Multidimensional Optimization with a Fuzzy Genetic Algorithm

S. Voget[1], Robert Bosch GmbH, Frankfurt
and
M. Kolonko[2], Institut für Mathematik, TU Clausthal

Abstract: We present a new heuristic method to approximate the set of Pareto-optimal solutions in multi-criteria optimization problems. We use genetic algorithms with an adaptive selection mechanism. The direction of the selection pressure is adapted to the actual state of the population and forces it to explore a broad range of so far undominated solutions. The adaptation is done by a fuzzy rule based control of the selection procedure and the fitness function. As an application we present a timetable optimization problem where we used this method to derive cost-benefit curves for the investment into railway nets. These results show that our fuzzy adaptive approach avoids most of the empirical shortcomings of other multiobjective genetic algorithms.

KEYWORDS: OPTIMIZATION WITH MULTIPLE CRITERIA, GENETIC ALGORITHMS, ADAPTIVE SELECTION PROCEDURE, PARETO-OPTIMAL SOLUTIONS, COST-BENEFIT ANALYSIS

[1] Dr. Stefan Voget, Robert Bosch GmbH, Abt FV/SLD, Kleyerstr. 94, D-60326 Frankfurt, Germany, e-mail Stefan.Voget@fr.bosch.de

[2] Prof. Dr. Michael Kolonko, Institut für Mathematik, TU Clausthal, Erzstr. 1, D-38670 Clausthal-Zellerfeld, Germany, e-mail kolonko@math.tu-clausthal.de

1 Introduction

This paper brings together multidimensional optimization, genetic algorithms and a fuzzy control of the genetic algorithm. We start by describing the multidimensional optimization problem.

We consider the cost function $f(x) = (f_1(x), \ldots, f_K(x))$ where the solution x varies over a set S and $f_i : S \to \mathrm{I\!R}_+,\ 1 \leq i \leq K$, are the K components of the cost function. The aim is to find solutions x which are 'good' with respect to all $f_1, \ldots, f_K$ simultaneously. In a strict sense, such solutions will exist in rare cases only. Instead, we search for undominated, so-called Pareto-optimal solutions. We say $x \in S$ *dominates* $x' \in S$ $(x \preceq x')$ if $f_i(x) \leq f_i(x')$ for all $i = 1, \ldots, K$ and $f_j(x) < f_j(x')$ for at least one $1 \leq j \leq K$. Let P be the set of undominated or *Pareto-optimal* solutions (see e.g. [18]), i.e.

$$P := \{\, x \in S \mid \neg \exists x' \in S \;\; x' \preceq x \,\}.$$

A Pareto-optimal solution cannot be improved with respect to any cost function f_i without increasing the value of some other f_j, see Fig. 1. We use the term 'cost function' for the components f_i as well as for the $K-$dimensional $f = (f_1, \ldots, f_K)$

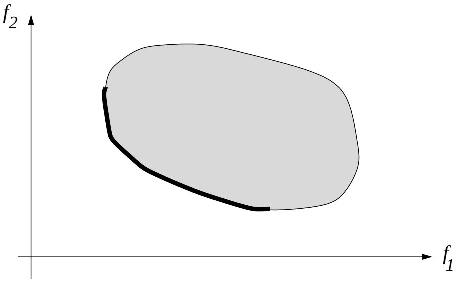


Figure 1: For the shaded area of possible cost function values, the undominated points representing the Pareto-optimal solutions are on the bold curve.

In the classical literature on multidimensional optimization problems, different approaches are used. Most of them avoid the direct search for Pareto-optimal solutions. Often an artificial one-dimensional cost function $\hat{f}$ is used, e.g.

$$\hat{f}(x) := \max_{1 \leq k \leq K} f_k(x)$$

in the so-called *min-max* approach or

$$\hat{f}(x) := \sum_{k=1}^{K} \alpha_k f_k(x)$$

in the *Bayesian* approach where one has to fix coefficients $\alpha_1, \ldots, \alpha_K$. Then the best solution with respect to $\hat{f}$ is searched for. Multidimensionality is preserved in the *lexicographic* approach where the cost functions $f_1, ..., f_K$ are assumed to be ordered with decreasing importance. Then a solution x with lexicographically smallest cost vector $(f_1(x), ..., f_K(x))$ is searched for. These approaches all have in common that they have to make additional assumptions e.g. about the relative importance of the $f_1(x), ..., f_K(x)$.

Finding the set P (or an approximation to it as in our approach) is a much more complex task. However, it gives greater freedom to the decision maker when selecting solutions. He can compare different solutions even with respect to additional criteria which have not been formalized (e.g. acceptance by the management or the public). If for example, for $K = 2$, f_1 denotes the money invested into some project and f_2 the benefit of this investment then P represents a cost-benefit analysis useful for a general understanding of the problem, see Section 7 for an application. Visualising P (at least in the two- or three-dimensional case) yields a curve or surface that can give some new insight into the structure of the problem, e.g. about the hardness or sensitivity of the different criteria. Note that min-max- (Bayes- or lexicographically) optimal solutions are Pareto-optimal solutions themselves, i.e. the optimization with respect to these criteria can be restricted to the set P.

Genetic Algorithms (GA) (see e.g. [14] for an overview) provide a general optimization framework which is particularly suited for treating multidimensional optimization problems and approximating Pareto-optimal solutions (see e.g. [5, 9, 10, 13, 17]). Generally, GA use a whole set G_n (a *population*) of solutions from S at each step (*generation*) n. The population G_n evolves by producing 'offspring' solutions from its elements using nature-like operations like 'crossover', 'mutation' etc. We call this the *production step* (cp. [1]). In the *reduction step*, the enlarged population G_n^+ of parents and offspring is reduced to its original size forming the new population G_{n+1}. This is done by selecting solutions (*individuals*) x from G_n^+ according to a real-valued function $g(x)$ called *fitness*. E.g. in an 'elitist' reduction, only the N solutions with the best fitness values are selected for G_{n+1}, where N is the size of G_{n+1} (*population size*). For more details on the elements of general GAs see [14].

If the fitness $g(x)$ is related to the cost functions $f_1, \ldots, f_K$ then in the long run, individuals with low costs will emerge from the process. (Note that we minimize the fitness as we are dealing with cost functions.) The sequence of populations G_n explores the solution space S with an emphasis on individuals with (in our case) low fitness values. In the one-dimensional case one would use $g(x) := f_1(x)$.

Different approaches for dealing with multidimensional optimization tasks have been considered in the GA-literature. The undominated solutions found up to the n-th generation are taken as an approximation to the set P of Pareto optimal solutions. In [13], a particular crossover operation is introduced that takes into account the multidimensionality of the costfunction. In [10], the way candidates are selected for crossover depends on all the cost functions$f_1, \ldots, f_K$ and in [4, 5, 17], the cost functions are used for the reduction (selection of the next generation), similar to our approach. The three most successful algorithms seem to be the MOGA (multi-objective genetic algorithm) of [4, 5] the algorithm of [10] and VEGA (vector evaluated genetic algorithm) from [17]. More details on these approaches are given in Section 4 below.

It has been noted before that the performance of the GA can be improved if the algorithm is adapted to the random evolution of the population and that this can be done best by a fuzzy control mechanism, see [8] for an overview and a bibliography on these approaches. For one-dimensional problems the goal of the adaptation is mainly to prevent 'premature convergence' of the population. Based on some measure of diversity for the population the parameters and/or operators of the algorithm are changed if necessary, e.g. the mutation rate can be increased if the population tends to converge at an early stage of the optimization.

We extend this approach to multidimensional optimization problems. Here the goal is not only to prevent convergence but to enable a uniform approximation of the Pareto optimal solutions. We use a fuzzy rule based system to select the appropriate type of reduction and fitness function for each generation. In our present set-up the production step (i.e. crossover and mutation) is not changed during optimization. It is supposed to be done in a standard fashion depending on the problem and is not considered in any detail here.

We start by explicitely defining the desirable 'goals' for the evolution of the population in Section 2. Here we formulate our belief of a 'good' approximation of the Pareto-optimal solutions in terms of desirable geometric properties of the cloud of cost values F_n where

$$F_n := \{ \, (f_1(x), \ldots, f_K(x)) \mid x \in G_n \, \}.$$

Similarly F_n^+ denotes the cost function values for the extended population G_n^+. Next we need to measure the deviation of a population from the goals. We do this in Section 3 by introducing suitable statistics for F_n. These values form the 'state' Γ_n of a population G_n. In Section 4, we fix a set of possible actions which could serve as 'countermeasures' to decrease the deviation as measured in Γ_n. These actions are different reduction methods, based on the classical ones mentioned above. Finally we need to decide which action should actually be taken, given the state of the present population. As we can formulate our goals in vague terms only, a fuzzy controller seems to be appropriate, it is introduced

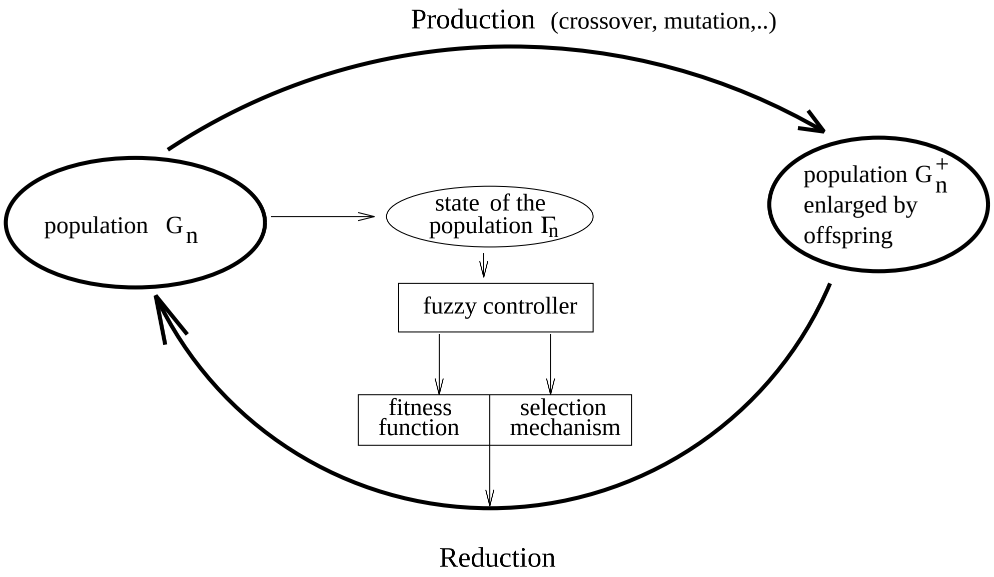

Production (crossover, mutation,..)
population G_n
state of the population Γ_n
population G_n^+ enlarged by offspring
fuzzy controller
fitness function
selection mechanism
Reduction

Figure 2: The production - reduction cycle of a genetic algorithm uses a selection mechanism and a fitness function that are adapted to the state of the population by a fuzzy controller.

in Section 5. Fig. 2 shows an overview of our approach which is summarized in detail in Section 6. In the final Section 7 we present some empirical results made with this method. We applied it to the optimization of timetables for railway networks and obtained good cost-benefit diagrams for the investment into railway networks. Though at present, we only have representative results with $K = 2$ cost functions, we formulate our model in the following Section for general K.

2 Defining Desirable Goals

In order to make our concept transparent, we are explicitely stating the main goals of the adaptation in advance. If one wants to define what a 'desirable' behaviour of the cost value cloud F_n is, in order to approximate the Pareto-optimal points, one needs overall knowledge of the cost landscape which is not available. Instead, we take a random sample $x_1, \dots, x_N$ of solutions and examine their cost function values. We take the statistical properties of this sample as an estimate of basic properties of the problem.

In the context of genetic algorithms, a suitably chosen initial population G_0 provides a sample of the solution space. We therefore compare the statistics of G_0 (or rather F_0) with those of G_n and collect the results in the state Γ_n. Γ_n later on serves as input for the fuzzy controller. As these values have to be computed very often, we can take only very rough statistics of F_n. In fact we replace F_n by its enclosing rectangle. Therefore we only have to calculate the following basic data for each population G_n, $n \geq 0$:

$$\begin{aligned} \lambda_i(n) &:= \min_{x \in G_n} f_i(x) \\ \sigma_i(n) &:= \max_{x \in G_n} f_i(x) - \min_{x \in G_n} f_i(x) \\ \mu_i(n) &:= \lambda_i(n) + \frac{\sigma_i(n)}{2} \end{aligned} \tag{1}$$

for $i = 1, \dots, K$, see Figure 3. Note that most figures are for the case $K = 2$ only.

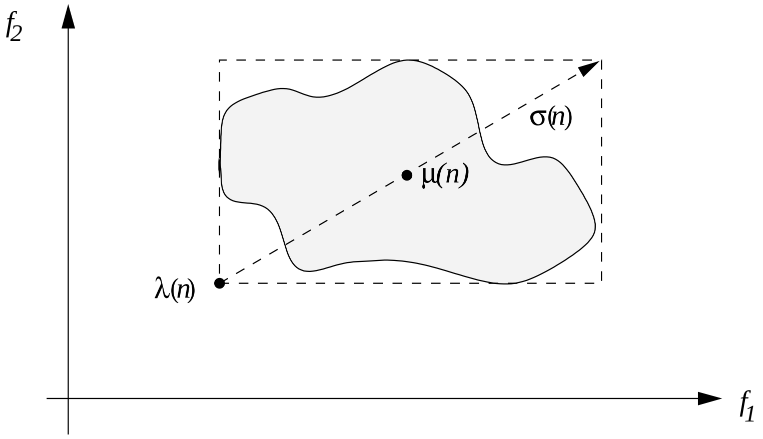


Figure 3: The main data of the population G_n.

Then $\lambda(n) := (\lambda_1(n), \dots, \lambda_K(n))$ describes the lower left corner of the enclosing rectangle of F_n. $\mu(n) := (\mu_1(n), \dots, \mu_K(n))$ is the center of the rectangle and $\sigma(n) := (\sigma_1(n), \dots, \sigma_K(n))$ its diagonal. $\sigma_1(n), \dots, \sigma_K(n)$ also represent the side lengths of the enclosing rectangle. $\lambda(n)$ and $\sigma(n)$ completely determine the enclosing rectangle. Note that $\mu(n)$ and $\sigma(n)$ are fast (and crude) replacements for the mean and variance of F_n.

Our overall goal is to push F_n towards the origin $0 = (0, \ldots, 0)$ in $\mathbb{R}_+^K$ with a broad but connected 'lower left' front (as indicated in Fig. 4). If lower bounds $L_i > 0$ for f_i $(1 \leq i \leq K)$ are known then the origin 0 should be replaced by the point $(L_1, \ldots, L_K)$ in the sequel. Similarly, instead of the positive quadrant $\mathbb{R}_+^K = [0, \infty)^K$ we would then consider $[L_1, \infty) \times \cdots \times [L_K, \infty)$ as our image space of f.

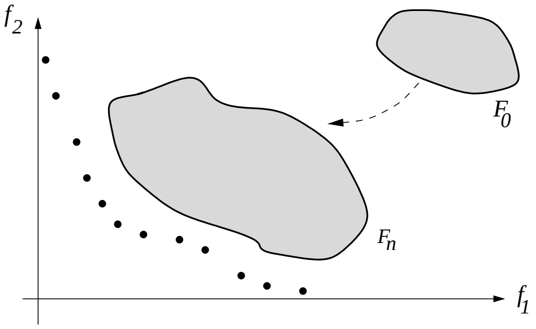


Figure 4: The ideal evolution of the cost function values. The dots mark the (unknown) Pareto optimal solutions.

From this overall goal we may derive the following more detailed goals which are discussed below.

Goal 1: $\mu(n)$ should stay on the line $\overline{0, \mu(0)}$ connecting 0 and $\mu(0)$.

Goal 2: $\sigma(n)$ should not become smaller than $\sigma(0)$.

Goal 3: F_n should form a connected cloud, i.e. G_n should not be split into separate subpopulations.

Goal 4: The lower left corner $\lambda(n)$ of the enclosing rectangle of F_n should move along $\overline{0, \lambda(0)}$.

In Figure 5 the goals are illustrated. In Figure 5 (a) the center $\mu(n)$ of population G_n deviates too much from the line $\overline{0, \mu(0)}$ whereas $\mu(m)$ is quite close. Goal 1 reflects our experience that it is particularly difficult to obtain good solutions in the 'center', i.e. with all $f_i(x)$, $1 \leq i \leq K$, small. Goal 2 avoids loosing the variation within a population which easily occurs if the selection is too strict, cp. population G_n in Figure 5 (b). Goal 3 is to avoid situations as in Figure 5 (c), which typically appears in the system VEGA, see also Figure 14 below. Goal 4 enforces Goal 1 in that it requires not only the center $\mu(n)$ but also the lower left corner of the enclosing rectangle to move along the line $\overline{0, \lambda(0)}$. This controls the shape of F_n to some extent, it should not differ to much from the shape of F_0, cp. F_n and F_m in Figure 5 (d).

Note that one could easily add other goals in a similar spirit. The above collection turned out to be quite successful in our experiments.

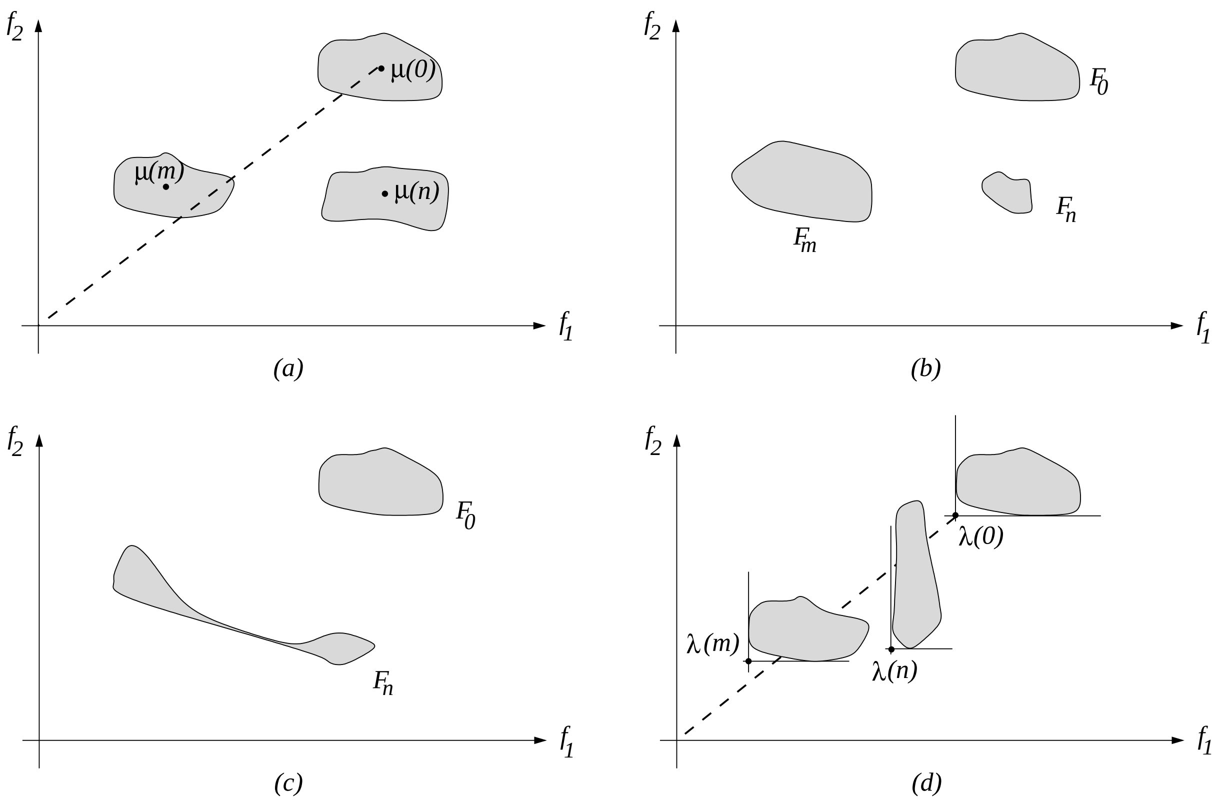


Figure 5: Illustration of the goals 1 - 4. The population in generation n is deviating from the goals, whereas in generation m it is as desired.

These goals take the initial population as a guide line. Hence the algorithm itself will depend on the initial population. Our experiments indicate however that the performance is rather stable as long as the initial population is created randomly, i.e. without applying heuristics for local optimization. This corresponds to the view of the initial population as a representative sample of the solution space.

In the next section we shall introduce measures for the deviation of the population from these goals. These measures will yield the state Γ_n of the population, a 'crisp' state in the fuzzy terminology which is later on transformed into a fuzzy state.

3 The State of the Population

The state Γ_n of population G_n contains the deviations $\Delta_i = \Delta_i(G_n)$ from the ith goal and, if appropriate, additional information like the direction of the deviation $\Theta_i = \Theta_i(G_n)$. We choose the measures Δ_i such that they take on values between 0 and 1, 1 indicating maximal deviation.

3.1 Deviation of the center $\mu(n)$

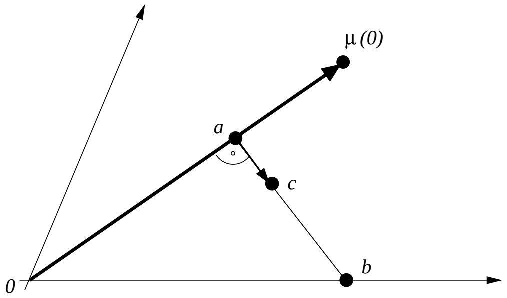


Figure 6: The distance of a point c from $\overline{0,\mu(0)}$ is calculated in the plane determined by c, $\mu(0)$ and 0. Note that the intersections of this plane with the coordinate planes of $\mathrm{I\!R}_+^K$ need not be orthogonal to each other.

To measure the deviation of a point c from the line $\overline{0,\mu(0)}$, we use a normed distance as indicated in Figure 6 : let a be the point on $\overline{0,\mu(0)}$ nearest to c and let b be the point where the line from c through a hits one of the coordinate planes. Put

$$\delta(c) := \frac{c-a}{b-a} \tag{2}$$

where $y^2 = \sum_{k=1}^K y_k^2$ is the Euclidean norm of a vector $y = (y_1, \ldots, y_K)$. $\delta(c)$ is the perpendicular distance $c-a$ of c from the line $\overline{0,\mu(0)}$ normed by the maximal distance $b-a$ a point can have from $\overline{0,\mu(0)}$ (in that direction). The next Lemma shows how to calculate $\delta(\cdot)$ efficiently.

Lemma 1

With the notation as in Figure 6 we have for all $c = (c_1, \ldots, c_n) \in \mathrm{I\!R}_+^K$

a) $0 \le \delta(c) \le 1$

b) $\delta(c) = 1 - \frac{\sum_{k=1}^K \mu_k(0)^2}{\sum_{k=1}^K \mu_k(0) c_k} \min_{1\le k\le K} \frac{c_k}{\mu_k(0)}.$

Proof : (a) follows from the definition. For (b) note that a is the projection of c onto $\mu(0)$, taken as vectors in $\mathbb{R}^K$ (see Fig. 6), hence we obtain from elementary linear algebra that for

$$\alpha := \frac{\langle \mu(0), c \rangle}{\mu(0)^2} = \frac{\sum_{k=1}^{K} \mu_k(0) c_k}{\sum_{k=1}^{K} \mu_k(0)^2}$$

we have

$$a = \alpha \cdot \mu(0) \tag{3}$$

i.e. for $a = (a_1, \ldots, a_K)$

$$a_k = \alpha \cdot \mu_k(0), \qquad \text{for } k = 1, \ldots, K. \tag{4}$$

We want to find the point b as indicated in Fig. 6. To do so we travel from a past the point c until we hit a coordinate plane. Mathematically spoken we want to find the smallest positive λ such that the vector $a + \lambda(c - a)$ has at least one component equal to zero. Then $b = a + \lambda(c - a)$. Assume that $b_k = 0$, then c_k, is the coordinate in which c deviates most from the line $\overline{0, \mu(0)}$. It is important to know this coordinate because if we want to force point c back to the line, then k indicates the right direction, see Section 4 below.

We first want to determine b. Let

$$\lambda_k := \frac{a_k}{a_k - c_k}, \tag{5}$$

then $a_k + \lambda_k(c_k - a_k) = 0$ and $\lambda_k \geq 0$ iff $c_k \leq a_k$. Let

$$\lambda_* := \min\{\, \lambda_k \mid 1 \leq k \leq K, c_k \leq a_k\} \tag{6}$$

be the smallest $\lambda_k \geq 0$. Then $b = a + \lambda_*(c - a)$ and from Eqs. (5), (6) and (4) we obtain

$$\begin{aligned}
\delta(c) &= \frac{c-a}{b-a} = \frac{c-a}{\lambda_*(c-a)} = \frac{1}{\lambda_*} = \frac{1}{\min\{\, \lambda_k \mid 1 \leq k \leq K, c_k \leq a_k\}} \\
&= \max\{\frac{1}{\lambda_k} \mid 1 \leq k \leq K, c_k \leq a_k\} = \max\{\frac{a_k - c_k}{a_k} \mid 1 \leq k \leq K, c_k \leq a_k\} \\
&= 1 - \min\left\{ \frac{c_k}{a_k} \mid 1 \leq k \leq K, c_k \leq a_k \right\} \\
&= 1 - \frac{1}{\alpha} \min\left\{ \frac{c_k}{\mu_k(0)} \mid 1 \leq k \leq K, c_k \leq a_k \right\} \\
&= 1 - \frac{1}{\alpha} \min\left\{ \frac{c_k}{\mu_k(0)} \mid 1 \leq k \leq K \right\}.
\end{aligned} \tag{7}$$

For the last equation note that for the minimizing indices k^* with

$$\frac{c_{k^*}}{\mu_{k^*}(0)} = \min\left\{ \frac{c_k}{\mu_k(0)} \mid 1 \leq k \leq K \right\}$$

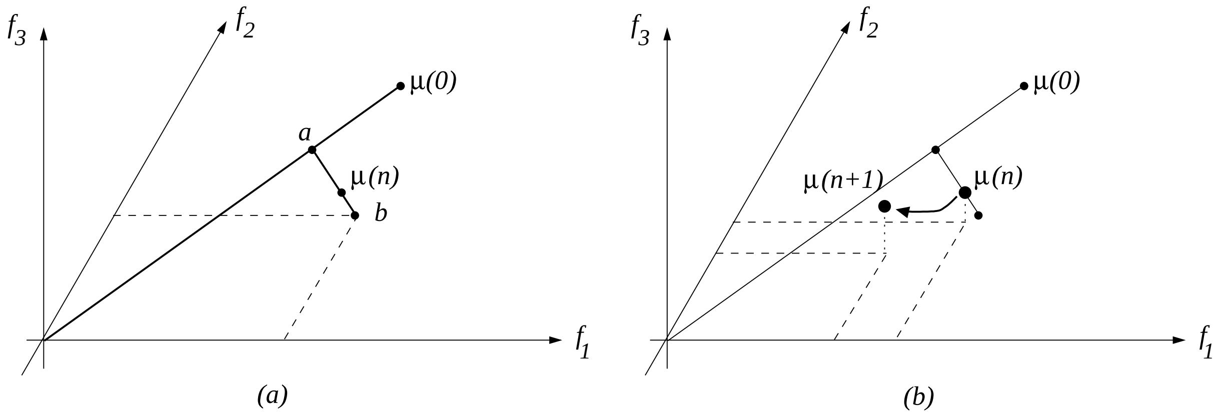


Figure 7: In (a) the point b lies on the f_1, f_2-hyperplane, determined by $\Theta_1(G_n) = \{3\}$ for the case $K = 3$. (b) If we reduce the values of f_1, f_2 in $G - n$ and allow larger values for f_3 then the center $\mu(n)$ will be forced closer to $\overline{0, \mu(0)}$

the constraint $c_{k^*} \leq a_{k^*}$ automatically holds. To prove this, assume that $c_{k^*} > a_{k^*} = \alpha \cdot \mu_{k^*}(0)$. Then, from the minimizing property of k^* we obtain for all k

$$\frac{c_k}{\mu_k(0)} \geq \frac{c_{k^*}}{\mu_{k^*}(0)} > \frac{a_{k^*}}{\mu_{k^*}(0)} = \alpha.$$

Multiplying with $\mu_k(0)^2$ and summing over all k then yields a contradiction :

$$\langle c, \mu(0) \rangle = \sum_{k=1}^{K} c_k \mu_k(0) > \alpha \sum_{k=1}^{K} \mu_k^2(0) = \frac{\langle \mu(0), c \rangle}{\mu(0)^2} \sum_{k=1}^{K} \mu_k^2(0) = \langle \mu(0), c \rangle.$$

■

With $\delta(\cdot)$ as in Eq.(2) we put

$$\Delta_1(G_n) := \delta(\mu(n)) \tag{8}$$

As mentioned above, the minimizing indices k in Eq. (7) determine the direction, i.e. those cost functions f_k in whose direction $\mu(n)$ deviates most from $\overline{0, \mu(0)}$, see Fig. 7. Hence we keep this information in an auxiliary variable Θ_1 :

$$\Theta_1(G_n) := \left\{ l \mid 1 \leq l \leq K, \frac{\mu_l(n)}{\mu_l(0)} = \min_{1 \leq k \leq K} \frac{\mu_k(n)}{\mu_k(0)} \right\} \tag{9}$$

In most cases, Θ_1 will contain only one value. Note that for $K = 2$, $\Theta_1(G_n) = \{1\}$ means that $\mu(n)$ lies above the line $\overline{0, \mu(0)}$ and $\Theta_1(G_n) = \{2\}$ means that $\mu(n)$ lies below the line $\overline{0, \mu(0)}$, see Figure 8.

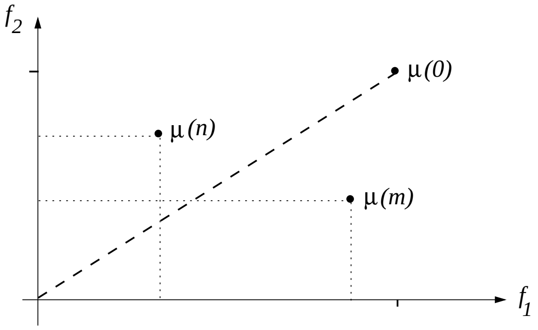


Figure 8: The position of the center $\mu(\cdot)$ for $\Theta_1(G_n) = \{1\}$ and $\Theta_1(G_m) = \{2\}$.

3.2 Deviation of the Variation $\sigma(n)$

To measure the violation of goal 2 we compare the total 'span' $\sum_{i=1}^{K} \sigma_i(n)$ of all values in F_n with those in the initial population. In order to obtain values between 0 and 1 we define

$$\Delta_2(G_n) := \left[1 - \frac{\sum_{i=1}^{K} \sigma_i(n)}{\sum_{i=1}^{K} \sigma_i(0)}\right]^+, \tag{10}$$

where $[\cdot]^+$ denotes the positive part with $[y]^+ = y$ for $y \geq 0$ and $[y]^+ = 0$ for $y < 0$. Values $\Delta_2(G_n)$ near 1 indicate that the population lost diversity in its cost function values whereas populations with $\sum_{k=1}^{K} \sigma_k(n)$ as least as large as in the initial population have $\Delta_2 = 0$. Of course, this is only a very rough indicator, but in conjunction with the other components of Γ_n it turned out to be sufficient. An analogon to Θ_1 seems not reasonable for goal 2.

3.3 Deviation from Goal 3

Goal 3 requires a 'connected' population. Deviation from this property could be measured using sophisticated methods from statistical cluster theory (see e.g. [7]). These methods usually require at least $O(N^2)$ operations.

In our context, we only need a rough indicator of configurations as in Fig. 5 (c). To detect separation into subpopulations we shrink the enclosing rectangle of F_n around its center $\mu(n)$ until it only contains the innermost third of the points $f(x)$, $x \in G_n$. If the points in F_n are evenly spread then the shrinking factor should be $\sqrt[K]{1/3}$. If they are concentrated in the outer part of F_n, as it will be the case if F_n tends to split into several parts, then the shrinking factor should be larger, near one, see Figure 9.

To shrink the enclosing rectangle we multiply the side length vector $\sigma(n)$ as defined in Section 2 with a factor $0 < r < 1$. Note that a point $f(x) = (f_1(x), \ldots, f_k(x))$ lies within a rectangle shrinked by r iff

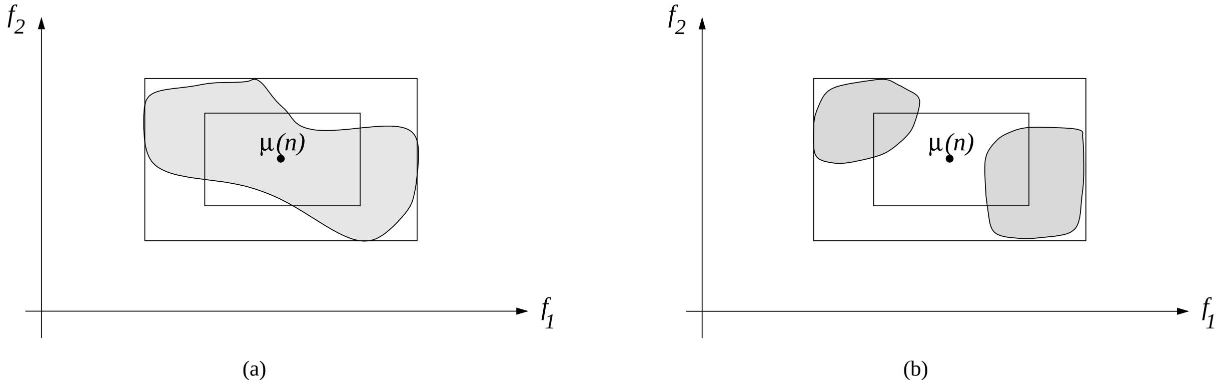


Figure 9: The inner rectangle is obtained by shrinking with the factor $r = \sqrt{1/3} \sim 0.57$. (a) shows a population with $\Delta_3 = 0$, whereas in (b), the inner rectangle will contain less than 1/3 of all solutions, hence $\Delta_3 > 0$.

$$\mu_k(n) \; - \; \frac{1}{2} r\sigma_k(n) \; \leq \; f_k(x) \; \leq \; \mu_k(n) \; + \; \frac{1}{2} r\sigma_k(n) \tag{11}$$

for all $1 \leq k \leq K$, in other words iff

$$d(x) := 2 \cdot \max_{1 \leq k \leq K} \frac{|f_k(x) - \mu_k(n)|}{\sigma_k(n)} \; \leq r.$$

(cp. Figure 10). Then $|\{x \in G_n \| d(x) \leq r\}|$ is the number of solutions in population G_n with cost function values within the rectangle shrinked by factor r. Now let $\rho_\beta(G_n)$ be the smallest factor such that the shrinked rectangle contains $\beta \cdot 100\%$ of the points of F_n, i.e.

$$\rho_\beta(G_n) := \min\{ \, r > 0 \mid |\{ \, x \in G_n \mid d(x) \leq r \, \}| \geq \beta \cdot N\},$$

see Fig. 9. $\rho_\beta(G_n)$ is a kind of empirical β quantile with $0 \leq \rho_\beta(G_n) \leq 1$ for $0 \leq \beta \leq 1$ and $\rho_1(G_n) = 1$. If the points of F_n are evenly distributed around $\mu(n)$ then $\rho_\beta(G_n) \sim \sqrt[K]{\beta}$. A measure for the deviation from an even distribution therefore is (with $\beta := 1/3$) the

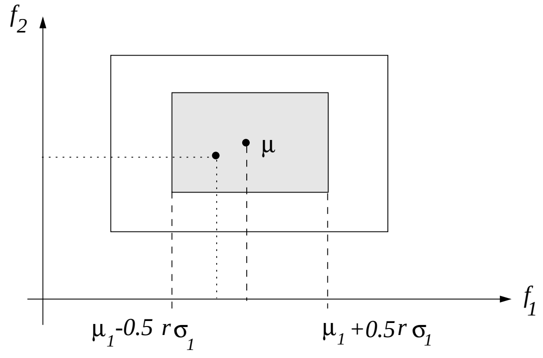


Figure 10: A point $(f_1(x), f_2(x))$ lies within the rectangle shrinked by a factor r if it fulfills Eq. 11

difference $[\rho_{1/3}(G_n) - \sqrt[K]{1/3}]^+$. In order to obtain values between zero and one we divide this expression by its maximal value $1 - \sqrt[K]{1/3}$ and obtain after simplifying

$$\Delta_3(G_n) := \left[\frac{\rho_{1/3}(G_n) \cdot \sqrt[K]{3} - 1}{\sqrt[K]{3} - 1}\right]^+ \tag{12}$$

Again this is only a very rough measure that does not necessarily detect any kind of splitting of the population but it turned out sufficient for our purposes. Typically, a non-separated cloud will have more than one third of its points in the inner rectangle (i.e. $\Delta_3 = 0$) as its point will be more sparse near the border of the enclosing rectangle, see Fig. 9 a). This gives our measure a bias towards zero and triggers the action for a more spread population (see Section 4) more often. This is a desirable effect as it counteracts the general tendencies of GAs to loose variation within the population.

A definition of $\Theta_3(G_n)$ does not seem reasonable.

3.4 Deviation from Goal 4

Goal 4 requires the lower left corner $\lambda(n)$ of F_n to move along the line $\overline{0, \lambda(0)}$. Hence we may use the distance measure from 3.1 with $\mu(0)$ replaced by $\lambda(0)$ and obtain

$$\Delta_4(G_n) = 1 - \frac{\sum_{k=1}^K \lambda_k(0)^2}{\sum_{k=1}^K \lambda_k(0)\lambda_k(n)} \min_{1\le k\le K} \frac{\lambda_k(n)}{\lambda_k(0)}. \tag{13}$$

In $\Theta_1(G_n)$ we only kept the minimizing coordinates, i.e. those in which $\mu(n)$ deviates most. Here, we keep more detailed information about the position of $\lambda(n)$. Let

$$p_k(n) := \frac{\lambda_k(n)/\lambda_k(0)}{\sum_{l=1}^K \lambda_l(n)/\lambda_l(0)} \tag{14}$$

then $0 \le p_k(n) \le 1$ and $\sum_{k=1}^K p_k(n) = 1$. $p_1(n), \ldots, p_K(n)$ represents a relative weighting of the coordinates giving least weight to those coordinates in which $\lambda(n)$ deviates most

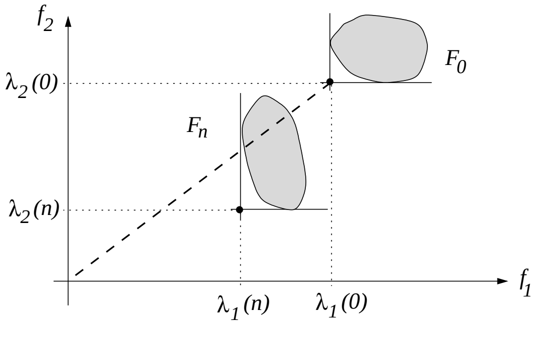


Figure 11: An example with $\lambda_1(n)/\lambda_1(0) \sim 2\ \lambda_2(n)/\lambda_2(0)$, hence we have $p_1(n) = 2/3, p_2(n) = 1/3$.

from $\overline{0, \lambda(0)}$, see Figure 11. This will be used later to calibrate the countermeasure. Therefore we put

$$\Theta_4(G_n) := \{p_1(n), \ldots, p_K(n)\}. \tag{15}$$

Now the complete (crisp) state of population G_n is given as

$$\Gamma_n = \big(\Delta_1(G_n), \Delta_2(G_n), \Delta_3(G_n), \Delta_4(G_n), \Theta_1(G_n), \Theta_4(G_n)\big) \tag{16}$$

defined in Eqs. (8), (10), (12), (13) , (9) and (15).

4 Actions to Reduce Deviation

Next we want to identify possible control actions that could help to diminish the deviation of a population. We restrict ourselves here to different reduction methods.

As indicated in Fig. 4, an "ideal" state is given by

$$\Gamma^* := (0,0,0,0,\{1,...,K\},\{1/K,\ldots,1/K\})$$

with no deviations, high variance, and 'connected' population. We need a decision on what kind of reduction "$G_n^+ \to G_{n+1}$" is to be used given an arbitrary state $\Gamma_n = (\Delta_1,\ldots,\Delta_4,\Theta_1,\Theta_4)$. The reduction should lead to a population G_{n+1} which has its state Γ_{n+1} 'closer' to Γ^* than Γ_n if the values $\Delta_1(G_n),...,\Delta_4(G_n)$ are too large.

We define a set of reduction methods that from our experience are suited to lessen the deviation for each of the goals. Our methods are modifications of standard methods from the literature. We briefly characterize their behaviour in our experiments (see also the quantitative description in Section 7 below).

"E" (Elite): Use the fitness function

$$g(x) = \sum_{\substack{k=1 \\ k\notin\Theta_1(G_n)}}^{K} f_k(x), \qquad x \in S$$

and take the N best individuals of G_n^+ for the next generation G_{n+1}. Here, the f_k with $k \in \Theta_1(G_n)$ are dropped from the fitness function.

Let us recall that $\Theta_1(G_n)$ contains those indices k where $\mu_k(n)$ deviates most from $\overline{0,\mu(0)}$. Hence, the selection pressure during the reduction with $g(\cdot)$ as above, will concentrate on the coordinates $k \notin \Theta_1(G_n)$. Consequently, in the cloud F_{n+1} the points will have smaller values in the coordinates $k \notin \Theta_1(G_n)$, whereas the f_k- values for $k \in \Theta_1(G_n)$ will tend to increase. Then, $\mu(n+1)$ will lie closer to $\overline{0,\mu(0)}$ than $\mu(n)$, see Figure 7 (b) for an example.

"P" (Pareto-rank): This is taken from [4, 5]. Each individual x is ranked by the number

$$r(x) := 1 + |\{\, x' \in G_n^+ \mid x' \preceq x \,\}| \tag{17}$$

of dominating co-individuals. $r(x)$ is called Pareto rank of x. The smaller $r(x)$ is, the better the solution x is. The N solutions with lowest $r(\cdot)$-value are selected for G_{n+1}. From our observations this reduction will tend to concentrate the cloud F_{n+1}, reducing the variation and possible tendencies to split. If applied constantly, a large deviation of the center may occur as shown in Figure 16 below.

"V" (VEGA): The VEGA algorithm (see [17]) takes the $\frac{N}{K}$ best solutions from G_n^+ with respect to f_1 and puts it into G_{n+1}, then, from the remaining solutions in G_n^+, it takes the $\frac{N}{K}$ best solutions with respect to f_2 and puts it into G_{n+1} and so on until all K cost functions $f_1, \ldots, f_K$ have been used. As has been noticed before (see [10]), this procedure leads to a splitting of the population as indicated in Fig. 5 (c) (see also Figure 14 below). The positive effect is that the overall variation in the population will increase.

"V*" (VEGA*): This is a slight modification of VEGA. Instead of a fixed fraction $\frac{N}{K}$, take the $N \cdot p_k(n)$ best individuals with respect to f_k, $1 \leq k \leq K$, where the $p_k(n)$, see Eq. (14), are obtained from $\Theta_4(G_n)$.

To understand the effect of this choice let us consider a coordinate k with large value $p_k(n)$. Then the deviation of $\lambda(n)$ from $\overline{0, \lambda(0)}$ is relatively small in the k-th component. As $N \cdot p(k)$ is a large portion, many individuals are chosen according to f_k. For these individuals the values of $f_{k'}$, $k' \neq k$, are not taken into account for reduction, hence the variation in coordinates $k' \neq k$ will increase in F_{n+1}. This enables the next population to decrease the distance of $\lambda(\cdot)$ from $\overline{0, \lambda(0)}$. Similar to VEGA, this procedure tends to spread the population, see also Figure 15.

Note that each of these methods has several disadvantages if used constantly. The main point of our method "FuReGa" (Fuzzy Reduction Genetic Algorithm) is the adaptation to the actual state of the population after each generation.

In [10] a method is presented where the two individuals for crossover are selected with a probability proportional to the fitness function $g(x) := \sum \alpha_i f_i$ where $\alpha_1, \ldots \alpha_K$ are chosen randomly for each pair of individuals. The offspring individuals are taken as new population without any additional reduction. The system also includes the survival of some of the good individuals without crossover or selection. This method seems to be better than the others if applied constantly. Still the experiments in Section 7 show that our system yields better results.

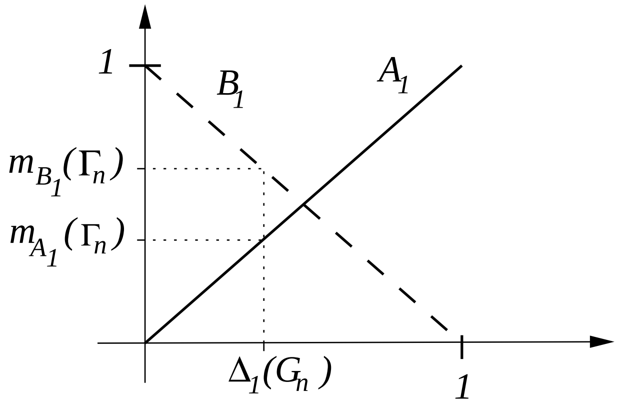


Figure 12: The membership functions used to describe the two complementary fuzzy sets A_i, B_i.

5 The Fuzzy Reduction Step

Though the different countermeasures in Section 4 are chosen to meet our goals, their properties can only be given in vague terms, in particular as they are of stochastic nature. For the selection of the appropriate reduction method we therefore suggest a fuzzy rule based system. It takes as input the state Γ_n of the present population and yields one of the methods “E”, “P”, “V” or “V^*” as output. We follow the fuzzy rule based system (FRBS) as treated, e.g., in [2] and [3]. Instead of introducing the method in all its generality we give a step by step explanation of it applied to our context.

We start by classifying the state Γ_n with respect to each of our four goals as either bad or good. More precisely, we define *fuzzy sets* A_i and B_i for each of the goals $i = 1, \ldots, 4$. The set A_i represents the predicate “deviation $\Delta_i(G_n)$ is too high”, whereas B_i represents “deviation $\Delta_i(G_n)$ is tolerable”. The fuzzy sets are defined by their individual *membership functions* m_{A_i} resp. m_{B_i}. $m_{A_i}(z)$ is a value between 0 and 1, expressing the degree (belief) to which the point z belongs to set A_i, i.e. the degree to which we are willing to consider a deviation z as ’too high’. Though the predicates represented by A_i and B_i are complementary, a point may well have positive membership values for both sets expressing the fuzzy nature of this classification. E.g., for a given deviation z we may have $m_{A_1}(z) = .3$, $m_{B_1}(z) = .7$ which means that the deviation z is tolerable though not zero.

To keep things simple, we chose as membership functions for all $i = 1, \ldots, 4$

$$m_{A_i}(z) \ := \ z \quad \text{and} \quad m_{B_i}(z) \ = \ 1 - z$$

for $0 \leq z \leq 1$, see Fig. 12.

IF	rule no.					
F_n is in state	1	2	3	4	5	6
	$*$	$*$	A_1	B_1	B_1	B_1
AND	A_2	A_2	B_2	B_2	B_2	B_2
AND	$*$	$*$	$*$	A_3	B_3	B_3
AND	A_4	B_4	$*$	$*$	A_4	B_4
THEN						
use method	V*	V	E	P	V*	V

Table 1: 6 rules for all combinations of the four properties. A "*" in the i -th row means 'A_i OR B_i'

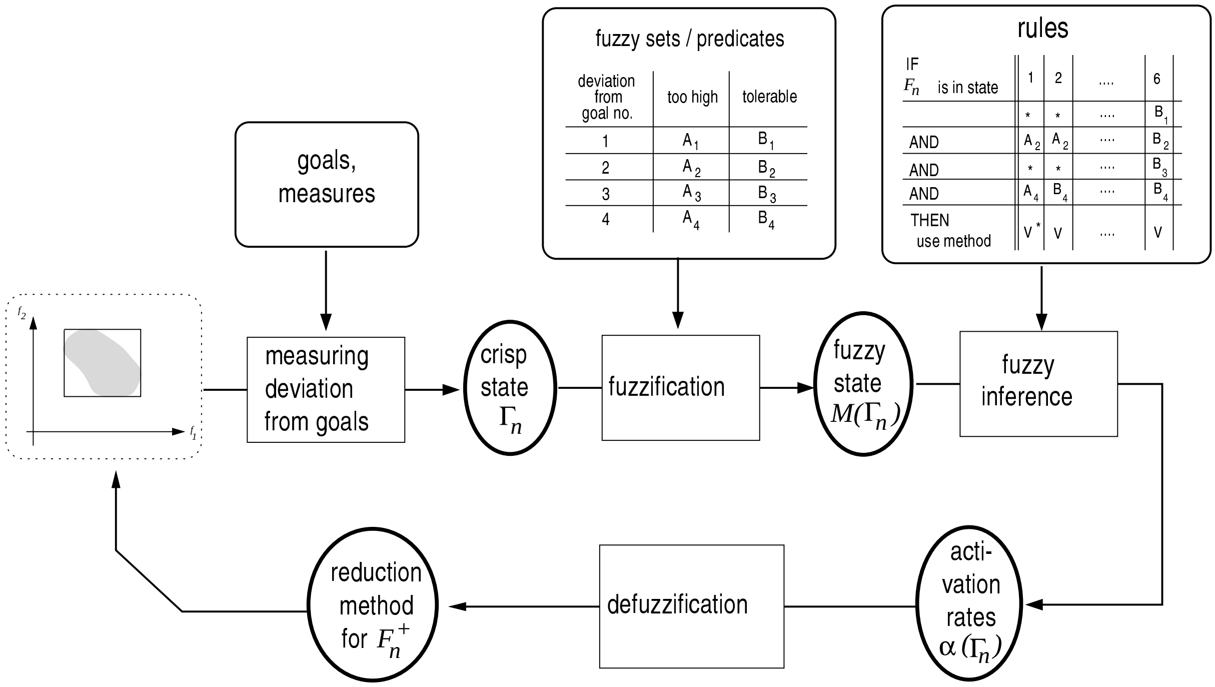
To simplify notation, we put $m_{A_i}(\Gamma_n) \ := \ m_{A_i}(\Delta_i(G_n))$ and similarly for B_i. The complete *fuzzy classification* of a state Γ_n with respect to the eight predicates is the sequence of its membership values for the different fuzzy sets :

$$\begin{aligned} M(\Gamma_n) \ &:= \ (m_{A_1}, m_{A_2}, m_{A_3}, m_{A_4}, m_{B_1}, m_{B_2}, m_{B_3}, m_{B_4})(\Gamma_n) \\ &= \ (\Delta_1, \Delta_2, \Delta_3, \Delta_4, 1-\Delta_1, 1-\Delta_2, 1-\Delta_3, 1-\Delta_4) \qquad (18) \end{aligned}$$

where we have written Δ_i for $\Delta_i(G_n)$ for short. Note that only the Δ_i-parts of state Γ_n are evaluated.

In the next step, this fuzzy classification is fed into a set of rules. Each *rule* has the form " IF $\langle$ condition $\rangle$ THEN $\langle$action$\rangle$". Here, conditions are AND-combinations of the predicates described by the fuzzy sets A_i resp. B_i and actions are the reduction methods introduced in the last Section. The rules we used in our application in Section 7 are given in Table 1. E.g., rule 4 in that table tells us, that if the deviation of $\mu(n)$ is not too high ("B_1") and the variation within the cost values is large enough ("B_2") and if the population is split ("A_3") then, no matter how the fourth goal is met ("*"), we should use method "P" for the next reduction. This is reasonable because we know that "P" diminishes variation. Formulating the set of rules is a difficult step as we have to cast our vague experience into a formal framework.

We now have a set of rules referring to the predicates and a fuzzy classification $M(\Gamma_n)$ of the state of the population with the membership values for all these predicates as given in Eq. (18) . To apply the rules to $M(\Gamma_n)$ we multiply the membership values of


goals, measures
f_2
f_1
measuring deviation from goals
crisp state Γ_n
fuzzification
fuzzy state $M(\Gamma_n)$
fuzzy inference
acti-vation rates $\alpha(\Gamma_n)$
defuzzification
reduction method for F_n^+
fuzzy sets / predicates
rules

deviation from goal no.	too high	tolerable
1	A_1	B_1
2	A_2	B_2
3	A_3	B_3
4	A_4	B_4

IF F_n is in state	1	2	....	6
	*	*	....	B_1
AND	A_2	A_2	....	B_2
AND	*	*	....	B_3
AND	A_4	B_4	....	B_4
THEN use method	V^*	V	....	V

Figure 13: An overview of the fuzzy rule based system as used by our algorithm FuReGa.

those fuzzy sets combined by AND's in the IF-part of the rule (and add them for OR's which are implicitely contained in the "*"-entries of the table meaning A_i OR B_i). This transfers the sequence of membership values in $M(\Gamma_n)$ into *one* real number for each rule. This number reflects the degree of fulfillment of the corresponding rule and is called the *activation rate* of the action in the THEN-part of the rule. If there are several rules with identical actions in their THEN-parts, then the values from all these rules are added to yield the overall activation rate of this action. In our example we obtain as activation rate α_P for action "P" from rule no. 4 :

$$\begin{aligned}\alpha_P(\Gamma_n) &:= m_{B_1}(\Gamma_n)\cdot m_{B_2}(\Gamma_n)\cdot m_{A_3}(\Gamma_n)\cdot(m_{A_4}(\Gamma_n)+m_{B_4}(\Gamma_n))\\ &= (1-\Delta_1)(1-\Delta_2)\Delta_3\end{aligned}$$

The activation rates for the other actions from Table 1 are

$$\begin{aligned}\alpha_E(\Gamma_n) &= \Delta_1(1-\Delta_2) \qquad \text{(rule 3)}\\ \alpha_V(\Gamma_n) &= \Delta_2(1-\Delta_4)+(1-\Delta_1)(1-\Delta_2)(1-\Delta_3)(1-\Delta_4) \qquad \text{(rules 2, 6)}\\ \alpha_{V^*}(\Gamma_n) &= \Delta_2\Delta_4+(1-\Delta_1)(1-\Delta_2)(1-\Delta_3)\Delta_4 \qquad \text{(rules 1,5).}\end{aligned}$$

Activation rates give a weight to the actions expressing the intensity by which the rules recommend each action given the fuzzy classification $M(\Gamma_n)$. There are different ways of *defuzzification*, i.e. different ways to finally select the action that is actually activated : e.g. one could choose the one with maximal activation rate. For our table of (exhaustive) rules with the complementary membership functions it can be shown that the activation rates $(\alpha_E, \alpha_P, \alpha_V, \alpha_{V^*})$ always represent a probability distribution on the set of possible actions {"E", “P”, “V”, “V*”}. It therefore seems natural to select an action randomly, according to this distribution.

Figure 13 sums up our fuzzy rule based system.

6 Summary of the FuReGa Method

We can now summarize our method FuReGa with all its steps. Note that we do not discuss how to choose the population size, the mutation probability and other details of the GA. Instead we concentrate on the new features of our method. See also Fig. 2 and Fig. 13.

(1) Produce an initial population G_0 of size $N \in \mathbb{N}$. Preferably, the individuals should be produced at random.

(2) Put generation number $n = 0$.

(3) Steps 4 and 5 could be performed in parallel :

(4) Select the appropriate selection method for G_n :

> **(4.1)** Evaluate the present population G_n by calculating the enclosing rectangle with $\mu(n), \sigma(n), \lambda(n)$ as in Eq. (1).
> Calculate the state Γ_n as in Eq. (16).
>
> **(4.2)** Apply the fuzzy rule based system as in Fig. 13 :
> Determine the membership values of Γ_n for the fuzzy sets A_i, B_i, $i = 1, \ldots, 4$.
> Determine the activation rates for the actions "E", "P", "V", "V*" from a table of rules as in Table 1.
> Select a reduction method according to the activation rates found.

(5) Produce offspring from population G_n using standard crossover and mutation mechanisms. The result is an enlarged population G_n^+ of size $N+$ no. of offspring.

(6) Apply the reduction method selected in step 5 to the enlarged population G_n^+. The result is population G_{n+1} of size N.

(7) Update the set P of Pareto-optimal solutions found so far by adding new undominated solutions and deleting those which have become dominated by the newly found solutions.

(8) Increase the generation number $n = n + 1$.

(9) If a stopping criterion becomes true, e.g. n exceeds a given number of generations then stop. P then contains the approximation to the Pareto-optimal solutions. Otherwise go to step 3.

7 Cost benefit analysis for timetables

As a practical example, which was developed in cooperation with German traffic companies, we considered a problem of timetable optimization for railway networks.

Railway companies in Europe are presently trying to minimize the waiting times that occur when passengers have to change lines within a network, see [6, 11, 12] (in German). To obtain so-called integrated timetables with no waiting times at the important stations the companies have to invest into their networks (tunnels, faster trains, larger stations, etc.). We implemented a system 'HiTT' that uses the German intercity network to demonstrate the effect of invested capital on the minimal total waiting time W of the net. This is achieved by applying FuReGa to timetables with the two cost functions 'waiting time' and 'additional investments'. For the details of the system see e.g. [15, 16].

A traffic net in this model consists of a fixed number of stations and tracks connecting them. There is a fixed set of lines (i.e. sequences of stations and tracks) which have to be served periodically with given frequencies. For each track there is also a set of possible improvements, each listed with the amount of money necessary for its realization and its effect on the running time of the trains on that track. These can be viewed as local cost functions $c_i(t)$ giving the amount of money necessary to shorten the travel time on track i by t minutes.

A time table $x = (\pi, \varrho)$ consists of scheduled departure and travel times. $\pi = (\pi_{l,s})$ is the vector of departure times $\pi_{l,s}$ for each station s and each line l. As the trains are assumed to run periodically every hour, $\pi_{s,l}$ is simply the offset from the full hour. $\varrho = (\varrho_i)$ is the vector of scheduled travel times ϱ_i on each track i. The time table $x = (\pi, \varrho)$ is evaluated by two cost functions. $f_1(x) = f_1(\pi, \varrho)$ sums up the waiting times from all possible line changes in the net, each change weighted with the number of passengers expected to use this change. The second cost function $f_2(x) = f_2(\varrho)$ is the sum of money as given by $c_i(\cdot)$ that has to be invested into improvements of the tracks in order to obtain the travel times prescribed by ϱ.

The cost function values of the Pareto-optimal solutions (time tables) then yield a cost-benefit-curve showing the effect of investments on the overall waiting times. An integrated time table x^*, e.g., would have $f_1(x^*) = 0$.

In the genetic algorithm formulation of this model we used the x as individuals. Mutation and crossover can easily be formulated as x consists of two strings π and ϱ varying in simple rectangular sets, such that any disturbance or k-point crossover leads to feasible time tables, see [15] for details.

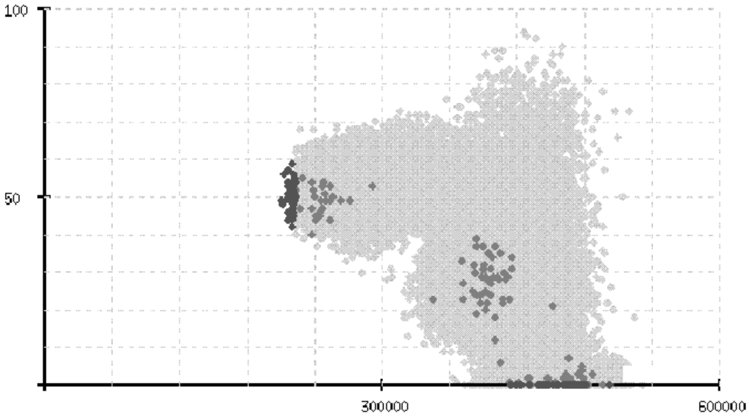


Figure 14: All cost values of 1000 generations using "V"

We first applied the procedure "V" (VEGA) to this problem. Fig. 14 shows the $(f_1(x), f_2(x))$ pairs of all individuals produced in 1000 generations constantly using VEGA. The black points refer to timetables presently selected for the next population G_{n+1}. The dark grey ones are the values from G_n^+ that were not selected for G_{n+1} and the light grey points are from former populations. The population G_{n+1} is obviously split into two extreme parts. Its lower envelope gives only a poor impression of possible Pareto-optimal solutions. A similar result for "V^*" is shown in Fig. 15. In Fig. 16 the corresponding result for "P" (from [4]) is shown. Here, the cost function values are concentrated very much and the population evades the difficult central part of the Pareto curve.

In Fig. 17 the corresponding result for our new fuzzy approach FuReGa is shown. Here, the lower envelope is broader and covers more of the central region. Note that in Fig. 17 the scaling of the axis is different from the other pictures.

The result with the procedure from [10] is shown in Figure 18. Though the population has the desired shape the overall quality of our FuReGa approach is slightly superior in this example.

Though these pictures show only one run, they are representative for the majority of trials we have made.

In Fig. 19 the deviation $\Delta_1(G_n)$ of $\mu(n)$ from the line $\overline{0, \mu(0)}$ is plotted for the procedure "P" used for all 1000 generations (upper line) and for our FuReGa approach. As can be seen, the fuzzy adaption reacts to a deviation and keeps it on a much lower level than with "P" alone. Similar pictures can be given for the other combination of goals and reduction methods. They enable us to qualify the behaviour of the different methods when applied repeatedly and may serve as an empirical basis for the formulation of the inference rules of the fuzzy rule based system.

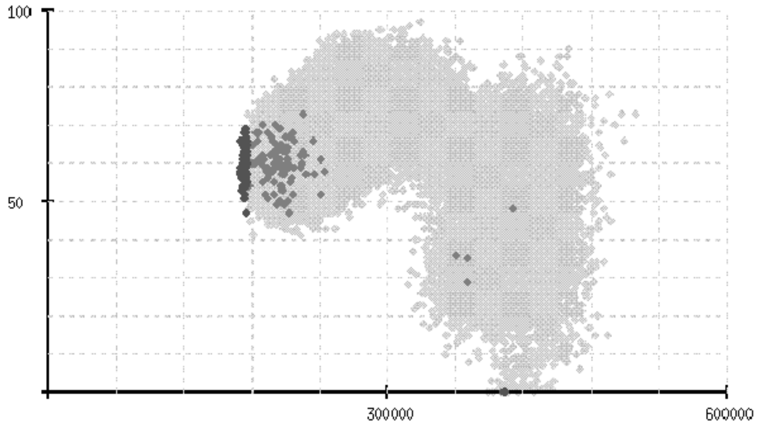


Figure 15: All cost values of 1000 generations using “V*”

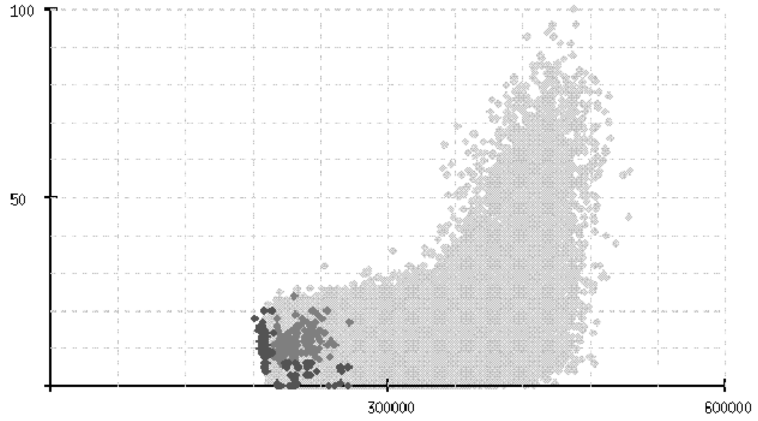


Figure 16: All cost values of 1000 generations using ”P” (Pareto-rank)

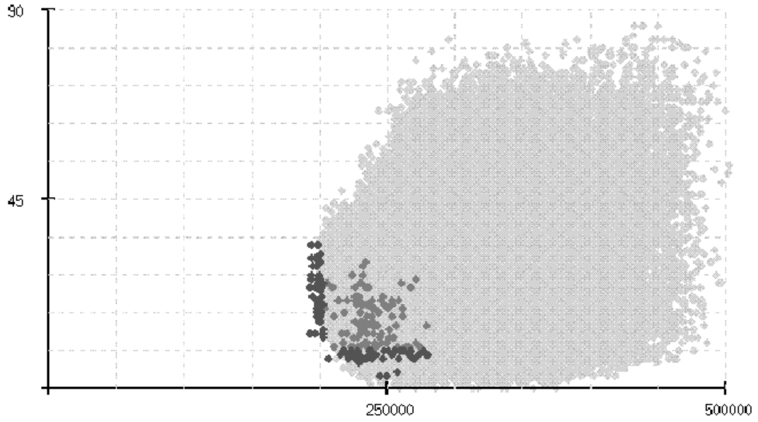


Figure 17: The cost values after 1000 generations using FuReGa. Note that the scaling of the axis has changed.

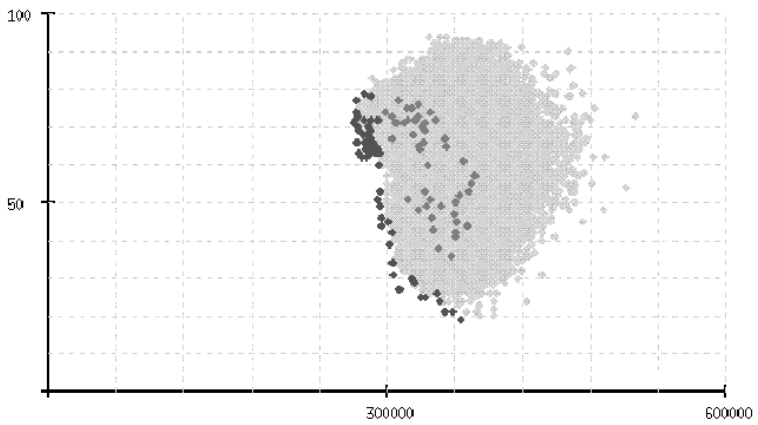


Figure 18: All cost values of 1000 generations using the procedure from [10]

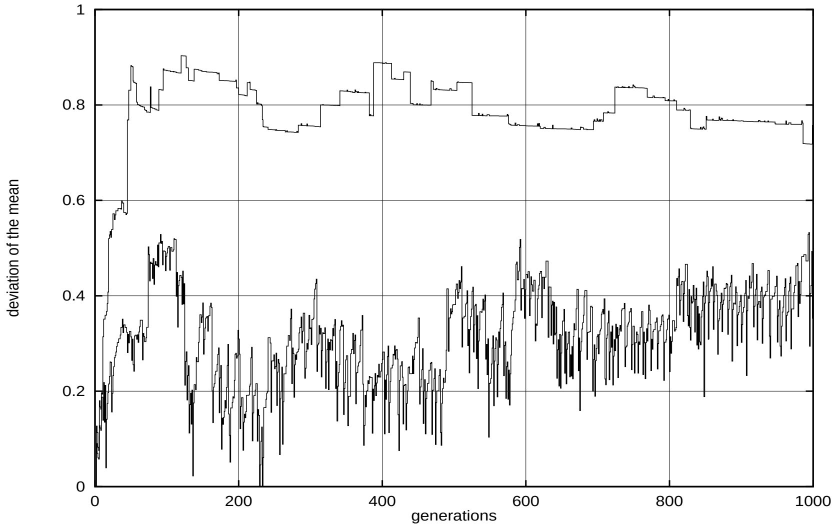


Figure 19: The values of $\Delta_1(G_n)$ for $1 \leq n \leq 1000$ using "P“ (upper line) and FuReGa (lower line)

8 Conclusion

We introduced an adaptive reduction scheme that switches between several multi-objective reduction methods. It evaluates the deviation from explicitly formulated goals as simple numerical values. It feeds them into a fuzzy rule based system that selects a suitable reduction method.

Empirical tests with a ('real world') timetable optimization problem with a two-dimensional cost function showed that the shortcomings of the classical methods of multi-criteria genetic algorithms could be avoided by the adaptive character of our procedure. Still, these limited experiments are not enough to finally decide on the empirical quality of the procedure, in particular in higher dimensions than $K = 2$. Presently we are incorporating further cost functions (number of vehicles needed to run the timetable) into our timetable optimization system. First results look very encouraging.

Also we are experimenting with additional goals and measures of deviation. For a more comprehensive characterization of the reduction methods needed to develop better rules we are planning to perform more experiments with different reduction methods.

Acknowledgement : We would like to thank the referees for their remarks and for drawing our attention to the additional literature about fuzzy genetic algorithms.

References

[1] E.H.L. Aarts, A.E. Eiben, and K.M. van Hee. A general theory of genetic algorithms. *Computing science notes*, 1989.

[2] H. Bandemer and S. Gottwald. *Einfuehrung in Fuzzy - Methoden.* Akademie Verlag Berlin, 1993.

[3] A. Bardossy and L Duckstein. *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems.* CRC Press, New York, 1995.

[4] C.M Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *5th International Conference on Genetic Algorithms*, pages 416–423, 1993.

[5] C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[6] R. Goebertshahn. Der integrale taktfahrplan. *Die Deutsche Bahn*, 5:357–362, 1993.

[7] J.A. Hartigan. *Clustering algorithms.* Wiley, New York, 1975.

[8] F. Herrera and M. Lozano. Adaptation of genetic algorithms parameters based on fuzzy logic controllers. In F. Herrera and J.L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, pages 95 – 125. Physica Verlag, 1996.

[9] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. *IEEE Conference on Evolutionary Computation*, 1:82–87, 1994.

[10] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 119–124, 1996.

[11] M. Kolonko, K. Nachtigall, and S. Voget. Exponat der universitaet hildesheim auf der cebit 96: Optimierung von integralen taktfahrplaenen mit genetischen algorithmen. *Hildesheimer Informatik Berichte*, 8/96:1–39, Maerz 1996.

[12] M. Lichtenegger. Der integrierte taktfahrplan. *ETR - Eisenbahntechnische Rundschau*, 40:171–175, 1991.

[13] S.J. Louis and G.J.E. Rawlins. Pareto optimality, ga easiness and deception. In *International Conference on Genetic Algorithms*, pages 118–123, 1993.

[14] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, 1992.

[15] K. Nachtigall and S. Voget. A genetic algorithm approach to periodic railway synchronization. *Computers And Operations Research*, 23(5):453–463, 1996.

[16] K. Nachtigall and S. Voget. Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks. *European Journal of Operational Research*, 1997. to appear.

[17] J.D. Schaffer and J.J. Grefenstette. Multi-objective learning via genetic algorithms. In *Proceedings of the ninth International Joint Conference on Artificial Intelligence*, pages 593–595, 1985.

[18] P.L. Yu. *Multiple Criteria Decision Making: Concepts Techniques and Extensions.* Plenum Press, New York, 1985.