A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism

Marco Laumanns **Eckart Zitzler** **Lothar Thiele**

Computer Engineering and Networks Laboratory
Swiss Federal Institute of Technology
Gloriastrasse 35
CH - 8092 Zurich
{laumanns,zitzler,thiele}@tik.ee.ethz.ch

Abstract- Though it has been claimed that elitism could improve evolutionary multi-objective search significantly, a thorough and extensive evaluation of its effects is still missing. Guidelines on how elitism could successfully be incorporated have not yet been developed. This paper presents a unified model of multi-objective evolutionary algorithms, in which arbitrary variation and selection operators can be combined as building blocks, including archiving and re-insertion strategies. The presented model enables most specific multi-objective (evolutionary) algorithm to be formulated as an instance of it, which will be demonstrated by simple examples. We will further show how elitism can be quantified by the model's parameters and how this allows an easy evaluation of the effect of elitism on different algorithms.

1 Introduction

The aim of this study is to provide a systematic approach to elitism in multi-objective evolutionary algorithms (MOEA). Multi-objective optimization can be seen as a special case of Multiple Criteria Decision Making (Steuer 1986). Corresponding algorithms serve as a tool of decision analysis. In this context it may be desirable to find or to approximate the Pareto-optimal solutions, i.e. the minimal elements of a partially ordered set (Rudolph 1998). The ordering relation is the decision maker's preference relation on the set of decision alternatives. There are, of course, some algorithmic tools devoted to related tasks of decision analysis, for instance goal programming (Deb 1998) or methods of a priori or interactive incorporation of preference information (Fonseca and Fleming 1997). These approaches, however, are not considered in this study. Instead, we refer to Horn (1997), Veldhuizen (1999), or Coello (1999) for a detailed survey on MCDM and multi-objective optimization in the field of evolutionary algorithms and for the problem specifications and definitions.

Recently several authors indicated that elitism could improve evolutionary multi-objective search significantly, e.g. (Parks and Miller 1998; Obayashi, Takahashi, and Takeguchi 1998; Zitzler, Deb, and Thiele 2000). These assumptions are naturally based on a very individual notion of 'elitism' as well as on problem-specific results with individual implementations and test problems. As a consequence, in the multi-objective case it has remained unclear up to now

- how to define 'elite' individuals or solutions,
- how to best incorporate the information from this elite effectively into the search, and
- which are the effects of elitism on evolutionary search for different algorithms and (classes of) multi-objective problems.

In this study we present a basis on which a thorough examination of the effects and possible benefits of elitism can be built. Hitherto, an appropriate model is proposed that helps both, the researcher in doing empirical and analytical evaluations, and the user in building customized algorithms.

We start by recalling the use of archiving and elitism in evolutionary algorithms in section 2 in order to transform the common-sense notion into a formal definition. The model presented in section 3 incorporates archiving and elitism in a general MOEA and includes a parameter by which the 'elitism intensity' can be scaled. In section 4 we show how some well-known MOEAs can be formulated as an instance of this unified model. Finally, section 5 shall demonstrate how the benefit of elitism can be traced empirically.

2 The Concept of Elitism

Elitism has always been a major topic of discussion in the evolutionary computation community. This mainly concerns cross-generational interaction. Though it is clear that all natural individuals have a limited lifespan, there is no general answer to how this concept should be applied to evolutionary algorithms. Interpretations are diverse: Canonical genetic algorithms, for instance, allow no parental survivability at all. In generation gap methods at least a fraction of the next generation is reserved for good parents, and in the $(\mu + \lambda)$ evolution strategies parents live as long as they are surpassed by better offspring.

The question of when to accept a newly generated solution, in spite of previously found ones, arises in all kinds of iterative melioration processes, e.g. simulated annealing, where the probability of accepting (temporary) worsening is controlled by a 'cooling' schedule.

2.1 Elitism in the presence of multiple objectives

As opposed to single-objective optimization, the question of elitism becomes even more complicated with MOEAs. Here, the set of decision alternatives is no longer totally ordered, and alternatives might be incomparable. This makes an 'elitist' decision difficult, so a number of different MOEA implementations emerge with elitistic features. These features may be classified along different criteria:

- Using a secondary 'elitist' population vs. implicit elitism through a 'plus' selection as in evolution strategies;
- The elitism strategy, or how the elitist population is updated;
- The evaluation strategy, or how the elite individuals affect the fitness assignment of the current generation and vice versa;
- The re-insertion strategy, or how elite individuals take part in the production of offspring;
- The control flow, or when archiving and re-insertion take place.

As this classification is not hierarchical, algorithms can arbitrarily combine different features.

Most algorithms under concern make use of a second population of elite individuals. The Strength Pareto EA (SPEA) by Zitzler and Thiele (1999), for example, stores all non-dominated solutions separately from the 'normal' population. Individuals are chosen from the population and the non-dominated set as recombination partners, while the latter get higher selection probabilities.

Parks and Miller (1998) also use an archive of non-dominated solutions, from which a random subset is re-inserted into the population of each generation. They found that the performance of the search raises with the percentage of the number of individuals been taken from the archive.

The Pareto Archived Evolution Strategy (PAES) by Knowles and Corne (1999) is a multi-objective $(1+1)$ evolution strategy that uses the dominance relation as the selection criterion. Here, elitism is guaranteed by the 'plus' selection, while the archive is used only as a comparison set for incomparable individuals.

The $(1 + 1)$-EA examined by Rudolph (1998) uses a selection criterion that changes randomly. But since only two individuals are compared, the dominated one will never be preferred. In his study, Rudolph uses this algorithm as an example to demonstrate theoretic results about convergence to the Pareto set. It is interesting to note that the algorithm for which the derived properties are valid is an elitist EA. Additional theory about different properties of selection schemes in MOEAs can be found in (Hanne 1999).

Certainly many more versions of elitist MOEAs exist[1], but these recent examples already show the great variety in the implementation of elitism. In the next section we construct a model to provide a general framework for elitism in multi-objective evolutionary algorithms. This model, which should be capable to simulate most MOEA implementations, is based on the following definition.

[1] See, e.g. (Horn 1997; Veldhuizen 1999).

2.2 Characterization of elitism

According to the authors' notion, elitism means that 'elite' individuals cannot be expelled from the active gene-pool of the population in favor of worse individuals. This concept can be formalized as follows:

Definition 1 (Elitism) *Let P^t denote the Population of a given evolutionary algorithm EA after $t \in \mathbb{N}$ iterations (generations). Let $p^t(x)$ denote the probability of individual $x \in P^t$ being selected as an operand for the variation operator in generation t. Then EA is said to be elitistic, if and only if for any preference relation $\prec$ given by a decision problem the following condition holds:*

$$\forall t \in \mathbb{N} : \exists x \in \mathcal{P}^{*t} : p^{t+1}(x) > 0 \qquad (1)$$

$$\begin{aligned} \textit{with} \quad \mathcal{P}^{*t} &= \{x \in \mathcal{P}^t \ : \ \nexists x' \in \mathcal{P}^t \ : \ x' \prec x\}, \\ \mathcal{P}^t &= \bigcup_{\tau \leq t} P^\tau \end{aligned}$$

In this definition $\mathcal{P}^t$ refers to the set of all individuals produced so far and $\mathcal{P}^{*t}$ to all non-dominated individuals out of $\mathcal{P}^t$.

Regardless whether an EA uses elitism or not, it is common practice in the single objective case to keep track of the best solution found during the run off-line. This solution can be seen as the best approximation of the (unknown) optimum in reach, given the information produced by the algorithm. The multi-objective analogue is to store all solutions that are not dominated by any other solution found during the run. This archive then represents the best approximation of the true Pareto set available. Thus, to supply an archive of non-dominated solutions is a sensible extension to a MOEA that helps to exploit the produced information. Of course, if the archive is only used as a store, the behavior of the EA won't change. Archives like this have been used by e.g. Kursawe (1991). Further examples are listed in (Veldhuizen 1999, pp. 3-26f) named "MOEA Secondary Populations". Here, the author also questions "how to best utilize this population" or - in other words - how to utilize the information provided by the additional solutions to guide the search process more effectively. This is one of the crucial topics our study wants to approach.

3 A Unified Model for (Elitist) MOEAs

Building generic models of EAs is a common practice to abstract from implementation-specific details and to provide a

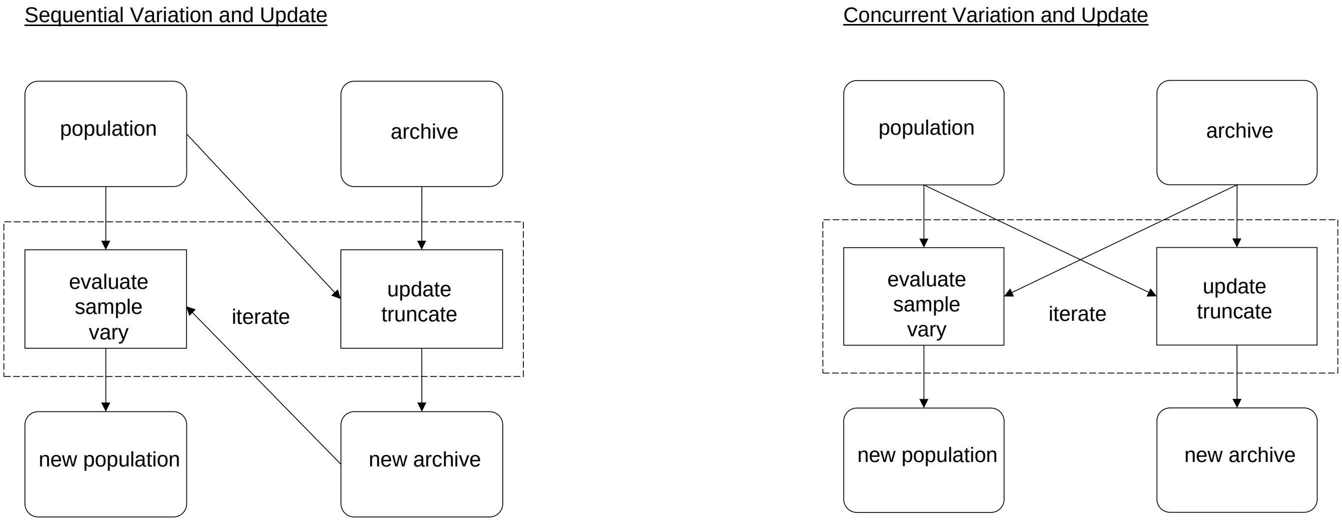

Figure 1: Scheme of a sequential (left) and a concurrent (right) transition function

general formulation. A universal EA is given in (Bäck, Hammel, and Schwefel 1997). The parameters of these models mainly consist of (stochastic) operators that represent the 'genetic' or 'evolutionary' operations.

Veldhuizen (1999) gives a general formulation of a MOEA that models the population size by a sequence instead of a single value to cope with different sizes of parent and offspring populations. However, the model does not allow for an implementation of an archive as an active part of the algorithm. In the following, we present a model which takes archiving into account.

3.1 Model description

In order to build a unified model for multi-objective evolutionary algorithms we combine features of standard EA models with those mentioned in section 2. This leads us to the following definition:

Definition 2 (UMMEA) *The Unified Model of Multi-objective Evolutionary Algorithms is defined by the individual space I and the probabilistic operators*[2]*:*

- $initialize : (\mathcal{I} \times \mathcal{I} \times [0,1])$
- $evaluate : (\mathcal{I} \times \mathcal{I} \times [0,1]) \mapsto \tilde{\mathcal{I}}$ *where*

$$\begin{aligned} evaluate(A, B, p_e) &= S_A \oplus S_B, \\ S_A &= \{(a, p_e \cdot p) : a \in A\}, \\ S_B &= \{(b, (1-p_e) \cdot p) : b \in B\}, \end{aligned}$$

$$p_e \in [0,1] \wedge \sum_{(a,p_e \cdot p) \in S_A} p = \sum_{(b,(1-p_e)\cdot p) \in S_B} p = 1$$

- $sample : \tilde{\mathcal{I}} \mapsto \mathcal{I}$
- $adapt : (\mathcal{I} \times \mathcal{I} \times [0,1]) \mapsto [0,1]$
- $vary : \mathcal{I} \mapsto \mathcal{I}$
- $update : (\mathcal{I} \times \mathcal{I}) \mapsto \mathcal{I}$
- $truncate : \mathcal{I} \mapsto \mathcal{I}$
- $iterate : (\mathcal{I} \times \mathcal{I}) \mapsto (\mathcal{I} \times \mathcal{I})$
- $terminate : (\mathcal{I} \times \mathcal{I} \times \mathbb{N}) \mapsto \{true, false\}$.

$\mathcal{I}$ denotes the set of all multi-sets with elements out of I and $\tilde{\mathcal{I}}$ the set of all multi-sets with elements out of (I × [0,1]). The symbol $\oplus$ denotes the union on multi-sets.

The main concept of this model is the coexistence of a 'normal' offspring population (B) and a generic archive of elite individuals (A). These two multi-sets of individuals and a further parameter (the 'elitism intensity' p_e) are initialized through the $initialize$ operator.

The offspring population is produced by using parents from both, the previous offspring population and the elite population. The involvement of elite individuals in this process is controlled by the elitism intensity p_e, which basically denotes the probability to choose a parent individual from the elite population. Dependent on this parameter the operator $evaluate$ assigns the target sampling rates to all individuals as a product of the sampling rate p within the individual set A (archive) or B (population), respectively, and the elitism intensity p_e. The resulting multi-set - an element of $\tilde{\mathcal{I}}$ - consists of pairs of individuals and the associated sampling rates. By implementing this $evaluate$ operator appropriately, it should be possible to simulate different fitness assignment strategies and selection schemes. The sampling procedure, however, is independent of the assignment of the sampling rates and can

[2]Due to the nature of evolutionary computation, most instances of this model will use probabilistic operators, though some might be fully deterministic as well.

$$
\begin{array}{l}
t := 0 \\
(A^0, B^0, p_e^0) := initialize() \\
\textbf{while } terminate(A^t, B^t, t) = false \textbf{ do} \\
\quad t := t + 1 \\
\quad A^t := truncate(update(A^{t-1}, B^{t-1}) \\
\quad p_e^t := adapt(A^t, B^{t-1}, p_e^{t-1}) \\
\quad B^t := vary(sample(evaluate(A^t, B^{t-1}, p_e^t))) \\
\textbf{end while}
\end{array}
$$

Figure 2: A general (elitist) MOEA. A^t denotes the archive, B^t the population and p_e^t the elitism intensity at generation t.

be expressed through the operator $sample$. The $vary$ operator capsules the recombination and mutation steps to create the new offspring population.

The elite population on the other hand represents a possibility to store arbitrary individuals which may be of interest for the decision maker. The elitism strategy, i. e. the decision which individuals to store, must be implemented in the $update$ operator. The $truncate$ operator then provides a means to cope with limited resources and to bound the size of the elite population, e. g. by clustering.

Finally, the interplay of these operators - or the timing - must be specified by a transition function via the operator $iterate$. Fig. 1 shows two different approaches, a concurrent ($iterate_c$) and a sequential ($iterate_s$) version. Though both look quite similar, there is a fundamental difference in respect to the use of elite individuals from the archive. In the sequential version the archive is updated before it takes part in the reproduction process. Thus, the archive always offers the current best individuals in contrast to the concurrent version, where new elite individuals have a delay of one iteration before they are available through the archive. This could mean better progress rates for the sequential version - especially for high elitism intensity - since information is exploited immediately. On the other hand, concurrency is better suited for parallel computing which can reduce the total execution time, even if more iterations are needed. On the contrary, the concurrent version guarantees that there are no multiple copies of individuals in both populations at the time of sampling. This version can also be regarded as a more 'conservative' one, because 'older' elite individuals receive a 'second chance' to reproduce, which may also have a positive effect on diversity.

For the sequential version the algorithm is outlined in Fig. 2. Analogously, an algorithm corresponding to $iterate_c$ may be formulated.

This framework is kept as general as possible to cover most of the different (elitist) algorithms proposed so far. The modular structure of the model allows for nearly arbitrary combinations of different operators to build a customized scalable elitist MOEA.

In the next section the methodology to decompose a given MOEA into its building blocks and re-formulating it as an instance of the unified model will be demonstrated.

4 Decomposing Existing Algorithms

The following examples show how existing algorithms can be decomposed and simulated using appropriate parameters for the unified model defined in the previous section.

4.1 General $(\mu \overset{+}{,} \lambda)$-ES

Evolution strategies with μ possible parents and λ offspring can be modeled without specifying the individual space I and the operators $initialize$, $sample$, and $vary$. The $update$ operator simply returns the μ best individuals from the union of the archive and the population ($A \oplus B$). The $evaluate$ operator sorts the individuals of the population according to their fitness values and assigns a sampling rate of $(1 - p_e) \cdot 1/\mu$ to the μ best individuals and zero to the others. The μ archive members are assigned $p_e \cdot 1/\mu$ as the target sampling rate.

The degree of elitism can be controlled through p_e. The extreme cases are $p_e = 1$, which yields the $(\mu + \lambda)$-ES, and $p_a = 0$ for the (μ , λ)-ES. The values in-between lead to some kind of mixed version. For the evolution strategies the sequential transition function $iterate_s$ must be applied.

4.2 SPEA

Also for SPEA the transition function $iterate_s$ governs the control flow. At first the archive is updated with $update$ returning all non-dominated individuals from $A \oplus B$ to the new archive. If the size of A exceeds a maximum value, $truncate$ clusters the archive using the average linkage method and then removes all but the centroid individual from each cluster. Global fitness values are assigned by $evaluate$ in respect to both A and B. SPEA's mating selection process is a binary tournament selection from the union of the (updated) archive and the population. The only chance for a population member to reproduce is to be chosen for a tournament against another population member, since archive members are guaranteed to have better fitness values than all individuals from the population. To simulate this, an adaptive and time-dependent p_e^t is needed and controlled by the $adapt$ operator. In each iteration the probability to sample an individual from the archive is given by

$$
p_e^t := adapt(A, B, p_e^{t-1}) := 1 - \left(\frac{|B|}{|A| + |B|}\right)^2
$$

4.3 Random insertion methods

By this headline we refer to algorithms that periodically insert (clones of randomly chosen) individuals from the archive back into the population without variation, e.g. (Ishibuchi and Murata 1996). This population constitutes the mating pool and thus corresponds to the output of the $sample$ operator in UMMEA. The elitism intensity of these algorithms is given by the fraction of the mating pool that is filled with archive members and can directly be transferred to our model. The $sample$ operator then has to make sure to take exactly the required number of individuals from the archive.

5 Preliminary Empirical Study on Elitism

In this section we study the effects of the elitism by comparing three simple instances of UMMEA. These example algorithms use common features of MOEAs such as selection based on the dominance relation (Pareto-based selection), discrete recombination of the decision variables and fixed mutation strength. An archive of non-dominated solution is supplied to all algorithms, where elitism is introduced by inserting archive members into the (mating) selection process. The performance measure, by which the outcomes of the UMMEA instances are compared, is based on the relative size of the search space the algorithm identifies as dominated. The test functions under consideration are constructed using the methodology proposed by Deb (1999).

5.1 Model instantiation and parameter settings

To describe the instances of UMMEA for the empirical study we first refer to their differences which are revealed in the operators $evaluate$, $update$, and $iterate$. The representation of the individuals as well as the $vary$ and $sample$ operators are kept constant for all three algorithms.

5.1.1 Evaluation and updating operators

For the evaluation operator $evaluate$ we use three different definitions. These will characterize the different algorithms that are to be compared in the following. However, the fitness assignment strategy will be the same for all, namely the dominance level in respect to the other individuals of the corresponding set (working population and archive, respectively, or the union of both). The sampling rates for the individuals of the population will be calculated based on the individuals' rank to simulate binary tournament selection ($t = 2$):

$$p(B, b) := \frac{t}{|B|} \prod_{i=1}^{t-1} (1 + \frac{1 - rank(b)}{|B| - i}) \qquad (2)$$

where b refers to the individual and B to the population. All archive members are assigned equal sampling rates. The sampling operator then samples the required number of individuals independently according to the sampling rates calculated as above. Minor differences are revealed by the $update$ operator and the transition function $iterate$ as described in the following.

Alg1 : In this algorithm, the archive is updated by taking all non-dominated solutions from $A \oplus B$, i.e.

$$\begin{aligned} update_1 &= \{x \in A \oplus B : \\ &\quad \nexists x' \in A \oplus B : x' \prec x\} \end{aligned}$$

The archive is not truncated ($truncate_1 = id$). The $evaluate_1$ operator ranks the individuals of the population according to their dominance level in the population. These ranks are used to calculate the sampling rates in the population according to equation 2.

The elitism intensity is controlled by the parameter p_e: $p_e = 0$ means that the archive is not actively used in the evolution process, while $p_e = 1$ means that only members of the archive are selected for recombination and mutation. This scheme is similar to the one investigated in (Parks and Miller 1998), however, no niching mechanism is incorporated here. The transition function is given by $iterate_s$.

Alg2 : This algorithm represents the tunable $(\mu \overset{+}{,} \lambda)$ evolution strategy described previously. In order to simulate the 'plus' selection through the archive A, the $update_2$ operator returns the best μ individuals from the union of the previous archive and the population concerning the dominance level, where $\mu = 1/10 \cdot |B|$. As in Alg1, $evaluate_2$ starts by calculating the ranks of the individuals in the population B based on their dominance level in the population only. Then, the best μ individuals of the population and the μ archive members are assigned the sampling rate of $(1-p_e) \cdot 1/\mu$ and $p_e \cdot 1/\mu$, respectively, while all other individuals have no chance to reproduce.

Alg3 : The third algorithm implements a scheme similar to the one proposed by Zitzler (1999) which is characterized by the concurrent transition function $iterate_c$. In the original version, the elitism intensity[3] varies with the size of the archive and does not allow for a systematic assessment of elitism. Instead, we use fixed p_e values here. The $evaluate_3$ assigns fitness values based on the individuals' dominance level in the union of archive and population. Then, archive and population are ranked independently and target sampling rates are assigned according to equation (2).

5.1.2 Test problems, representation, and variation operators

As it is known from the single objective case, elitist EAs have advantages on some unimodal problems. To verify whether this phenomenon carries over to the multi-objective case we first examine a type of function with no locally optimal sets that are not globally optimal. A second type of function then exhibits several locally optimal sets, where elitist algorithms are in danger of getting stuck to locally optimal solutions before reaching the globally optimal ones. To construct our test problems systematically, we apply the methodology proposed by Deb (1999). Deb suggests to use a generic two-objective problem

$$\begin{array}{llcl} \text{Min} & t(x) & = & (f_1(x_1), f_2(x)) \\ \text{s.t.} & f_2(x) & = & g(x_2, \ldots, x_n) \cdot h(f_1(x_1), g(x_2, \ldots, x_n)) \\ & x & = & (x_1, \ldots, x_n) \end{array}$$

[3]Though this concept is not applied in the referenced algorithm, an empirical elitism intensity could be determined a posteriori through the relative frequency of having sampled individuals from the archive.

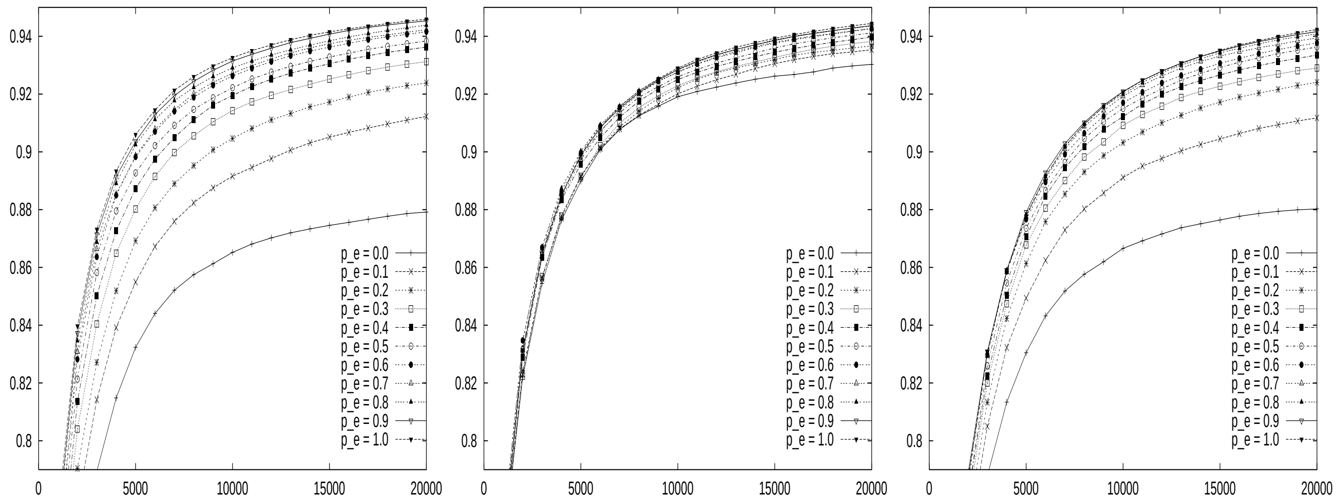

Figure 3: Size of the dominated space (measure $s(\cdot,\cdot)$) over time (number of function evaluations) for function t_1 produced by Alg1 (left) Alg2 (middle), and Alg3 (right). The elitism parameter p_e was varied from 0 to 1 in steps of 0.1.

where f_1 controls the search space along the Pareto-optimal front and g the search space lateral to the Pareto-optimal front. h specifies the shape of the Pareto-optimal front. Thus, like Zitzler (1999) we choose t_1 with

$$\begin{aligned} f_1(x) &= x_1 \\ g(x) &= 1 + \tfrac{9}{n-1} \cdot \textstyle\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned}$$

and $n = 100$, $x \in [0,1]^n$ as a representative for the first type and t_4

$$\begin{aligned} f_1(x) &= x_1 \\ g(x) &= 1 + 10(n-1) + \textstyle\sum_{i=2}^{m}(x_i^2 - 10\cos(4\pi x_i)) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned}$$

where $n = 50$, $x_1 \in [0,1]$ and $x_2, \dots, x_n \in [-5,5]$ for the second.

As the two considered functions are defined on a closed subset of $\mathbb{R}^n$, each decision variable is coded by a bit vector of length 32. Mutation is carried out by a bit mutation rate of 0.01. For all three algorithms the $initialize$ operator creates a random starting population according to the uniform distribution on the individual space. The archive is initialized as the empty set except for the Alg3, where the non-dominated individuals of the initial population are chosen since the concurrent transition function requires a non-empty archive for the first iteration.

The recombination operator repeatedly takes $r_1 = 2$ individuals from the set produced by the sampling operator and creates $r_2 = 1$ offspring until a population size $|B| = 100$ is reached. For each component defining one decision variable one of the parents' component is picked with equal probability (discrete recombination of decision variables). Thus, $sample$ has to provide $r_1/r_2 \cdot |B| = 200$ individuals from the archive and the population as recombination partners, while multiple copies are allowed.

5.1.3 Performance measure

For the performance of an algorithm on a specific problem we measure the (normalized) size of the dominated objective space similar to the definitions by Zitzler (1999) and Laumanns, Rudolph, and Schwefel (1999):

Definition 3 (Size of the dominated space) *Consider an arbitrary vector-valued objective function $F : \mathcal{X} \mapsto \mathcal{F}$, $\mathcal{X} \subseteq \mathbb{R}^n, \mathcal{F} \subseteq \mathbb{R}^m$. Let $\underline{f} = (\underline{f}_1, \underline{f}_2, \dots, \underline{f}_m)$ be the vector of the component minima of f and $\overline{f} = (\overline{f}_1, \overline{f}_2, \dots, \overline{f}_m)$ the vector of the component maxima. Then*

$$\begin{aligned} \mathcal{V}(F) := \{ \quad & f \in \mathbb{R}^m : \\ & f = \underline{f} + \textstyle\sum_{i=1}^{m} a_i((\underline{f}_1, \dots \overline{f}_i, \dots, \underline{f}_m) - \underline{f}), \\ & a_i \in [0,1] \quad \} \end{aligned} \tag{3}$$

For a multi-set of objective vectors A^* the dominated space will be

$$\mathcal{D}(A^*, F) := \bigcup_{f = F(x) \in A^*} \{ f' = F(x') \in \mathcal{V} : x \prec x' \} \tag{4}$$

and the size of the dominated space

$$s(A^*, F) := \frac{\lambda(\mathcal{D})}{\lambda(\mathcal{V})} \tag{5}$$

with $\lambda(A)$ as the Lebesgue measure of the bounded set A.

As suggested by Laumanns, Rudolph, and Schwefel (1999), the measure is performed over time, which allows for a better insight into the dynamic behavior of the algorithm. Since

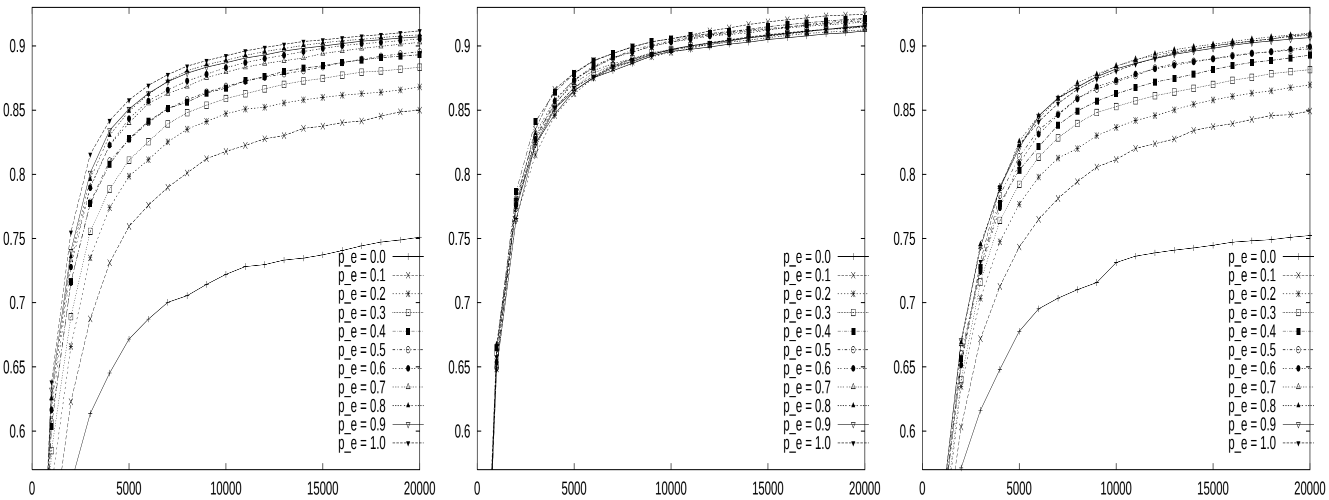

Figure 4: Size of the dominated space (measure $s(\cdot,\cdot)$) over time (number of function evaluations) for function t_4 produced by Alg1 (left) Alg2 (middle), and Alg3 (right). The elitism parameter p_e was varied from 0 to 1 in steps of 0.1.

the total objective space can be considerably greater than the region around the Pareto set (especially when the objective space scales with the number of decision variables) we visualize the results by zooming the regions of interest. Though the numerical difference between different $s(\cdot,\cdot)$ values can be very small, this does not necessarily mean that the differences between the corresponding solution sets are neglectably small.

5.2 Test results and discussion

These preliminary test results focus on the influence of the parameter p_e (the elitism intensity) on a simple and scalable elitist MOEA (Alg1) in comparison to other simplified versions of standard elitist MOEAs (Alg2 and Alg3). All three algorithms under concern have been implemented as an instance of UMMEA as described before. The values for p_e are varied from 0 to 1 in steps of 0.1. For the measure $s(\cdot,\cdot)$ all individuals produced during the run are considered for the set A^*. The median of a sample of s values of 20 independent runs is calculated and displayed in the diagrams.

On the simpler problem t_1 all three algorithms exhibit quite regular behavior. Not surprisingly, the performance improves with increasing p_e (Fig. 3) in all three cases. This suggests that in the multi-objective case elitism generally speeds up convergence to the next set of locally non-dominated solutions.

On t_4 this situation changes only for low search space dimensions. The best results are produced using some p_e value between the extremes 0 and 1. However, Fig. 4 shows for $n = 50$ that for high dimensional search spaces elitism is especially useful. Again, Alg1 and Alg3 obtain their best results for $p_e \to 1$.

As far as the qualitative behavior is concerned, the most obvious difference is the good performance of Alg2, even for low elitism intensity. This may be explained as follows: During most of the time the best μ individuals of the population are non-dominated among the population members. Thus we conclude, that the good results are mainly caused by the stronger selection pressure of the (μ, λ) selection.

On both functions Alg1 and Alg3 show very similar behavior. This is most apparent for $p_e = 0$, which suggests that the difference in the $evaluate$ operator to calculate the dominance level is of no effect at all for the problems considered here. The somewhat faster ascending curves for Alg1 lead us to the assumption that the sequential transition function ($iterate_s$) improves convergence on these problems, as we have supposed in Section 3. For other problems, where interfrontal diversity of individuals is more important to proceed, $iterate_c$ may be advantageous though.

6 Conclusion and Outlook

The presented model is intended to form a basis on which researchers can evaluate the effects of different operators and parameter settings on multi-objective evolutionary search. It also provides a more systematic approach to build new, customized MOEAs for possible users.

Our results underline the assumption that elitism is especially beneficial in the presence of multiple objectives and that the use of elitism speeds up convergence to the Pareto set. In this context the actual variant how elitism is incorporated in the algorithms seems to be of less importance as to the similar behavior of the three algorithms examined here. The danger of premature convergence caused by elitism is less imminent for the multi-objective problems of this study, because the elite set already contains a number of diverse solutions.

Unfortunately, general guidelines for the appropriate elitism intensity are hard to derive. However, we presented a methodology to approach this problem systematically. For the test cases considered here, the best results were produced using the highest elitism intensity. In how far these results carry over to other problems is subject to further investigation. Additional studies should also focus on the interaction of selection pressure and elitism intensity. Furthermore, the effects of elitism on the diversity of individuals need to be dealt with, since diversity keeping methods, either through fitness assignment (e. g. fitness sharing) or through clustering of the archive (by the truncation operator) have not yet been addressed.

Generally, the parameterization of the different operators is a major topic of in the field of evolutionary computation. In this case future research could try to develop robust self-adaptive mechanisms also for the elitism intensity to avoid choosing a priori values that are only based on a limited number of experiments or on intuition.

Bibliography

Bäck, T., U. Hammel, and H.-P. Schwefel (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation 1*(1), 3–17.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization. *Knowledge and Information Systems 1*(3), 269–308.

Deb, K. (1998). Non-linear goal programming using multi-objective genetic algorithms. Technical Report No. CI-60/98, Department of Computer Science/XI, University of Dortmund, Germany.

Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation 7*(3), 205–230.

Fonseca, C. M. and P. J. Fleming (1997). Multiobjective optimization. In T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, Bristol, UK, pp. C4.5:1–9. IOP Publishing and Oxford University Press.

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

Horn, J. (1997). Multicriterion decision making. In T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, Bristol, UK, pp. F1.9:1–15. IOP Publishing and Oxford Univertity Press.

Ishibuchi, H. and T. Murata (1996). Multi-objective genetic local search algorithm. In *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, Piscataway, NJ, pp. 119–124. IEEE.

Knowles, J. and D. 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.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In H.-P. Schwefel and R. Männer (Eds.), *Parallel Problem Solving from Nature — Proc. 1st Workshop PPSN*, Berlin, pp. 193–197. Springer.

Laumanns, M., G. Rudolph, and H.-P. Schwefel (1999). Approximating the pareto set: Concepts, diversity issues, and performance assessment. Technical Report No. CI-72/99, Department of Computer Science/XI, University of Dortmund, Germany.

Obayashi, S., S. Takahashi, and Y. Takeguchi (1998). Niching and elitist models for mogas. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel (Eds.), *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Berlin, Germany, pp. 260–269. Springer.

Parks, G. T. and I. Miller (1998). Selective breeding in a multiobjective genetic algorithm. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel (Eds.), *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Berlin, Germany, pp. 250–259. Springer.

Rudolph, G. (1998). Evolutionary search for minimal elements in partially ordered sets. In *Evolutionary Programming VII – Proc. Seventh Annual Conf. on Evolutionary Programming (EP-98)*, San Diego CA. The MIT Press, Cambridge MA.

Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application*. New York: Wiley.

Veldhuizen, D. A. V. (1999). *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, Germany, ISBN 3-8265-6831-1.

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