continuous problems were chosen.

As combinatorial problems three instances of the knapsack problem were taken from (Zitzler and Thiele 1999), each with 750 items and 2, 3, and 4 objectives, respectively. For the random choice of the profit and weight values as well as the constraint handling technique we refer to the original study. The individuals are represented as bit strings, where each bit corresponds to one decision variable. Recombination of two individuals is performed by one-point crossover. Point mutations are used where each bit is flipped with a probability of 0.006, this value is taken using the guidelines derived in (Laumanns, Zitzler, and Thiele 2001). The population size and the archive size were set to 250 for $m = 2$, to 300 for $m = 3$, and to 400 for $m = 4$.

In the continuous test functions different problems difficulties arise, for a discussion we refer to (Veldhuizen 1999). Here, we enhanced the difficulty of each problem by taking 100 decision variables in each case. For the Sphere Model (SPH-m) and for Kursawe's function (KUR) we also chose large domains in order to test the algorithms' ability to locate the Pareto-optimal set in a large objective space. For all continuous problems, the individuals are coded as real vectors, where the SBX-20 operator is used for recombination and a polynomial distribution for mutation (Deb and Agrawal 1995). Furthermore, the population size and the archive size were set to 100.

The function SPH-m is a multi-objective generalization of the Sphere Model, a symmetric unimodal function where the isosurfaces are given by hyperspheres. The Sphere Model has been subject to intensive theoretical and empirical investigations with evolution strategies, especially in the context of self-adaptation. In a multi-objective environment a two-variable version of it was used for empirical evaluation of VEGA (Schaffer 1985), while in (Rudolph 1998) it was used for theoretical con-

9

n Type	Domain	Objective functions
SPH-m (Schaffer 1985; Laumanns, Rudolph, and Schwefel 2001)		
100 min	$[-10^3, 10^3]^n$	$f_j(x) = \sum_{1 \le i \le n, i \ne j}(x_i)^2 + (x_j - 1)^2$ $1 \le j \le m, \quad m = 2, 3$
ZDT6 (Zitzler, Deb, and Thiele 2000)		
100 min	$[0, 1]^n$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)\left[1 - (f_1(x)/g(x))^2\right]$ $g(x) = 1 + (n-1) \cdot ((\sum_{i=2}^m x_i)/(m-1))^{0.25}$
QV (Quagliarella and Vicini 1997)		
100 min	$[-5, 5]^n$	$f_1(x) = (\frac{1}{n}\sum_{i=1}^n(x_i^2 - 10\cos(2\pi x_i) + 10))^{\frac{1}{4}}$ $f_2(x) = (\frac{1}{n}\sum_{i=1}^n((x_i - 1.5)^2 - 10\cos(2\pi(x_i - 1.5)) + 10))^{\frac{1}{4}}$
KUR (Kursawe 1991)		
100 min	$[-10^3, 10^3]^n$	$f_1(x) = \sum_{i=1}^{n-1}(-10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}})$ $f_2(x) = \sum_{i=1}^n(|x_i|^{0.8} + \sin^3(x_i)$
KP-750-m (Zitzler and Thiele 1999)		
750 max	$\{0, 1\}^n$	$f_j(x) = \sum_{i=1}^n x_i \cdot p_{i,j}$ s.t. $g_j(x) = \sum_{i=1}^n x_i \cdot w_{i,j} \le W_j$ $p_{i,j}$ (profit values) and $w_{i,j}$ (weight values) randomly chosen $1 \le j \le m, \quad m = 2, 3, 4$

Table 1: Test problems used in this study. The objective functions are given by $f_j, 1 \le j \le m$, where m denotes the number of objectives and n the number of decision variables. The type of the objectives is given in the left column (minimization or maximization).

vergence analysis. Here, a two (SPH-2) and a three (SPH-3) objective instance are considered.

Zitzler, Deb, and Thiele's $\mathcal{T}_6$ (Zitzler, Deb, and Thiele 1999), here referred to as ZDT6, is also unimodal and has a non-uniformly distributed objective space, both or-

thogonal and lateral to the Pareto-optimal front. It has been proposed to test the algorithms' ability to find a good distribution of points even in this case.

The components of the function QV are two multi-modal functions of Rastrigin's type. Here, we employ the version of (Quagliarella and Vicini 1997), where the main difficulty besides the multi-modality is the extreme concave Pareto-optimal front, together with a diminishing density of solutions towards the extreme points.

Kursawe's function (Kursawe 1991) finally has a multi-modal function in one component and pair-wise interactions among the variables in the other component. The Pareto-optimal front is not connected and has an isolated point as well as concave and convex regions.

4.2   Performance assessment

For each algorithm and each problem, 30 runs with different random seeds have been carried out. For the quality or performance measure we apply a volume-based approach according to (Zitzler and Thiele 1999) with slight modifications. Here, a reference volume between the origin and an utopia point – defined by the profit sums of all items in each objective – is taken into account. The aim is to minimize the fraction of that space, which is not dominated by any of the final archive members. We consider this as the most appropriate scalar indicator since it combines both the distance of solutions (towards some utopian trade-off surface) and the spread of solutions. For each run, we measure the (normalized) size of the nondominated objective space over time, which leads to a sample of 30 values for each time step in each experiment.

5   Results and Discussion

5.1   Continuous Test Problems

The Multi-objective Sphere Model   The problem an algorithm faces on this function is to first locate the region of the Pareto-optimal set, which – dependent on the decision variable ranges – might form a tiny region in the search space. In proceeding to this region, a linear convergence order is desirable, as in the single-objective case. Near the Pareto-optimal front, the precision must be adjusted to approximate it properly, while in addition it becomes more and more difficult to proceed since the success rate decreases very quickly (see (Laumanns, Rudolph, and Schwefel 2001)).

The results are depicted in Fig. 3. In both cases SPEA fails to converge to the vicinity of the Pareto-optimal front. This is certainly due to the fitness assignment strategy (see Section 2.2): Far away from the Pareto-optimal front, almost all individuals form a chain in the objective space. The archive consists of only one individual, which dominates all others giving all of them the same fitness value and hence the same selection probability.

The plots suggest that the other three algorithms indeed exhibit linear order convergence in the first, quasi-single-objective phase. Here, PESA is faster than the other two candidates in locating the Pareto-optimal set in the first part of the runs. This occurs probably due to a higher elitism intensity, since in this stage the archive mostly
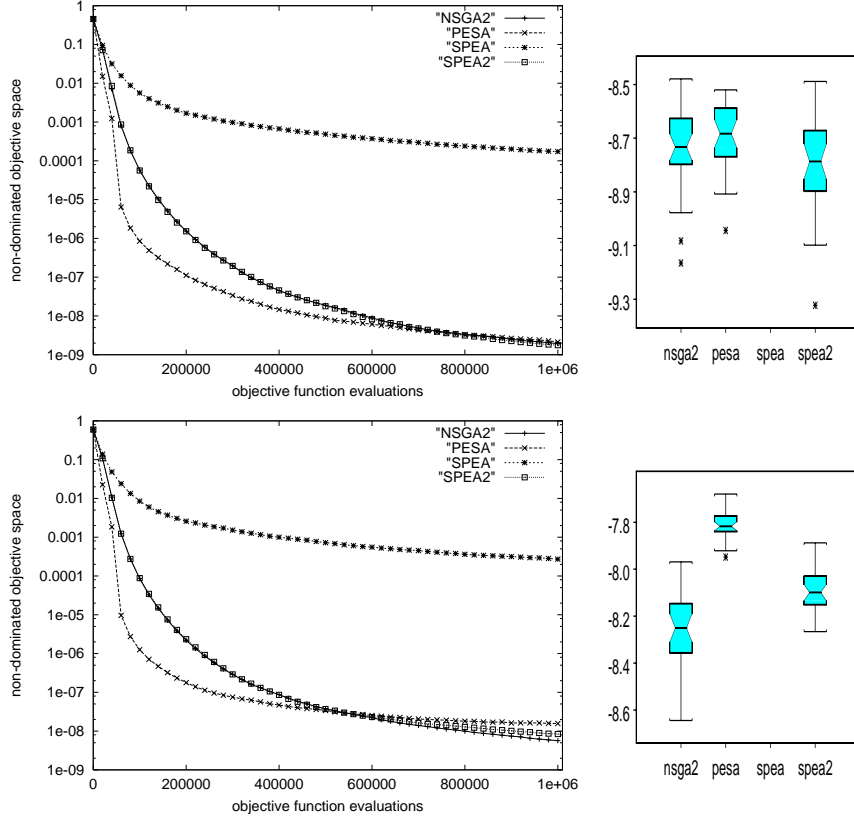
Figure 3: Performance values for SPH-2 (top) and SPH-3 (bottom). The graphs show the average values of 30 runs for each algorithm over time (left), and the box-plots represent the distributions of the 30 values at the end of the runs (right).

consists of only a few points. Close to the Pareto-optimal front, SPEA2 and NSGA-II overtake PESA with regard to the average performance measure. The boxplots of the final performance values show SPEA2 slightly ahead of NSGA-II for $m = 2$, while NSGA-II seems to achieve better values for $m = 3$. This might be due to NSGA-II's emphasizing the boundary solutions of the current nondominated set, which is likely to improve the performance indicator more. As to the visual impression, however, the distribution of solutions is more uniform with SPEA2, especially on the three objective version.

Zitzler, Deb, and Thiele's $\mathcal{T}_6$ On this function the performance indicator provides a very good measure, since the first component of the function is only governed by one variable, which all test candidates manage to cope with; the resulting fronts are mostly mutually non-overlapping.

From the plots in Fig. 4 a similar behavior as before on the Sphere Model can be
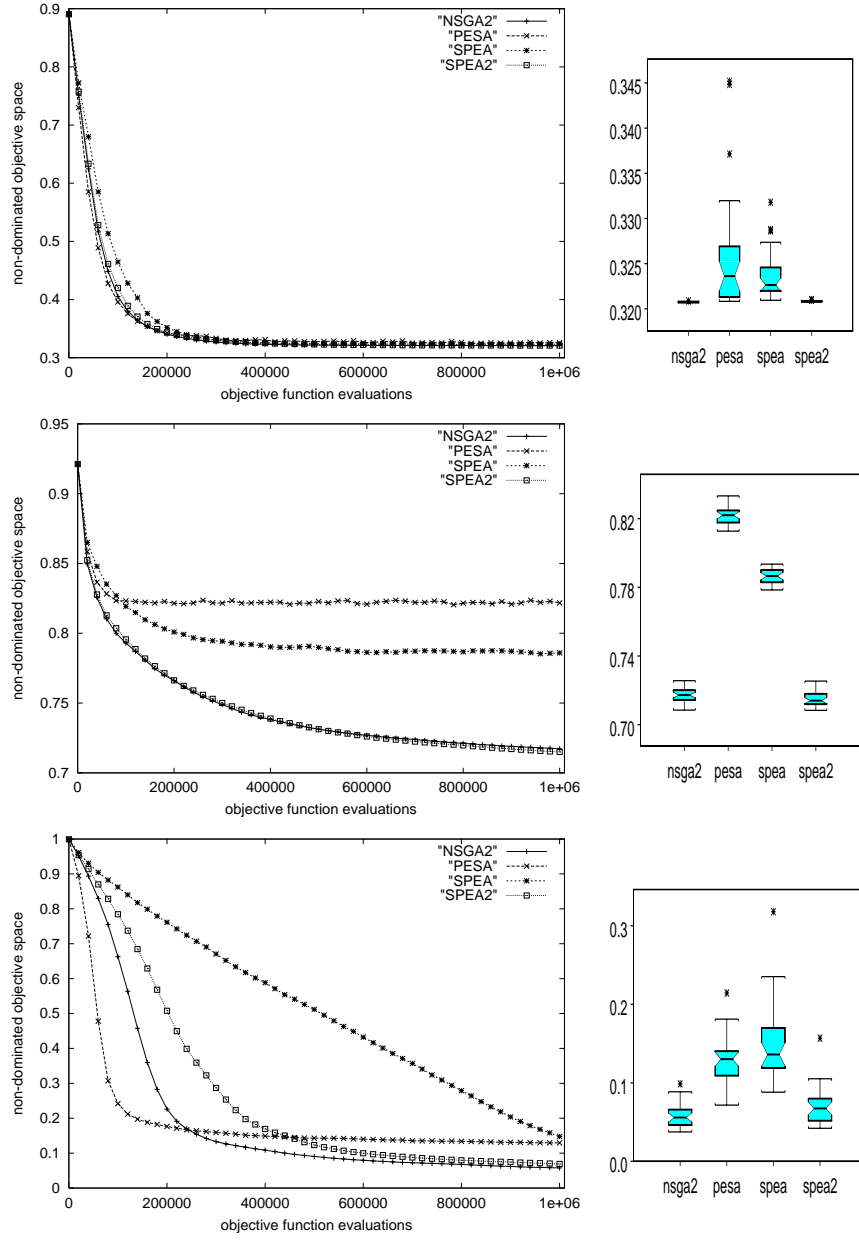
12

Figure 4: Performance values for ZDT6 (top), QV (middle) and KUR (bottom). The graphs show the average values of 30 runs for each algorithm over time (left), and the box-plots represent the distributions of the 30 values at the end of the runs (right).

13

traced: PESA converges quickly, but is overtaken by SPEA2 and NSGA-II. In comparison with SPEA2, NSGA-II appears to be slightly faster proceeding, but in the limit the results are of identical quality. In contrast, many of the PESA and SPEA runs have not converged to the Pareto-optimal set even after 1000000 function evaluations, the box plots show the corresponding spread in performance.

Multi-objective Rastrigin function On the test problem QV, Fig. 4 shows that with PESA – and later also for SPEA – the performance measure starts to stagnate very soon. This happens because of the extreme concavity of the Pareto set: The outer points of the nondominated set of PESA and SPEA are not guaranteed to stay in the archive, and on this type of problem, where also the density of solutions decreases at the extremes of the Pareto set, these points cannot be re-created as fast as they are lost. SPEA2 and NSGA-II, again, have almost identical performance.

Kursawe's function On Kursawe's function the differences in the average performance values are easily visible from Fig. 4. SPEA converges most slowly, probably due to the same effect as on the Sphere Model. However, the volume-based measure is not very instructive here: Even with a front, which is obviously farther behind another one and where almost all points are dominated by the other, a better value can be achieved. Unfortunately, the exact location of the true Pareto set cannot be determined, and thus other metrics relying on the true front are also not applicable.

The resulting fronts show that for all algorithms, not every run converges to the vicinity of the true Pareto set. In this sense SPEA2 has a success frequency of $18/30$ in contrast to PESA with $13/30$ and NSGA-II with $20/30$. The distribution of points of SPEA2 and NSGA-II appear equally good, which corresponds with NSGA-II having only an almost indiscriminative advantage of the performance value of the best run.

5.2 Combinatorial Test Problems

On the knapsack problems the results look structurally similar to the results on the Sphere Model. On the two objective version, SPEA2 produces on average the broadest distribution and hence the best performance values. On the three objective version, NSGA-II catches up, but – in contrast to the Sphere Model – does not reach the best performance value. Again, the distribution of solutions in higher objective space dimension becomes notably biased with NSGA-II, and in some runs remarkable patterns can be traced (see Fig. 6), which is probably due to the component-wise density estimation technique and the amplification of the extreme solutions.

As the box plots in Fig. 5 show, the performance differences increase with the number of objectives. With four objectives, a clear distinction between NSGA-II and SPEA2 on the one hand and PESA and SPEA on the other hand can be made: Both PESA and SPEA, which do not guarantee the extreme solutions to be kept in the archive, appear to stagnate without having reached a well spread distribution of solutions.

Moreover, pairwisely comparing the fronts produced by the different algorithms on the same initial populations suggests that SPEA2 has advantages over PESA and
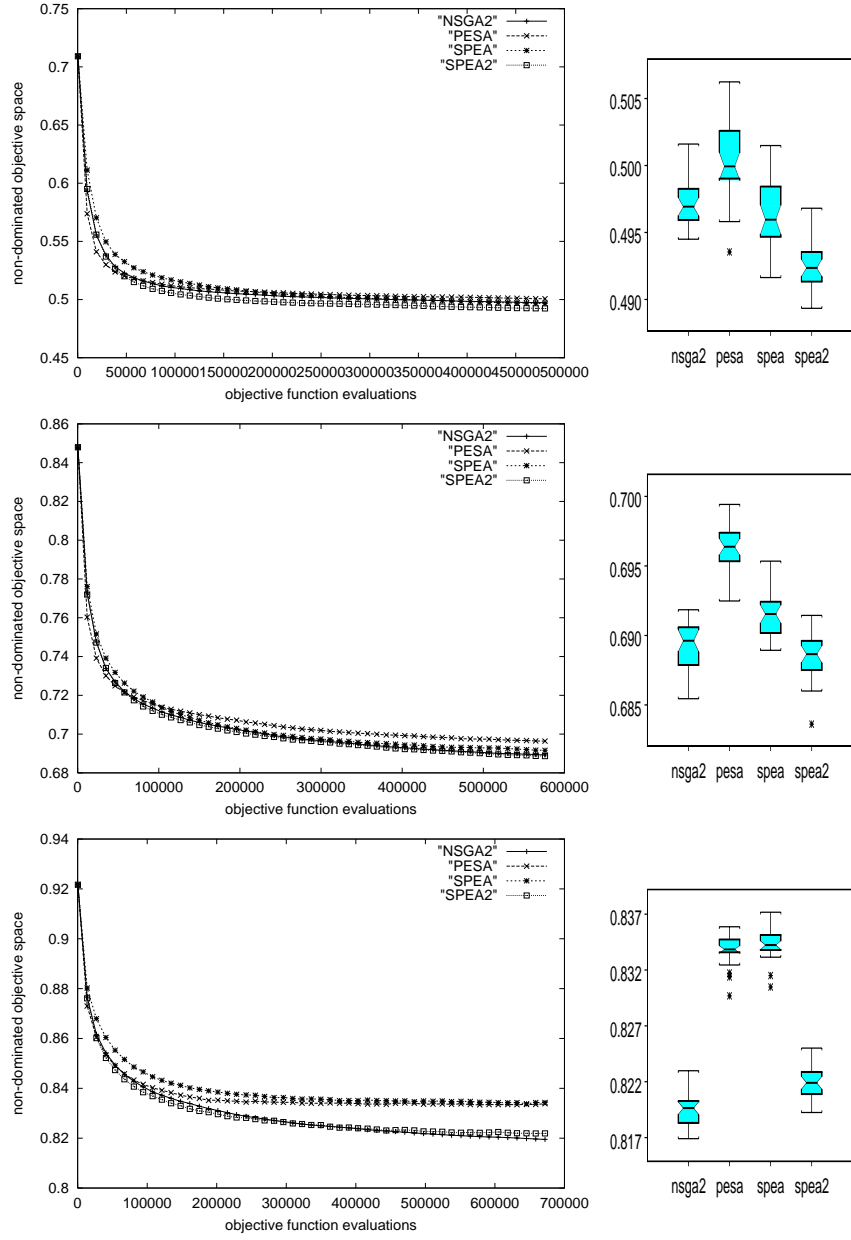
Figure 5: Performance values for KP-750-m with $m = 2$ (top), $m = 3$(middle) and $m = 4$ (bottom). The graphs show the average values of 30 runs for each algorithm over time (left) and the box-plots of the distribution of all 30 samples at the end of the run (right).
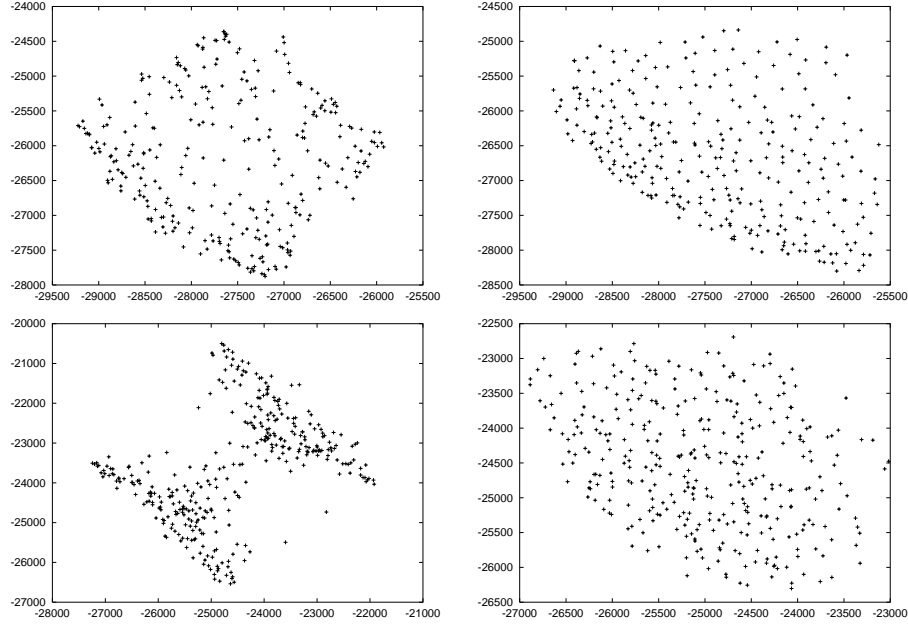
15

Figure 6: First two objectives of the final population of a single run of NSGA-II (left) and SPEA2 (right) on KP-750-3 (top) and KP-750-4 (bottom).

NSGA-II on higher dimensional objective spaces. On average, the nondominated set achieved by SPEA2 dominates about 80% of the nondominated solutions found by NSGA-II in the case of 3 and 4 objectives. Vice versa, the front obtained by NSGA-II dominates less than 2% of the nondominated solutions produced by SPEA2. With regard to PESA, a SPEA2 front dominates on average 33% and 54% of the corresponding PESA front with 3 and 4 objectives, respectively, while the nondominated set produced by PESA achieves only 22% and 7% dominance in comparison to SPEA2.

5.3 Summary

In all test cases, SPEA2 shows to constitute a significant improvement over its predecessor SPEA as it reaches better results on all considered problems. SPEA2 and NSGA-II seem to behave very similar on the different problems. In some cases NSGA-II reaches a broader spread and hence a better value of the performance measure, while SPEA2 provides a better distribution of points, especially when the number of objectives increases. PESA, however, tends to have difficulties to keep the outer solutions on certain test functions.

It is very instructive, however, to see how the performance develops over time, i.e. with the number of function evaluations. For many problems, PESA appears to be converging quicker at the beginning, which is probably due to its higher implicit elitism intensity. Both NSGA-II and SPEA2, which also allow dominated individuals

to maintain a minimum archive size, seem to make use of this increased diversity in the later stage of the run where they attain a broader distribution and hence better performance values.

6   Conclusions

In this report we have presented SPEA2, an improved elitist multi-objective evolutionary algorithm that employs an enhanced fitness assignment strategy compared to its predecessor SPEA as well as new techniques for archive truncation and density-based selection. Extensive numerical comparisons of SPEA2 with SPEA and with PESA and NSGA-II, two other recently proposed algorithms, have been carried out on various continuous and combinatorial test problems.

The key results of the comparison are:

- SPEA2 performs better that its predecessor SPEA on all problems.

- PESA has fastest convergence, probably due to its higher elitism intensity, but has difficulties on some problems because it does not always keep the boundary solutions.

- SPEA2 and NSGA-II show the best performance overall.

- In higher dimensional objective spaces, SPEA2 seems to have advantages over PESA and NSGA-II.

The comparative study emphasizes that many differences are only revealed for more objective than two. In higher dimensional objective spaces the number of non-dominated solutions increases rapidly. This presents a much greater challenge, e.g., for the archiving strategies and in general for the algorithms to keep the desired convergence properties and at the same time maintain a good distribution of solutions. Exactly this observation was the reason why great importance was attached to accurate density estimation with the design of SPEA2. Although PESA and NSGA-II are faster in terms of worst-case complexity, the issue of density estimation becomes the more important the more objectives are involved as the results on the knapsack problem suggest.

Furthermore, it became obvious that it is necessary to trace the performance over time to keep track of the dynamic behavior of the algorithms. Specifically, algorithms are likely to differ in convergence velocity or reveal effects such as premature convergence or stagnation, which cannot be seen from a static analysis after some arbitrary running time.

References

Bleuler, S., M. Brack, L. Thiele, and E. Zitzler (2001, May). Multiobjective genetic programming: Reducing bloat by using SPEA2. In *Congress on Evolutionary Computation (CEC-2001)*. To appear.

Corne, D. W., J. D. Knowles, and M. J. Oates (2000). The pareto envelope-based selection algorithm for multiobjective optimisation. In M. S. et al. (Ed.), *Parallel Problem Solving from Nature – PPSN VI*, Berlin, pp. 839–848. Springer.

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

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 M. S. et al. (Ed.), *Parallel Problem Solving from Nature – PPSN VI*, Berlin, pp. 849–858. Springer.

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

Horn, J., N. Nafpliotis, and D. E. Goldberg (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, Piscataway, NJ, pp. 82–87. IEEE Press.

Jaszkiewicz, A. (2000). On the performance of multiple objective genetic local search on the 0/1 knapsack problem. a comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Pozna University of Technology.

Knowles, J. D. and D. W. Corne (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Congress on Evolutionary Computation (CEC99)*, Volume 1, Piscataway, NJ, pp. 98–105. IEEE Press.

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

Lahanas, M., N. Milickovic, D. Baltas, and N. Zamboglou (2001). Application of multiobjective evolutionary algorithms for dose optimization problems in brachytherapy. In E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne (Eds.), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Volume 1993 of *Lecture Notes in Computer Science*, Berlin, pp. 574–587. Springer-Verlag.

Laumanns, M., G. Rudolph, and H.-P. Schwefel (2001, June). Mutation control and convergence in evolutionary multi-objective optimization. In *Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001), Brno, Czech Republic*.

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)*, Piscataway, NJ, pp. 46–53. IEEE.

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, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne (Eds.), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Volume 1993 of *Lecture Notes in Computer Science*, Berlin, pp. 181–196. Springer-Verlag.

Parks, G. T. and I. Miller (1998). Selective breeding in a multiobjective genetic algorithm. In A. E. E. et al. (Ed.), *Parallel Problem Solving from Nature – PPSN V*, Berlin, pp. 250–259. Springer.

Quagliarella, D. and A. Vicini (1997). Coupling genetic algorithms and gradient based optimization techniques. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter (Eds.), *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science — Recent advances and industrial applications*, pp. 289–309. Wiley, Chichester.

Rudolph, G. (1998). On a multi-objective evolutionary algorithm and its convergence to the pareto set. Technical Report No. CI-17/98, Department of Computer Science/XI, University of Dortmund.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.), *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, Pittsburgh, PA, pp. 93–100. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London: Chapman and Hall.

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

Tan, K. C., T. H. Lee, and E. F. Khor (2001). Incrementing multi-objective evolutionary algorithms: Performance studies and comparisons. In E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne (Eds.), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Volume 1993 of *Lecture Notes in Computer Science*, Berlin, pp. 111–125. Springer-Verlag.

Veldhuizen, D. A. V. (1999, June). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph. D. thesis, Graduate School of Engineering of the Air Force Institute of Technology, Air University.

Zitzler, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Aachen, Germany.

Zitzler, E., K. Deb, and L. Thiele (1999, December). Comparison of multiobjective evolutionary algorithms: Empirical results (revised version). Technical Report 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.

Zitzler, E., K. Deb, and L. Thiele (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation 8*(2), 173–195.

Zitzler, E. and L. Thiele (1998, May). An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.

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.

Errata

In Section 4.1, Table 1, the definitions of the test problems KUR and ZDT6 are misprinted. The functions functions used for this study are slightly different and defined as follows.

ZDT6

$$
\begin{aligned}
f_1(x) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\
f_2(x) &= g(x)\left[1 - (f_1(x)/g(x))^2\right] \\
g(x) &= 1 + 9 \cdot \left((\sum_{i=2}^{n} x_i)/(n-1)\right)^{0.25}
\end{aligned}
$$

KUR

$$
\begin{aligned}
f_1(x) &= \sum_{i=1}^{n}(|x_i|^{0.8} + 5 \cdot \sin^3(x_i) + 3.5828) \\
f_2(x) &= \sum_{i=1}^{n-1}(1 - e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}})
\end{aligned}
$$

We would like to thank Young-Hoon Kang for pointing out the inconsistency with the simulations results published on

```
http://www.tik.ee.ethz.ch/ zitzler/testdata.html.
```