An evolution strategy for the multiobjective optimization

To Thanh Binh* and Ulrich Korn†
Institute for Automation,
Otto–von–Guericke University,
Universitätsplatz 2, PF 4120, D–39016 Magdeburg, Germany,
Phone: +49 391 671 2732,
Fax: +49 391 671 2526/2229

Abstract

This paper presents an evolution strategy for the multiobjective optimization with any constraints. The main advantage of the evolution strategy is to allow to handle simultaneously multiple objectives and constraints, and to achieve the good approximation of the complete pareto–optimal set. However, it is essentially a global search method for the scalar optimization. As a general–purpose optimization approach, this evolution strategy is implemented in the MATLAB–based[1] environment and used to solve many optimization problems, especially control system design problems ([1, 2]).

1 Introduction

Many engineering design problems are generally multiobjective, in that several design aims (multiple measures of performance) need to be simultaneously achieved. If we describe quantitatively all these objectives as a set of N design objective functions $f_i(\underline{x}) \quad \forall i = \overline{1,N}$, where $\underline{x}$ denotes the n-dimensional vector of design parameters, the design problem could be formulated as a *multiobjective optimization* problem:

$$\min_{\underline{x}} \underline{f}(\underline{x}) = \min_{\underline{x}}(f_1(\underline{x}), f_2(\underline{x}), ..., f_N(\underline{x})).$$

Without loss generality, we suppose, the objective variable $\underline{x}$ must be in an universe Ω of the n-dimensional space, which is often determined by the *auxiliary design constraints*. The auxiliary constraints can have any natur, for example, they can be expressed in terms of function inequalities or equalities etc.. In most cases, the objective functions are in conflict, so that it is not possible to reduce any of the objective functions without increasing at least one of the other objective functions. This is known as the *concept of pareto optimality* ([3]). The multiobjective optimization problems tend to be characterized by a very large set of admissible solutions (trade–offs), known as the set of pareto–optimal solutions:

Definition 1 *An objective variable $\underline{x} \in \Omega$ is said to be pareto–optimal iff there is no $\underline{x}^* \in \Omega$ for which $\underline{f}(\underline{x}^*)$ dominates $\underline{f}(\underline{x})$, i. e., there is no $\underline{x}^* \in \Omega$ such that*

$$f_i(\underline{x}^*) \leq f_i(\underline{x}) \quad \forall i = \overline{1,N}.$$

*E–mail: binh@infaut.et.uni–magdeburg.de
†E–mail: korn@infaut.et.uni–magdeburg.de
[1] MATLAB is the Trademark of the MathWorks, Inc.

Definition 2 *A vector $\underline{x} = (x_1, x_2, ..., x_n)$ and a vector $\underline{y} = (y_1, y_2, ..., y_n)$ in the universe Ω are said to be nondominated to each other if neither $\underline{x}$ dominates $\underline{y}$ nor $\underline{y}$ dominates $\underline{x}$.*

Then, the set of the objective vectors in the objective function space, which corresponds to the set of pareto–optimal solutions, is the set of nondominated or noninferior vectors. Even in the simplest case of two objective functions without any constraints it is very difficult to get the set of pareto–optimal solutions. By handling a number of individuals in the current population the evolution strategies have been recognized to be possibly well-suited to the multiobjective optimization (to approximate the pareto–optimal set by a current population), and therefore they have been discussed quite extensively in the last few years.

In the next section, the evolution strategy for solving the above multiobjective optimization problem is introduced. It is based on the main genetic mechanisms: *Mutation, Reproduction and Selection*, and on ranking according to the *actual concept of pareto optimality.* This evolution strategy guarantees an equal probability of reproduction and selection to all the noninferior individuals in the current population. By this means the pareto–optimal set is approximated by the population in the current generation better than in the last generations. Some examples to illustrate the efficiency of the evolution strategy are shown in section 3.

2 Main elements of the evolution strategy

2.1 Individual representation

The key element of the evolution strategy is an individual that includes: the objective variable $\underline{x} = (x_1, x_2, ..., x_n)$, the strategy parameter vector $\underline{s} = (s_1, s_2, ..., s_n)$ and the corresponding objective function vector $\underline{f}(\underline{x})$, that means:

$$Ind = (\underline{x}, \underline{s}, \underline{f}).$$

Here, the strategy parameter vector $\underline{s}$ has the same dimension as the objective vector $\underline{x}$ and describes the "personal" experiences of each individual by the reproducing its offsprings.

2.2 Mutation

From a randomly chosen parent individual $Ind^{(i)}$ of the current population, the mutation is performed on $Ind^{(i)}$ by adding a normally distributed random vector $\underline{N}(\underline{0}, \underline{s}^{(i)})$ with expectation zero and standard deviation $\underline{s}^{(i)}$ to the objective variable $\underline{x}^{(i)}$, i. e., it generates an offspring with the objective variable :

$$\underline{x}^{(offspring)} \;=\; \underline{x}^{(i)} \;+\; \underline{N}(\underline{0}, \underline{s}^{(i)}).$$

It is clear, all offsprings of the mutated individual are in the hyperellipsoid with the half–axes $(s_1^{(i)}, s_2^{(i)}, ..., s_n^{(i)})$. To decide whether offsprings are viable, all offsprings must be compared with their parent individual using the following *selection scheme* ([2]):

Selection Scheme 1 *An offspring $Ind^{(offspring)}$ of the parent $Ind^{(i)}$ is said to be viable iff:*

- *$\underline{x}^{(offspring)} \in \Omega$, i. e., all the given constraints are satisfied by $\underline{x}^{(offspring)}$,*
- *either the objective vector $\underline{f}(\underline{x}^{(offspring)})$ dominates $\lambda_M \underline{f}(\underline{x}^{(i)})$ or both these objective vectors are non–dominated to each other.*

Here, $\lambda_M \in [0,1]$ is called the choosing–factor for the mutation ([4]).

The efficiency of the mutation on an individual is characterized by the number of its viable offsprings. The self–adaptation of the strategy parameters can be performed by rotating (permutations of $\underline{s}^{(i)}$) the mutation hyperellipsoid along each coordinate axes, until the biggest number of viable offsprings is achieved. By this way, the mutation allows to evolve its own strategy parameters during the search so that its offsprings can be in the down–climbing direction.

2.3 Reproduction

The standard mechanism of the reproduction is the well–known two–points–crossover. It is performed on the strategy parameters as well as on the objective variables. In this case, two crossover points are chosen at random and sorted into the ascending order. Then, the coordinate elements between successive crossover points are exchanged between two parents to produce two new offsprings. The section between the first coordinate element and the first crossover point is not exchanged between individuals. This process is illustrated here representatively only for the objective variables:

From two parents:

$$\underline{x}^{(p1)} = (x_1^{(p1)}, x_2^{(p1)}, ..., x_i^{(p1)}, x_{i+1}^{(p1)}, ..., x_j^{(p1)}, x_{j+1}^{(p1)}, ..., x_n^{(p1)})$$
$$\underline{x}^{(p2)} = (x_1^{(p2)}, x_2^{(p2)}, ..., x_i^{(p2)}, x_{i+1}^{(p2)}, ..., x_j^{(p2)}, x_{j+1}^{(p2)}, ..., x_n^{(p2)})$$

and two crossover points i and j, we have two following offsprings:

$$\underline{x}^{(o1)} = (x_1^{(p1)}, x_2^{(p1)}, ..., x_i^{(p1)}, x_{i+1}^{(p2)}, ..., x_j^{(p2)}, x_{j+1}^{(p1)}, ..., x_n^{(p1)})$$
$$\underline{x}^{(o2)} = (x_1^{(p2)}, x_2^{(p2)}, ..., x_i^{(p2)}, x_{i+1}^{(p1)}, ..., x_j^{(p1)}, x_{j+1}^{(p2)}, ..., x_n^{(p2)})$$

After that, the selection scheme 1 is used to make decision whether these offsprings are viable. The most important problem of the reproduction is to choose the better parent individuals for the crossover. It comes from a motivation: *the evolution strategy favours those individuals of higher quality to reproduce more often than worse individuals.* Therefore we recommend the following selection scheme for reproduction ([2]):

Selection Scheme 2 *The better individuals in the current population must have two properties:*

- *they are non–dominated to each other and to all individuals,*
- *either each of them dominates to the so–called ideal individual or each of them and the ideal individual are non–dominated to each other.*

The ideal individual of the current population can be defined as belows:

Let

$$\begin{aligned} \underline{f}^{(min)} &= (\min f_1, ..., \min f_N) = (f_1^{(min)}, ... f_N^{(min)}) \\ \underline{f}^{(max)} &= (\max f_1, ..., \max f_N) = (f_1^{(max)}, ... f_N^{(max)}), \end{aligned}$$

where the minimum and maximum operators are performed along each coordinate axes of the objective function space for all individuals of the population. Then, the ideal individual has the following objective function vector:

$$\underline{f}^{(ideal)} = \underline{f}^{(min)} + (1-\lambda_R)(\underline{f}^{(max)} - \underline{f}^{(min)}),$$

where $\lambda_R \in [0,1]$ is called the choosing–factor for the reproduction ([4]).

2.4 Selection

The selection is performed by using the pareto–optimality in the objective function space. The algorithm concludes the assigning rank 1 to the non–dominated individuals and rank 2 to the other of the current population. If the number of the non–dominated individuals (denoted by N_{better}) is less than the population size (denoted by N_{pop}), the population of the next generation should consist of these non–dominated individuals and $(N_{pop} - N_{better})$ individuals of the rank 2 that have the shortest distance to the origin of the objective function space. This describes often the situation in some first generations: the current population is so far from the real set of pareto–optimal solutions.

In the other case ($N_{pop} \leq N_{better}$), the population of the next generation includes N_{pop} of N_{better} non–dominated individuals. To choose them we recommend the following selection scheme:

Selection Scheme 3 *For the i-th objective function* ($i = \overline{1, N}$), $\frac{N_{pop}}{N+k}$ ($k \geq 0$) *individuals with the least objective values along the i-th coordinate are chosen. The other individuals are selected at random.*

This algorithm favours the concentrating individuals in the regions of the trade–offs surface, which are in the neighbourhood of the selfish minima, with the higher density than in the other regions. To avoid it and to create the uniform distribution of the current population on the trade–offs surface, the following selection scheme can be used:

Selection Scheme 4

- *For the set of the non–dominated individuals, we determine first the vectors* $\underline{f}^{(min)}$ *and* $\underline{f}^{(max)}$ *(see Selection Scheme 2). Then, the current trade–offs surface is bound in the hyperparallelogram* $\mathcal{R}$ *defined by* $\underline{f}^{(min)}$ *and* $\underline{f}^{(max)}$.
- *Dividing each interval* $[f_i^{(min)}, f_i^{(max)}]$ *into* N_{pop} *small sections* δ_i*, i. e.:*
$$\delta_i = \frac{f_i^{(max)} - f_i^{(min)}}{N_{pop}}.$$
- *In the i-th coordinate axes of the objective function space, the best individual in each of* $\frac{N_{pop}}{N+k}$ *the first sections is selected.*

3 Some applications of the evolution strategy

In this section, we would like to illustrate the efficiency of the evolution strategy using a mathematical example with multiobjective optimization and an application in the control system design.

3.1 Application 1

Consider the simple bi-objective problem of simultaneously minimizing:

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 + x_2^2 \\ f_2(x_1, x_2) &= (x_1 - 5)^2 + (x_2 - 5)^2 \end{aligned}$$

in a region of the objective variable space, defined by $-5 \leq x_1 \leq 10$ and $-5 \leq x_2 \leq 10$. The objective functions have their selfish minima at the points (0,0) and (5,5), respectively. The set of pareto–optimal solutions is well–known and shown in *Fig. 1* (*on the left*) ([2, 4]). From the starting point (-2,6), the current population moves quickly to the pareto–optimal set and reachs it after 5 generations (see *Fig. 1, on the right*).

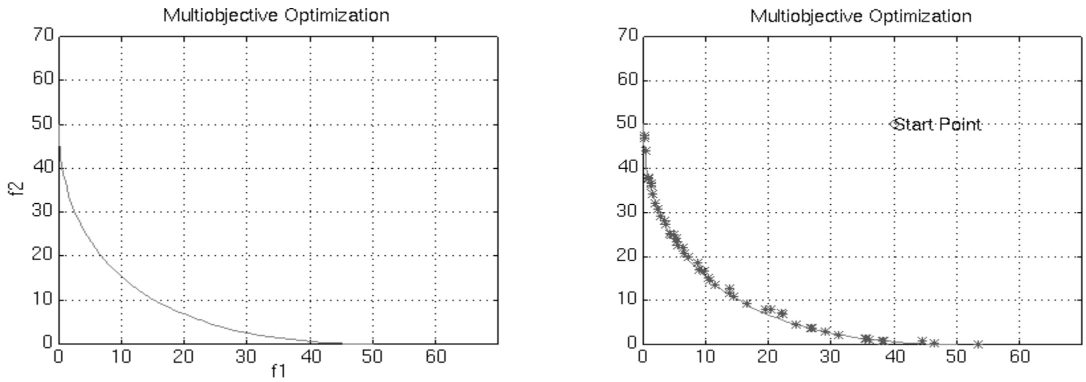

Figure 1: Pareto-optimal set of the bi-objective problem

3.2 Application 2

Here, an application of the evolution strategy to solve a control system design problem that is described as the IFAC 1993 benchmark control problem ([5]). Its specifications include stringent closed–loop performance requirements for a plant with parameters known only within a certain range. Depending on various production conditions, the plant operates at three different stress levels, with higher stress levels inducing larger time variations. For each stress level, design a robust controller to achieve as fast a rise time and a settling time as possible, subject to the following conditions: the plant output and input must be in the interval [-1.2,+1.2] and [-5,+5] at all time, respectively. Using the well–known methods to design a controller, for example, for the stress level 1, the rise time was achieved at the best value 2 sec. by μ-controllers and 2.6 sec. by PI-controllers.

This controller design problem describes a bi-objective problem of the simultaneously minimizing the rise time and settling time with the constraints by the plant input and output values. Using the evolution strategy to solve it, one can achieve a global overview about the structure of the pareto–set between the rise time and settling time in *Fig. 2* (on the left for level 1 and on the right for level 2). The rise time

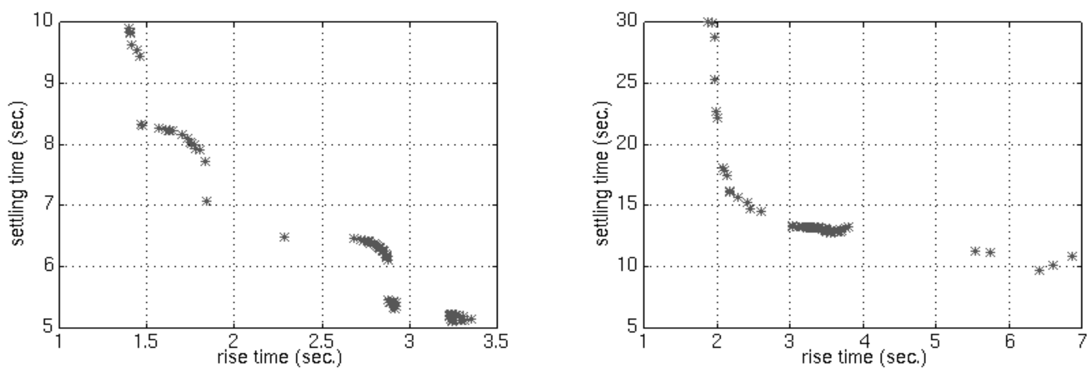

Figure 2: The trade–offs set between the rise time and the settling time

and the settling time for this plant are in conflict. This results are really better than the well-known ones ([5]).

4 The Evolution Strategy Toolbox

A number of MATLAB routines which implement the evolution strategy have been collected together into the Evolution Strategy Toolbox. This toolbox includes two main components:

- Routines for Evolution Strategy: setting the first population, mutation, reproduction and selection,
- Graphical User Interface (GUI): this provides the interface between the user and the evolutionary algorithms and enables the user to create and solve their optimization problems. Moreover, it allows the users to make intelligent dialogues to the achieved results by choosing the best solution from set of admissible solutions.

5 Conclusion

Different to the well-known optimization methods the evolution strategy has not any requirements on the continuity, differentiability of the objective functions and any requirements on the constraints. In general, they possess essentially many good characteristics desirable in the scalar optimization: high robustness to find the global minima or the set of pareto–optimal solutions by the scalar optimization or in the multiobjective optimization, respectively. From these reasons, the evolution strategy can be used effectively to solve many optimization problems.

References

[1] T. T. Binh. Eine entwurfsbeispiel für mehrgrößensysteme durch polgebietsvorgabe. *Automatisierungstechnik, vol. 41, No. 11, 414–417*, 1993.

[2] T. T. Binh. *Eine Entwurfsstrategie für Mehrgrößensysteme zur Polgebietsvorgabe.* PhD thesis, Institut für Automatisierungstechnik, Universität Magdeburg, 1993.

[3] D. Goldberg. *Genetic algorithms in Search, Optimization, and Machine Learning.* Addision–WEsley Publishing Company, Inc., New York, England, Bonn, Tokyo, 1. edition, 1989.

[4] J. Kahlert. *Vektorielle Optimierung mit Evolutionsstrategien und Anwendung in der Regelungstechnik.* Forschungsbericht VDI Reihe 8 Nr.234, 1991.

[5] J. Whidborne and etc. Robust control of an unknown plant – the ifac 93 benchmark. *Int. J. control, vol. 61, No.3, 589–640*, 1995.