A New Mutation Operator for Evolution Strategies for Constrained Problems

Oliver Kramer
International Graduate School
University of Paderborn
D-33098 Paderborn, Germany
okramer@upb.de

Chuan-Kang Ting
International Graduate School
University of Paderborn
D-33098 Paderborn, Germany
ckting@upb.de

Hans Kleine Büning
Department of Computer Science
University of Paderborn
D-33098 Paderborn, Germany
kbcsl@upb.de

Abstract- We propose a new mutation operator - the *biased mutation operator (BMO)* - for evolution strategies, which is capable of handling problems for constrained fitness landscapes. The idea of our approach is to bias the mutation ellipsoid in relation to the parent and therefore lead the mutations into a beneficial direction self-adaptively. This helps to improve the success rate to reproduce better offspring. Experimental results show this bias enhances the solution quality within constrained search domains. The number of the additional strategy parameters used in our approach equals to the number of dimensions of the problem. Compared to the correlated mutation, the BMO needs much less memory and supersedes the computation of the rotation matrix of the correlated mutation and the asymmetric probability density function of the directed mutation.

1 Introduction

Evolution strategies (ES) are the fourth main variant of evolutionary algorithms (EA) besides genetic algorithms (GA), evolutionary programming (EP) and genetic programming (GP). Evolution strategies are most appropriate to numerical optimization [21], as their operators are especially designed for numerical fitness landscapes. The main source for variation is the mutation operator, which is based on the Gaussian distribution to reproduce mutations in the vicinity of the parent, a consequence from the maximum entropy principle for unconstrained search spaces [5]. However, there are still drawbacks, especially in multimodal and constrained problems. There are three main principles for the design of mutation operators which are proposed by Beyer [4]:

- reachability,
- unbiasedness,
- scalability.

The first principle ensures that the whole search space and strategy parameter space can be reached within a finite number of generations. The scalability condition ensures that the mutation strength can adapt to values which guarantee improvements during the optimization process. The condition of unbiasedness is appropriate to unconstrained real search spaces. But for constrained problems evolutionary algorithms with a self-adaptive step size mechanism often suffer from a disadvantageous success probability at the boundary of the feasible search space [14], also see section 3. This results in premature step size reduction and fitness stagnation. In this paper we introduce a new mutation operator, the *biased mutation operator (BMO)*, which is capable of biasing the search into a certain direction and therefore increases the success rate to reproduce better offspring. The number of additional strategy parameters only grows linearly with the number of problem dimensions. Experimental results show that our approach is capable of dealing with constrained problems.

In section 2 we will review the most important mutation operators for evolution strategies: standard uncorrelated mutation with one and with N step sizes, correlated mutation and directed mutation for the purpose of comparison. In the next section 3 constrained problems and premature fitness stagnation is described. In section 4 we will introduce the new *biased mutation operator (BMO)*. In section 5 we will show empirically that our approach is superior to standard mutation. In the last section 6 we summarize the results and present the future work we plan to undertake.

2 Mutation operators for evolution strategies

For a comprehensive introduction to evolution strategies refer to [5]. Here it is important to keep in mind that an individual consists of a vector of objective and strategy variables. In real valued search spaces the objective variables are unencoded real values $x_1, ..., x_N$ representing the assignment of the variables of an N-dimensional optimization problem. The strategy variables contain additional information which are usually important for the mutation operator. There already exists a variety of mutation operators for evolution strategies. For the basic $(\mu \overset{+}{,} \lambda)$-ES [1] standard uncorrelated mutation was introduced by Rechenberg and Schwefel as well as correlated mutation by Schwefel [20]. Many extensions and variations were introduced within the last years, e.g. cumulative step size adaptation by Ostermeier, Gawelczyk and Hansen [18], covariance matrix adaption by Hansen [8], cauchy mutation by Yauo and Liu [23] or directed mutation by Hildebrand [9]. In the following subsection the most important ones are reviewed.

2.1 Uncorrelated mutation with one step size

Mutation is the main source for variation in the evolutionary process. In real-valued search spaces mutation means the addition or subtraction of a small variation, so that a child

[1] The $\overset{+}{,}$-notation combines the notation for both the selection schemes plus- and comma-selection.

0-7803-9363-5/05/$20.00

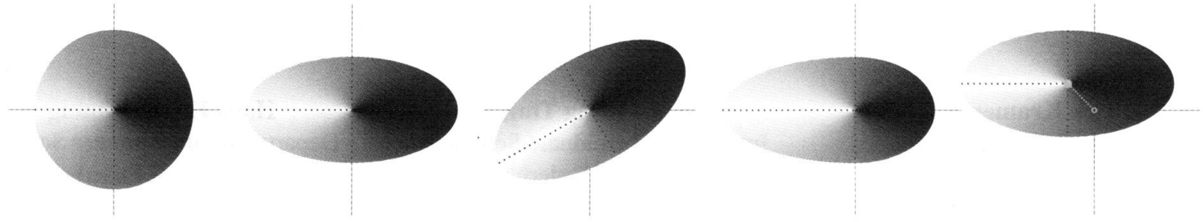

Figure 1: The mutation ellipsoids of the different mutation operators in a two-dimensional search space. For the purpose of better understanding we assume all mutations fall into the ellipsoids instead of taking the probability density functions into account. From left to the right: (a) uncorrelated mutation with one step size, (b) uncorrelated mutation with $N = 2$ step sizes, (c) correlated mutation resulting in rotation, (d) directed mutation with different skewness, and (e) biased mutation making use of the *bias coefficient vector*.

evolves which has got a high similarity to its parent, but is not identical. The simple standard mutation for evolution strategies makes use of only $n_\sigma = 1$ endogenous strategy variable σ, which represents the standard deviation for the normal distribution. For the purpose of better understanding it can be seen as the radius for the sphere in which mutations are created. For ES in real-valued search spaces, objective variables are mutated in the following way:

$$\vec{x}' := \vec{x} + \vec{z} \tag{1}$$

with the mutation

$$\vec{z} := \sigma(\mathcal{N}_1(0,1), ..., \mathcal{N}_N(0,1)) \tag{2}$$

where $\mathcal{N}_i(0,1)$ provides a random number based on a Gaussian distribution with expected value 0 and standard deviation 1. The strategy variable itself is mutated with the log-normal rule:

$$\sigma' := \sigma e^{(\tau \mathcal{N}(0,1))} \tag{3}$$

This mechanism is the key to self-adaptation of the step sizes.

2.2 Uncorrelated mutation with N step sizes

In the basic $(\mu \stackrel{+}{,} \lambda)$-ES normally a vector of $n_\sigma = N$ step sizes is used, which results in mutation ellipsoids:

$$\vec{z} := (\sigma_1 \mathcal{N}_1(0,1), ..., \sigma_N \mathcal{N}_N(0,1)) \tag{4}$$

The corresponding strategy parameter vector is mutated with the extended log-normal rule:

$$\vec{\sigma}' := e^{(\tau_0 \mathcal{N}_0(0,1))} \cdot (\sigma_1 e^{(\tau_1 \mathcal{N}_1(0,1)}, ..., \sigma_N e^{(\tau_1 \mathcal{N}_N(0,1)}) \tag{5}$$

The parameters τ_0 and τ_1 have to be tuned. Comprising, an individual $\vec{a}$ consists of the object parameter set x_i with $1 \leq i \leq N$, the mutation strength vector and the assigned fitness F(x). So it can be specified by

$$\vec{a} = (x_1, ..., x_N, \sigma_1, ..., \sigma_N, F(x)) \tag{6}$$

2.3 Correlated mutation

For some fitness landscapes it is more beneficial to use a rotated mutation ellipsoid for the purpose of an improvement of the success rate. Rotation of the mutation ellipsoid is achieved by the correlated mutation proposed by Schwefel [20]. For an N-dimensional problem $k = N(N-1)/2$ additional strategy parameters, the angles for the rotation of the mutation ellipsoid, are introduced. Let $\vec{\sigma}$ again be the vector of step sizes and $\mathbf{M}$ be the orthogonal rotation matrix. The mutations are reproduced in the following way:

$$\vec{z} := \mathbf{M}(\sigma_1 \mathcal{N}_1(0,1), ..., \sigma_N \mathcal{N}_N(0,1)) \tag{7}$$

For a self-adaptation process the k angles $\alpha_1, ...\alpha_k$ have to be mutated. Schwefel [20] proposed

$$\alpha' = \alpha + \beta \cdot \mathcal{N}(0,1) \tag{8}$$

with $\beta = 0.0873$ corresponding to $5°$.

2.4 Directed mutation

Similar to the idea of the correlated mutation, the assumption of the directed mutation is that in some parts of the landscape a success rate improvement can be achieved by skewing the search into a certain direction. The idea of the directed mutation proposed by Hildebrand [9] is the usage of an asymmetrical mutation operator. This approach demands an asymmetry parameter set of N additional parameters $\vec{c} = (c_1, ...c_N)$. These parameters determine the mutation direction and therefore only cause linear growth of the strategy parameters instead of quadratic growth of the correlated mutation. The idea of skewing the mutations into a certain direction is similar to our approach. But a rather complex calculation of the asymmetric probability density function is necessary, because it has to fulfill the corresponding mathematical specifications of a probability density function. Figure 1 visualizes the effect of the directed mutation operator when mutations are skewed into a certain direction. The highest probability to reproduce mutations is still in the neighborhood environment of the child. In order to decouple asymmetry from the step sizes a normalized directed mutation operator has recently been proposed by Berlik [3].

3 Constrained problems and premature fitness stagnation

3.1 The NLP Problem

Practical optimization problems, e.g. engineering problems, are in most cases constrained. The constrained nonlinear programming (NLP) problem is defined as follows: Find a solution

$$\vec{x} = (x_1, x_2, ..., x_n)^T, \quad (9)$$

which minimizes $f(\vec{x})$:

$$f(\vec{x}) \rightarrow \text{min.}, \quad \vec{x} \in \mathcal{X} \quad (10)$$

$$\text{inequalities} \quad g_i(\vec{x}) \leq 0, \quad i = 1, ..., k \quad (11)$$

$$\text{equalities} \quad h_j(\vec{x}) = 0, \quad j = 1, ..., l \quad (12)$$

A feasible solution $\vec{x}$ satisfies all k inequality and l equality constraints.

3.2 Constraint handling techniques

Various constraint handling techniques exist. For a survey of constraint handling techniques for evolutionary algorithms have a look at [7] or [16]. Most methods fall into the category of penalty functions. The idea of penalty functions is to allow the evolutionary process to discover the whole search space, but to penalize the infeasible part. Static penalty functions use defined penalties [10], [15], while dynamic penalties depend on the number of generations, e.g. see [12]. Annealing penalty functions take an external cooling scheme into account [12], while adaptive methods tune the penalties according to features determining the progress of the evolutionary search [1]. Constraint handling techniques exist that do not use the penalty approach. The decoder method [16], [13] builds up a relation between the constraint search space and an unconstrained one using a decoder function. Repair algorithms repair infeasible solutions using heuristics like minimizing the constraint violation [2], [7]. Multiobjective methods can be used as constraint handling methods when treating the constraints as separate objectives [11], [6], [22], [19]. Recently, Coello introduced a technique based on a multimembered evolution strategy combining a feasibility comparison mechanism with several modifications of the standard ES [17].

3.3 Premature fitness stagnation

Premature fitness stagnation is caused by a disadvantageous success probability near the infeasible part of the search space [14]. In the vicinity of the constraint boundary the success area can be shortened by the infeasible search space. The success area is denoted as the part of the mutation ellipsoid, in which infeasible solutions have got a better fitness than their parent. If the step sizes are bigger than the distance from the center of the mutation ellipsoid to the constraint boundary the success area is shortened. Figure 2 illustrates the situation. However, for smaller step sizes the success area is not shortened. So the process of self-adaptivity favors individuals with smaller step sizes. The consequence of this premature step size reduction is premature fitness stagnation before reaching the optimum. Experimentally premature fitness stagnation is shown in the experiments of section 5.1.

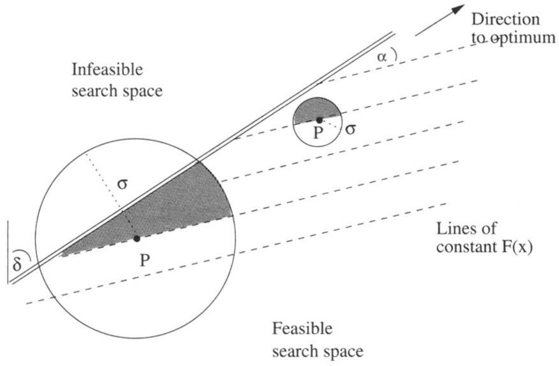

Figure 2: Premature fitness stagnation in the vicinity of the constraint boundary. We assume that all mutations fall into the circles in the case of uncorrelated mutation. The marked success area increases for smaller step sizes (right) in comparison to bigger ones (left).

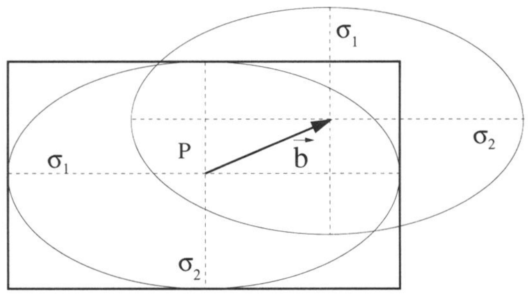

Figure 3: Principle of the BMO in two dimensions: The center of the mutation ellipsoid is shifted by the bias coefficient vector $\vec{b}$ within the bounds of the step sizes $\vec{\sigma}$.

4 The Biased mutation operator (BMO)

Unlike directed mutation, the BMO does not change the skewness, but biases the mean of Gaussian distribution to lead the search into a more beneficial direction. This is reflected in the success rate of reproducing superior offspring. For the BMO we introduce a *bias coefficient vector* $\vec{\xi}$, which indicates the level of bias relative to the standard deviation σ.

$$\vec{\xi} = (\xi_1, ..., \xi_N) \text{ with } -1 \leq \xi_i \leq 1 \quad (13)$$

For every $i \in 1, ..., N$ the bias vector $\vec{b} = (b_1, ..., b_N)$ is defined by:

$$b_i = \xi_i \cdot \sigma_i \quad (14)$$

Since the absolute value of bias coefficient ξ_i is less than or equal to 1, the bias will be bound to the step sizes σ_i. This restriction prevents the search from being biased too far away from the parent. Figure 3 illustrates the BMO. The BMO follows the standard way of mutation:

$$\vec{x}' := \vec{x} + \vec{z}. \quad (15)$$

The mutation in the BMO works as follows:

$$\vec{z} := (\sigma_1 \mathcal{N}_1(0,1) + b_1, ..., \sigma_N \mathcal{N}_N(0,1) + b_N) \quad (16)$$
$$= (\sigma_1 \mathcal{N}_1(0,1) + \xi_1\sigma_1, ..., \sigma_N \mathcal{N}_N(0,1) + \xi_N\sigma_N) \quad (17)$$
$$= (\sigma_1 \mathcal{N}_1(\xi_1, 1), ..., \sigma_N \mathcal{N}_N(\xi_N, 1)) \quad (18)$$

In terms of modifying the mutation strength, the aforementioned log-normal rule is applied. Furthermore, in the BMO the bias coefficients are mutated in the following way:

$$\xi_i{}' = \xi_i + \gamma \cdot \mathcal{N}(0,1) \quad i = 1, ..., N. \quad (19)$$

The parameter γ is a new parameter introduced for the BMO to determine the mutation strength on the bias. In section 5 recommendations will be proposed to tune this parameter. The BMO biases the mean of mutation and enables the

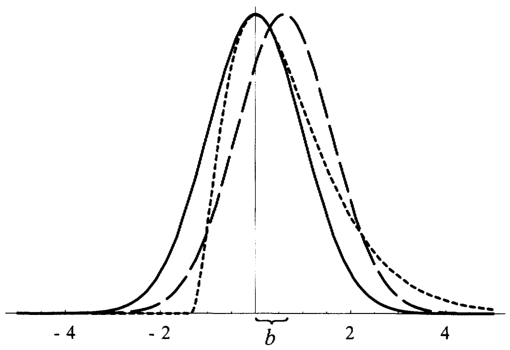

Figure 4: Comparison of standard mutation (solid), directed mutation (dotted), and biased mutation (dashed) with bias b.

evolution strategies to reproduce offspring outside the standard mutation ellipsoid. To direct the search, the BMO enables the mutation ellipsoid to move within the bounds of the regular step sizes. Without the BMO the success rate to reproduce better offspring is relatively low because many mutation lie beyond the feasible search space or have got a worse fitness, as described in section 3.3. The bias coefficient vector $\vec{\xi}$ improves the success rate situation as the success area increases. The BMO approach is as flexible as correlated and directed mutation, but is less computational expensive than both methods. In comparison to correlated mutation $\frac{N\cdot(N-1)}{2}$ additional strategy parameters can be saved. Furthermore, the rotation of the mutation ellipsoid which demands $O(N^2)$ steps can be saved. Concerning the directed mutation the computation of the random numbers of an asymmetric probability density function is usually more computationally expensive than the computation of Gaussian random numbers. Especially, in practise the implementation is less complex. Figure 4 shows a comparison of the different probability density functions of the three mentioned approaches.

5 The BMO on constrained numerical problems

For many optimization problems the search space is constrained due to a variety of practical conditions. For evolutionary algorithms with a self-adaptive step size mechanism like evolution strategies, it is not easy to find an optimum which lies on the boundary of the feasible search space due to premature step size reduction. This results in premature fitness stagnation before approximating the optimum, also see [14]. For big step sizes the constrained search space cuts off a big area of success to reproduce better mutations than the parent. So the step sizes reduce self-adaptively before reaching the area of the optimum.

5.1 Experimental results

First of all we show the behavior of a $(\mu \overset{+}{,} \lambda)$-ES with standard uncorrelated mutation operator on our four selected constrained test problems, see the appendix A. The experimental conditions are shown in table 1. We follow the recommendation for the parameters τ_0 and τ_1

$$\tau_0 = \frac{1}{\sqrt{2N}} \quad \text{and} \quad (20)$$

$$\tau_1 = \frac{1}{\sqrt{2\sqrt{N}}} \quad (21)$$

Intermediate recombination was used for objective and strategy variables. As constraint handling method the simple death penalty, i.e. the rejection of infeasible solutions, was used. Table 2 shows the results from 100 runs. The experiments show that the evolution strategies with uncorrelated standard mutation are not able to approximate any of the optima of the constrained problems within the given number of generations. Instead, the evolution strategies suffer from fitness stagnation before reaching the optimum.

problem	ES-type	γ	generations	runs
2.40	(15,300)	0.1	1000	100
2.41	(15,300)	0.1	500	100
g04	(15,100)	0.1	200	100
g09	(15,100)	0.1	500	100

Table 1: Parameter settings for the experiments on the test problems, the parameter γ is used only for the BMO.

The experimental results of our *biased mutation operator (BMO)* on the constrained test problems are presented in table 3. The results show that due to the BMO approach the ES are capable of approximating the optimum in every run on the three problems 2.40, 2.41 and HB. Experiments with different settings for the mutation strength parameter γ for the mutation of the bias coefficient vector $\vec{\xi}$ showed that the setting $\gamma = 0.1$ is a reasonable recommendation. The experiments on problem g09 show no significant improvement. Here the BMO is not capable of guiding the search into a more beneficial direction to prevent fitness stagnation. Moreover, we conduct one-tailed t-test to examine whether or not there exists statistically significant improvement of the BMO over uncorrelated standard mutation in solution quality. Table 4 shows the t-test results of the three of the four experiments: The extremely small p-values ($\ll 0.05$) in this table show that the BMO can improve the solution quality over standard uncorrelated

problem	optimum	best	mean	worst	std.dev
2.40	-5000.00	-4999.74	-4911.51	-4691.61	3.53
2.41	-17857.14	-17857.12	-17340.42	-15912.11	2.66
g04	-30665.539	-30665.539	-30660.15	-30631.48	0.54
g09	680.630	680.63	680.66	680.79	0.00279

Table 2: Experimental results of the evolution strategies (see table 1) with uncorrelated standard mutation and N step sizes on the four constrained test problems.

problem	optimum	best	mean	worst	std.dev
2.40	-5000.00	-5000.00	-5000.00	-5000.00	$1.4 \cdot 10^{-10}$
2.41	-17857.14	-17857.14	-17857.14	-17857.14	$3.3 \cdot 10^{-12}$
g04	-30665.539	-30665.54	-30665.54	-30665.49	$3.2 \cdot 10^{-4}$
g09	680.630	680.6310	680.67	680.78	0.001599

Table 3: Experimental results of the evolution strategies (see table 1) with the BMO on the constrained test problems.

mutation significantly. This fact validates the effectiveness of the BMO on these problems. However, no improvement is achieved on problem g09. Here the BMO is obviously not capable of improving the success rate by biasing the mutation self-adaptively. Further experiments showed that both the correlated and the directed mutation are capable of approximating the optimum of the constrained problems similar to the BMO.

problem	t-value	degree of freedom	p-value
2.40	250.67989	99	$6.912 \cdot 10^{-141}$
2.41	1942.5564	99	$6.861 \cdot 10^{-229}$
g04	99.814815	99	$1.7996 \cdot 10^{-101}$

Table 4: The t-test verifies that the BMO is superior to standard uncorrelated mutation on three of the constrained problems.

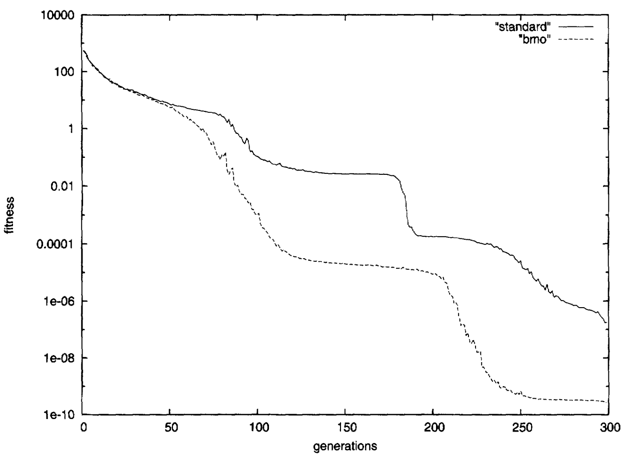

Figure 5: Fitness development of an (15,100)-ES with uncorrelated mutation with N steps sizes (standard) and with the BMO (bmo) on problem g04. On the y-axis the development of the difference between fitness and the optimum is shown on a logarithmic scale.

5.2 Convergence Speed

Figure 5 shows the fitness development of 20 runs of a (15,100)-ES on problem g04 with uncorrelated mutation (N step sizes) and with the BMO. On the y-axis the difference between the fitness in each generation and the optimum is visualized on a logarithmic scale. As the figure shows, the BMO converges faster than the standard approach. Over the whole run the BMO-fitness develops faster than the fitness of the standard mutation. For both approaches the fitness does not develop logarithmically linear, but comprises phases of slower convergence.

6 Conclusions and future work

We have introduced a new biased mutation operator (BMO) which is based on shifting the mutation ellipsoid self-adaptively to improve the success rate for reproducing better offspring. The BMO approach can be as flexible as correlated mutation and directed mutation, but is less computationally expensive. The adaptation process of the bias of mutation is more efficient than the adaptation of the correlated mutation angles because only N additional endogenous strategy variables are necessary. In comparison to directed mutation, the computational effort of calculating an asymmetric probability density function is not necessary.

Experimental results showed that the BMO approach is helpful to improve the ability of convergence. Further investigation is necessary to evaluate the efficiency of convergence on constrained and multimodal problems as well.

Bibliography

[1] J. C. Bean and A. B. Hadj-Alouane. A Dual Genetic Algorithm for Bounded Integer Programs. Technical Report TR 92-53, Department of Industrial and Operations Engineering, The University of Michigan, 1992.

[2] S. V. Belur. CORE: Constrained Optimization by Random Evolution. In J. R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference,*

pages 280–286, Stanford University, California, Juli 1997. Stanford Bookstore.

[3] S. Berlik. A step size preserving directed mutation operator. In *Genetic and Evolutionary Computation GECCO 2004: Genetic and Evolutionary Computation Conference*, pages 26–30. Springer, 2004.

[4] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer, Heidelberg, 2001.

[5] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[6] C. A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.

[7] C. A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.

[8] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[9] L. Hildebrand, B. Reusch, and M. Fathi. Directed mutation - a new selfadaptation for evolution strategies. In *Proceedings of Congress on Evolutionary Computation CEC99*, pages 289–294. The MIT Press, Februar 1996.

[10] A. Homaifar, S. H. Y. Lai, and X. Qi. Constrained Optimization via Genetic Algorithms. *Simulation*, 62(4):242–254, 1994.

[11] F. Jiménez and J. L. Verdegay. Evolutionary techniques for constrained optimization problems. In H.-J. Zimmermann, editor, *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, 1999. Verlag Mainz.

[12] J. Joines and C. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In D. Fogel, editor, *Proceedings of the first IEEE Conference on Evolutionary Computation*, pages 579–584, Orlando, Florida, 1994. IEEE Press.

[13] S. Koziel and Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, *7(1):19–44*, 1999.

[14] O. Kramer and H.-P. Schwefel. On three new approaches to handle constraints within evolution strategies (submitted). *Natural Computing*, 2005.

[15] A. Kuri-Morales and C. V. Quezada. A Universal Eclectic Genetic Algorithm for Constrained Optimization. In *Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98*, pages 518–522, Aachen, September 1998. Verlag Mainz.

[16] Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer-Verlag, 2000.

[17] E. M. Montes and C. A. Coello Coello. A simple multi-membered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17, February 2005.

[18] A. Ostermeier, A. Gawelczyk, and N. Hansen. A derandomized approach to self adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380, 1994.

[19] M. Schoenauer and S. Xanthakis. Constrained GA Optimization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, pages 573–580, San Mateo, California, Juli 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[20] H.-P. Schwefel. Adaptive Mechanismen in der biologischen Evolution und ihr Einfluss auf die Evolutionsgeschwindigkeit. Internal report of the group Bionik und Evolutionstechnik, Institut für Mess- und Regelungstechnik Re 215/3, Technical University of Berlin, July 1974.

[21] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley Interscience, New York, 1995.

[22] P. D. Surry, N. J. Radcliffe, and I. D. Boyd. A Multi-Objective Approach to Constrained Optimisation of Gas Supply Networks : The COMOGA Method. In T. C. Fogarty, editor, *Evolutionary Computing. AISB Workshop. Selected Papers*, Lecture Notes in Computer Science, pages 166–180. Springer-Verlag, Sheffield, U.K., 1995.

[23] X. Yao and Y. Liu. Fast evolution strategies. In P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, editors, *Evolutionary Programming VI*, pages 151–161, Berlin, 1997. Springer.

A Test problems

Problem 2.40 - Schwefel's problem 2.40 [21]

Minimize:

$$f(x) = -\sum_{i=1}^{5} x_i$$

Constraints:

$$g_j(x) = \begin{cases} x_j \geq 0, & \text{for } j = 1, \ldots, 5 \\ -\sum_{i=1}^{5} (9+i)x_i + 50000 \geq 0, & \text{for } j = 6 \end{cases}$$

Minimum:

$$x^* = (5000, 0, 0, 0, 0)^T$$

$$f(x^*) = -5000$$

g_2 bis g_6 active.

Problem 2.41 - Schwefel's problem 2.41 [21]
Minimize:

$$F(\vec{x}) = -\sum_{i=1}^{5}(ix_i)$$

Constraints like problem 2.40.
Minimum:

$$x^* = (0, 0, 0, 0, \frac{50000}{14})^T$$

$$f(x^*) = -\frac{250000}{14}$$

g_j active for $j = 1, 2, 3, 4, 6$.

Problem g04 - Himmelblau's nonlinear optimization problem. Minimize:

$$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Constraints:

$$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5$$

$$g_2(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$$

$$g_3(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4$$

$$0 \le g_1(\vec{x}) \le 92$$

$$90 \le g_2(\vec{x}) \le 110$$

$$20 \le g_3(\vec{x}) \le 25$$

$$78 \le x_1 \le 102$$

$$33 \le x_2 \le 45$$

$$27 \le x_i \le 45 \quad (i = 3, 4, 5)$$

Optimum:

$$x^* = (78.000, 33.000, 29.995, 45.000, 36.776)^T$$

$$f(x^*) = -30665.539$$

Problem g09 Minimize:

$$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2$$
$$+10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

Constraints:

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$$

$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$$

$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$$

$$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$$

$$x^* = (78.000, 33.000, 29.995, 45.000, 36.776)^T$$

$$-10 \le x_i \le 10 (i = 1, ..., 7)$$

$$x^* = (2.330499, 1.951372, -0, 4775414, 4.365726,$$
$$-0.6244870, 1.038131, 1.594227)$$

$$f(x^*) = 680.630$$