Evolution Strategy with Neighborhood Attraction – A Robust Evolution Strategy

Jutta Huhse
University of Tübingen, WSI-RA
Sand 1, D-72076 Tübingen, Germany
huhse@informatik.uni-tuebingen.de

Andreas Zell
University of Tübingen, WSI-RA
Sand 1, D-72076 Tübingen, Germany
zell@informatik.uni-tuebingen.de

Abstract

The evolution strategy with neighborhood attraction (EN) is a new combination of self-organizing maps (SOM) and evolution strategies (ES). It adapts the neighborhood relationship known from SOM to ES individuals in order to concentrate them around the optimum of the problem.

In this paper, detailed investigations on the robustness of the EN were performed on a variety of well-known optimization problems. The behavior of the EN was compared to that of several other known variants of ES such as ES with mutative step control, ES with covariance matrix adaptation, differential evolution and others. In this test series it was shown that EN is much more robust than the other ES variants.

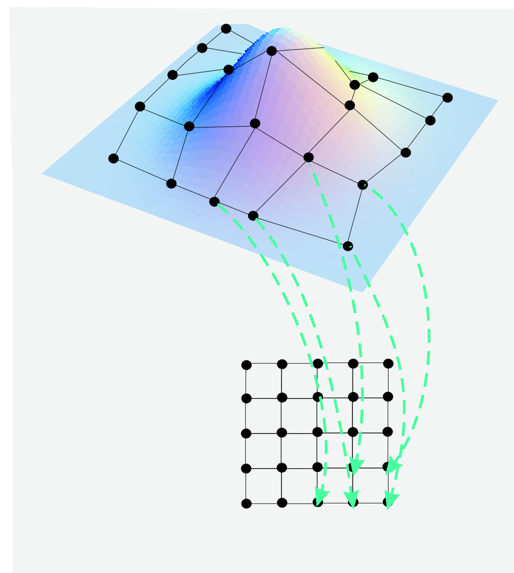

Figure 1: EN: Transfer of the SOM neighborhood onto ES individuals

1 INTRODUCTION

Evolution strategies with **N**eighborhood attraction (EN) are a combination of two different kinds of problem solvers: Evolution strategies (ES) and artificial neural networks, i.e. self-organizing maps (SOM), to be more precise.

ES were developed in the late 1960s by Rechenberg and Schwefel and later improved (see [Rechenberg, 1994], [Schwefel, 1995] and [Bäck et al., 1997]). Their main application is the optimization of real-valued multi-parameter problems. They directly use the information of the quality of a potential solution of the function to be optimized. ES work on a population P of potential solutions (individuals a) by manipulating these individuals with evolutionary operators.

A special class of neural networks - the self-organizing maps (SOM) - were developed in the 1980s by Kohonen [Kohonen, 1995]. The neurons of a SOM are organized in a neighborhood relationship, e.g. a two-dimensional grid. Learning takes place by adapting the weight vectors of the neurons (and thereby the neurons' positions in the problem space) according to a learning rule which incorporates the neighborhood relation defined among the neurons.

The idea behind EN is to transfer the neighborhood and the learning rule defined for SOM neurons onto the individuals of an ES (see fig. 1). Using this neighborhood concept in the new EN, better EN individuals can attract their worse neighbors and thus, the individuals will be concentrated around the optimum.

Previous benchmark tests were performed on a number of optimization tasks which could be solved by EN as well as by many conventional ES (cf. [Huhse and Zell, 2000]). The main focus was on the convergence velocity of the EN, and it could be shown that – especially for small populations – the performance of the EN is equivalent to or even better than comparable conventional ES on those benchmark problems.

This paper investigates the robustness of the EN. The focus lies on difficult optimization tasks which often cause problems to conventional ES. A test bed of many difficult optimization tasks was set up, and the reliability of the EN is compared to several different variants of ES like ES with mutative step control, ES with covariance matrix adaptation, differential evolution, and others. The experiments show that the EN has an apparently better robustness than the other ES variants on most of the tested optimization tasks.

A short description of the EN is given in section 2. The optimization problems used as a test bed for our investigations are described in section 3. Section 4 shows the test series that were performed on EN and the ES variants, and the results are discussed in section 5. Detailed information on the test functions can be found in the appendix.

2 EVOLUTION STRATEGY WITH NEIGHBORHOOD ATTRACTION

The individuals which are unordered in conventional ES, have neighborhood relations in the EN. The neighborhood between the μ parent individuals is constituted by arranging them in an orthogonal, elastic grid. As known from SOM, each individual can be identified by its fixed grid position, and two individuals are neighbors if they are directly connected on the grid (see fig. 2, left).

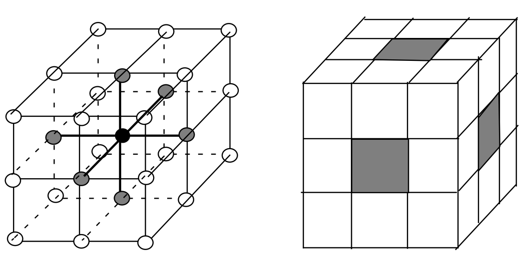

Figure 2: Left: Neighborhood grid between the parent individuals; all grey individuals are neighbors of the black individual.
Right: Division of the problem space into hyper-cubes; during initialization one individual is placed randomly into each hyper-cube

The dimension d_g of the grid depends on the dimension d_p of the problem space and on the number μ of the parent individuals (i.e. the number of the individuals constituting the grid). The grid is calculated in the following way: First, μ is divided into its prime factors; e.g. $\mu = 100 \Rightarrow 2 \cdot 2 \cdot 5 \cdot 5$; $n_f = 4$ is the number of the factors f_i. If the number n_f of the factors is smaller than the dimension of the problem, then the grid dimension is set to n_f. Otherwise, the smallest primes are multiplied until the number of factors is equal to the problem dimension. Thus, it holds: $d_g \leq d_p$. Inside the neighborhood grid the individuals are arranged according to the factorization. E.g. for $\mu = 100$ and a problem dimension $d_p = 10$, the grid dimension is $d_g = 4$ and $f_1 = 2$ individuals are in the first dimension, $f_2 = 2$ individuals in the second, $f_3 = 5$ individuals in the third and $f_4 = 5$ individuals in the fourth.

Because of the orthogonality of the grid the neighborhood is easily determined. The left and the right neighbor of one individual a_i can be determined independently for each dimension. E.g. for dimension $d = 1$ the two neighbors of a_i with the grid coordinates

$$\begin{aligned} a_i &= (a_0,\ a_1,\ \ a_2, \ldots, a_n) \quad \text{are} \\ a_{N_l} &= (a_0, a_1 - 1, a_2, \ldots, a_n) \quad \text{and} \\ a_{N_r} &= (a_0, a_1 + 1, a_2, \ldots, a_n). \end{aligned}$$

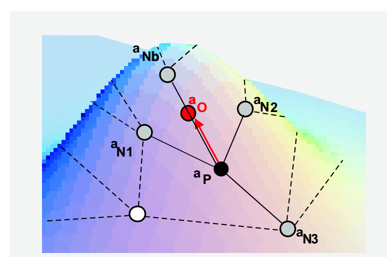

Figure 3: Neighborhood attraction in EN

In contrast to conventional ES and SOM, the initial values of the object variables of the EN individuals are not assigned randomly. Rather, the problem space is divided into equally sized hyper-cubes, each of them corresponding to one grid position (see fig. 2, right). The object variables of the associated individual are initialized with equally distributed random values within the ranges of its hyper-cube.

As is customary in ES, the EN individuals are evaluated using the fitness function.

The EN-specific evolutionary operator – the neighborhood attraction – manipulates the EN individuals according to one learning step in a SOM. Every parent individual a_P is attracted to its best neighbor a_{Nb} and thus becomes the offspring a_O (see figure 3). The object variables $\vec{x}_O$ of the offspring are calculated according to equation 1 and the neighborhood relations are retained unchanged.

$$\vec{x}_O = \vec{x}_P + \delta \cdot (\vec{x}_{Nb} - \vec{x}_P) \quad (1)$$

Here, $\vec{x}_P$ is the object variables vector of the parent and $\vec{x}_{Nb}$ is its best neighbor. The parameter δ defines the strength of the attraction along the difference vector and $\vec{x}_O$ denotes the object variables of the offspring.

If the parent individual a_P is considered better than all its neighbors a_{Nj} ($j = 1 \ldots g$, g is the number of neighbors) a "simple conventional" mutation (referred to as *ES-mutation* here) is performed. λ offsprings are generated according to 2.

$$\begin{array}{rcllcl} \vec{v}_{mut,l} & = & \vec{\mathcal{N}}(0,1) & l & = & 1 \ldots \lambda \\ d_{min} & = & \min(\|\vec{x}_P - \vec{x}_{Nj}\|) & j & = & 1 \ldots g \\ s_{eff} & = & \frac{1}{n} d_{min} & & & \\ \vec{x}_{O_i} & = & \vec{x}_P + s_{eff} \cdot \vec{v}_{mut_i} & i & = & 1 \ldots n \end{array} \quad (2)$$

n is the number of object variables.

The effective step size s_{eff} is determined by the reciprocal number of object variables [1] and by the distance d_{min} to the nearest neighbor. Thus, a mutation which jumps over a neighbor and an entanglement of the grid becomes less likely. During the contraction of the grid the effective step sizes decrease due to the influence of d_{min}.

Recombination is not explicitly used.

For details, please see [Huhse and Zell, 2000].

3 TEST FUNCTIONS

An extensive test bed of optimization tasks was constituted to permit thorough investigations on the robustness of the EN.

On the one hand, the functions used for previous test series [Huhse and Zell, 2000] were incorporated (f_1, f_2, f_6, f_9, f_{15}, f_{21}). These functions include uni-modal and multi-modal functions as well as symmetric and non-symmetric ones, and they were also used e.g. in [de Jong, 1975], [Bäck, 1992] and [Schwefel, 1995]. On most of these test functions almost all ES variants and also the EN converged.

On the other hand, special focus was set on very difficult test functions which are known to cause problems to many optimization tasks. E.g. f_5 (Shekel's foxholes) consists of a wide plateau with many steep and narrow holes embedded as local minima, where the individuals might get caught. f_{23} [Galar, 1991] consists of a plateau with one local maximum, connected to the global maximum by a single saddle. The varying dimensions of each test function are indicated in the appendix.

- f_1: Sphere model
- f_2: Generalized Rosenbrock's function
- f_5: Shekel's foxholes
- f_6: Schwefel's double sum
- f_9: Ackley's function
- f_{15}: Weighted sphere model
- f_{17}: Fletcher and Powell
- f_{18}: Shekel-5
- f_{19}: Shekel-7
- f_{20}: Shekel-10
- f_{21}: Griewangk
- f_{23}: Galar
- f_{24}: Kowalik

4 TEST SERIES

For each function of the test bed the EN was compared to the following ES variants:

- ES with uncorrelated self adaptation (uncorrelated)
- ES with covariance matrix adaptation (CMA) according to [Hansen and Ostermeier, 1996]
- ES with mutative step control (named MSR by [Rechenberg, 1994])
- ES with derandomized self adaptation (derand) [Ostermeier et al., 1993]
- ES without self adaptation (off)

[1] according to [Rechenberg, 1994], who proposes for his basic algorithm for a $(1 \overset{+}{,} \lambda)$ -ES to make the length of the mutation vector independent of the number n of variables by generating the normally distributed vector elements with the mean zero and with the variance $\sigma = \frac{1}{\sqrt{n}}$

- ES with self adaptation adopted from differential evolution (diffevol) [Price and Storn, 1995]

The following parameter settings were used for the EN: Size of the individual grid $\mu = 100$, attraction factor $\delta = 0.0011$, and the number of offsprings per parent generated during *ES-mutation* $\lambda = 2$.

For all ES variants, a $(10, 100)$-strategy without recombination was used. These settings were chosen because they are known to be practicable for many optimization tasks. Furthermore $\lambda = 100$ corresponds to the grid size of the EN, which means, that the number of function evaluations which are calculated in one generation of EN corresponds to that of one generation of ES.

We developed a special EN simulation program to perform the test series. For the comparision tests we used *EvA*, a simulation program for Evolutionary Algorithms which was developed in the same group [Wakunda and Zell, 1997].

A simulation run was stopped when the convergence value was reached or when the algorithm stagnated. Focus was not on the number of function evaluations but on the best function value reached. Every run was repeated 30 times with different random seeds. Then the best fitness values reached were averaged out and the standard deviation was calculated. The graphical representations below show for the different strategies (abscissa) the average of the best fitness values and the standard deviation added to and subtracted from that average (ordinate). [2] For clarity, the optimum of each test function is plotted as a thin line.

Not all test series can be represented graphically here. For the test functions f_1, f_6, f_9, f_{15} almost all strategies were equally reliable in finding almost always the optimum. Only the differential evolution ES had some problems.

The more interesting results for the difficult functions which could not be solved by some strategies are shown below:

For function f_5, only EN was totally reliable (fig. 4). All other strategies were frequently trapped in one of the local optima.

For function f_{17}, most strategies achieved good results (fig. 5). Uncorrelated ES, CMA-ES and derandomized ES found the optimum with only small variance, the other variants had some more problems. Again, only EN showed no problems in the optimization.

[2] Only one standard deviation was calculated for better and worse results. Note: The subtracted standard deviation does not imply that there were results better than the optimum. It is only shown to facilitate the comparision of very similar results, like CMA, derand, and EN in fig. 5.

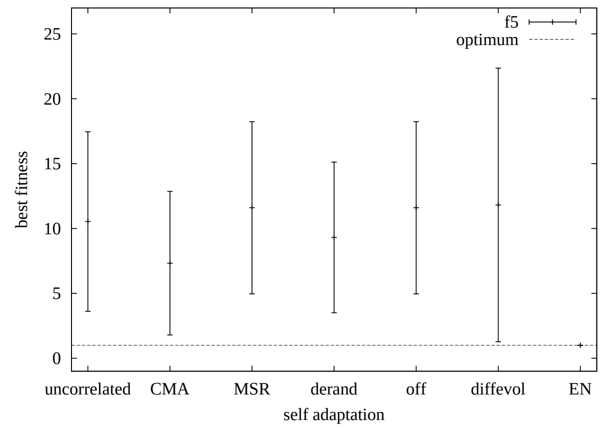

Figure 4: Function f_5, $\min(f_5(\vec{x})) = 0$

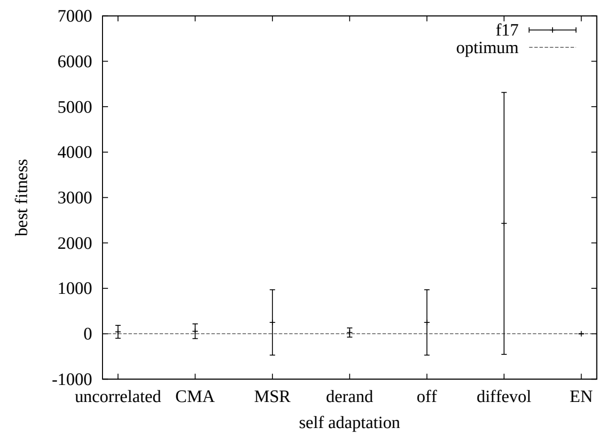

Figure 5: Function f_{17}, $\min(f_{17}(\vec{x})) = 0$

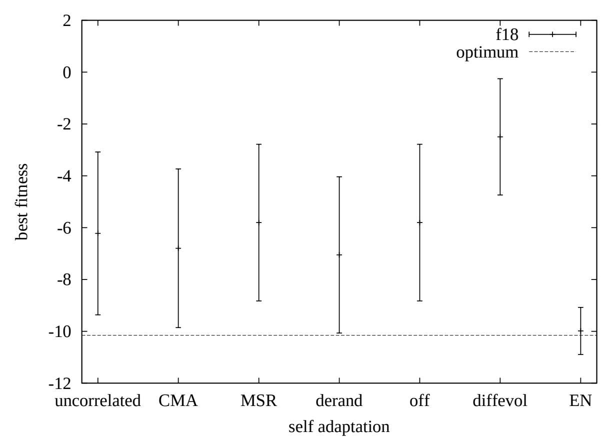

Figure 6: Function f_{18}, $\min(f_{18}(\vec{x})) = -10.1532$

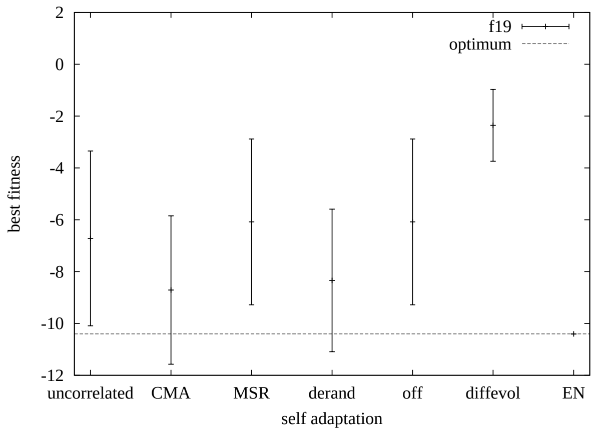

Figure 7: Function f_{19}, $\min(f_{19}(\vec{x})) = -10.4029$

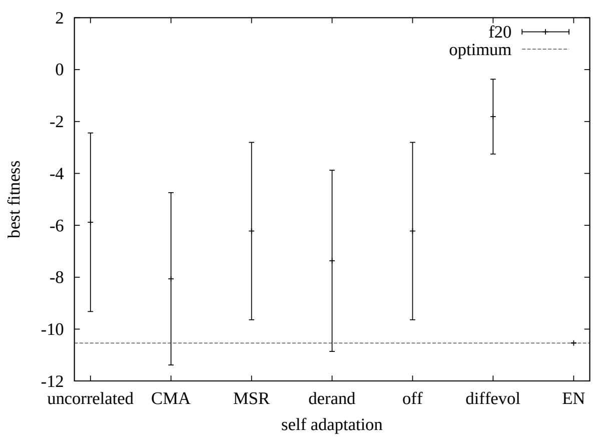

Figure 8: Function f_{20}, $\min(f_{20}(\vec{x})) = -10.5364$

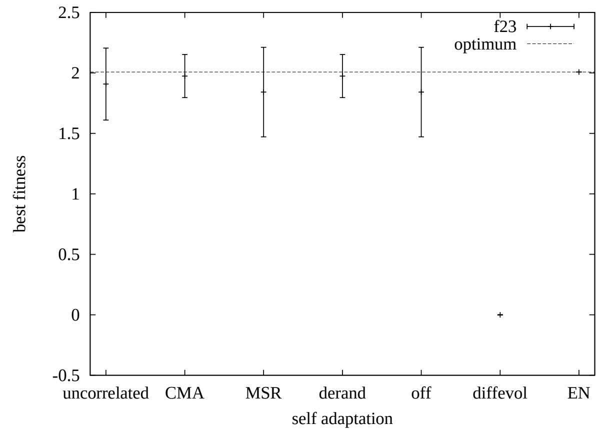

Figure 9: Function f_{23}, $\max(f_{23}(\vec{x})) = 2.00686$

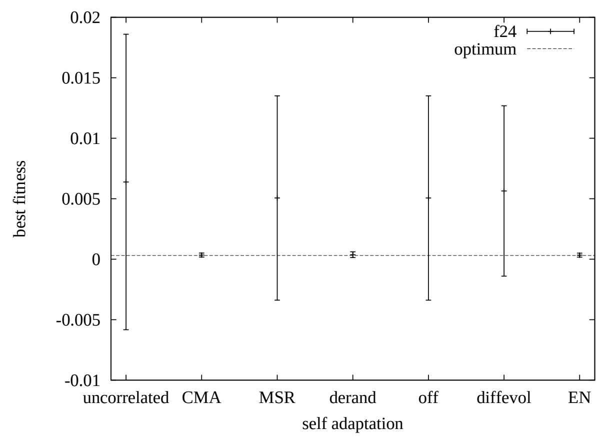

Figure 10: Function f_{24}, $\min(f_{24}(\vec{x})) = 0.0003075$

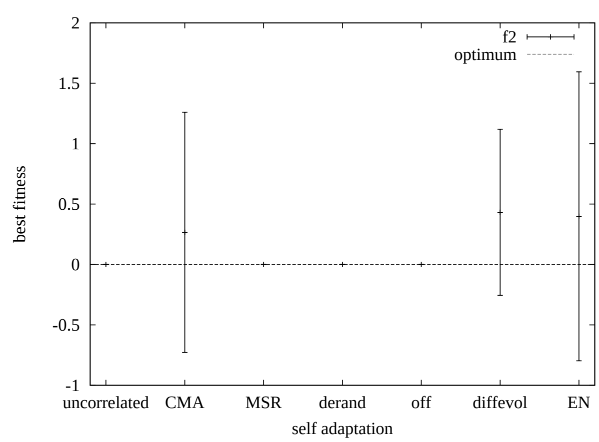

Figure 11: Function f_2, $\min(f_2(\vec{x})) = 0$

Also for f_{18}, EN is the most robust strategy (fig. 6). It stagnated only once, and compared to the other strategies, EN was clearly the best.

The test series for f_{19} and f_{20} led to similar results (fig. 7, fig. 8): EN was the only strategy which was able to always find the optimum. All other strategies stagnated repeatedly.

The same holds for the maximization problem f_{23} (fig. 9). EN always found the global optimum, while the other strategies often climbed on the local maximum, and one strategy (differential evolution ES) did not even leave the plateau from where the search started.

The test series with function f_{24} shows varying results (fig. 10). The best strategies are EN, CMA-ES, and derandomized ES. CMA-ES and EN are almost equal, derandomized ES performs a bit worse.

F_2 is the only test function where EN was outperformed by other strategies (fig. 11). While uncorrelated ES, MSR-ES, derandomized ES and even ES without self adaptation (off) converged always, EN had problems as well as CMA-ES and differential evo-

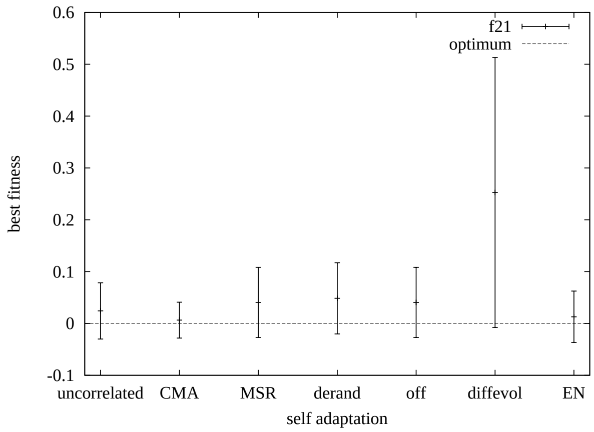

Figure 12: Function f_{21}, $\min(f_{21}(\vec{x})) = 0$

lution ES.

F_{21} is a quite difficult test function which could not be optimized reliably by any strategy (fig. 12). Comparing the averages of the best function values found, EN is the second best strategy after CMA-ES.

5 CONCLUSIONS

The robustness of the new EN strategy – Evolution strategy with Neighborhood attraction – was investigated in exhaustive test series using a large test bed of optimization tasks and many ES variations for comparision.

Only for one of the thirteen test series, some of the ES variants were more robust than EN, i.e. other strategies were able to optimize the test function more often than EN. For all test series, the EN was able to find the optimum in at least 90% of the test runs. For most of the test series, the EN converged always (for all seeds), and for many optimization tasks, EN was the only strategy that was able to converge always, while all other ES variants repeatedly stagnated in local optima.

It could be shown that the EN is much more robust than other ES-variants, especially for difficult, multimodal functions.

For further work it is conceivable to incorporate the mechanism of EN into other existing, elaborated ES to improve these. A combination of e.g. CMA and EN could be quite promising.

A TEST FUNCTIONS

A.1 f_1: Sphere model

[de Jong, 1975]

$$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$$

$$-5.12 \le x_i \le 5.12\,; \quad dim = 10$$

$$\min(f_1) = f_1(0,\ldots,0) = 0$$

A.2 f_2: Generalized Rosenbrock's function

[de Jong, 1975]

$$f_2(\vec{x}) = \sum_{i=1}^{n-1}(100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

$$-5.12 \le x_i \le 5.12\,; \quad dim = 10$$

$$\min(f_2) = f_2(1,\ldots,1) = 0$$

A.3 f_5: Shekel's foxholes

[de Jong, 1975]

$$\frac{1}{f_5(\vec{x})} = \frac{1}{K} + \sum_{j=1}^{25} \frac{1}{c_j + \sum_{i=1}^{2}(x_i - a_{ij})^6}$$

$$(a_{ij}) =$$

$$\begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32 \end{pmatrix}$$

$$K = 500 \quad f_5(a_{1j}, a_{2j}) \approx c_j = j$$

$$-65.536 \le x_i \le 65.536\,; \quad dim = 2$$

$$\min(f_5) = f_5(-32,-32) \approx 0.998004$$

A.4 f_6: Schwefel's double sum

[Schwefel, 1981, Schwefel, 1995] (function 1.2)

$$f_6(\vec{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2 = x^{\mathrm{T}}\mathbf{A}x$$

$$-65.536 \le x_i \le 65.536\,; \quad dim = 10$$

$$\min(f_6) = f_6(0,\ldots,0) = 0$$

A.5 f_9: Ackley's function

[Ackley, 1987]

$$f_9(\vec{x}) = -a \cdot \exp\left(-b\sqrt{\tfrac{1}{n} \cdot \sum_{i=1}^{n} x_i^2}\right)$$

$$- \exp\left(\tfrac{1}{n} \cdot \sum_{i=1}^{n} \cos(c \cdot x_i)\right) + a + e$$

$$a = 20\,; \quad b = 0.2\,; \quad c = 2\pi$$

$$-32.768 \le x_i \le 32.768\,; \quad dim = 10$$

$$\min(f_9) = f_9(0,\ldots,0) = 0$$

A.6 f_{15}: Weighted sphere model

[Schwefel, 1988]

$f_{15}(\vec{x}) = \sum_{i=1}^{n} i \cdot x_i^2$

$-5.12 \le x_i \le 5.12\,; \quad dim = 10$

$\min(f_{15}) = f_{15}(0,\ldots,0) = 0$

A.7 f_{17}: Fletcher and Powell

[Fletcher and Powell, 1963]

$f_{17}(\vec{x}) = \sum_{i=1}^{n} (A_i - B_i(x))^2$

$A_i = \sum_{j=1}^{n} (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j)$

$B_i = \sum_{j=1}^{n} (a_{ij} \sin x_j + b_{ij} \cos x_j)$

$a_{ij}, b_{ij} \in [-100, 100]$ (equ. distr. randoms)

$\alpha_j \in [-\pi, \pi]$ (equ. distr. randoms)

$-\pi \le x_i \le \pi\,; \quad dim = 4$

$\min(f_{17}) = f_{17}(\alpha_1, \alpha_2, \ldots, \alpha_n) = 0$

A.8 f_{18}: Shekel-5

[Törn and Žilinskas, 1989]

$f_{18}(\vec{x}) = -\sum_{i=1}^{m} \frac{1}{(\vec{x}-A(i))(\vec{x}-A(i))^T + c_i}$

$m = 5$

$0 \le x_i \le 10\,; \quad dim = 4$

i	$A(i)$				c_i	$f_{18}(\vec{x} = A(i))$
1	4	4	4	4	0.1	-10.1532
2	1	1	1	1	0.2	-5.0552
3	8	8	8	8	0.2	-5.10076
4	6	6	6	6	0.4	-2.68284
5	3	7	3	7	0.4	-2.6304

$\min(f_{18}) = f_{18}(0,\ldots,0) = -10.1532$

A.9 f_{19}: Shekel-7

[Törn and Žilinskas, 1989]

$f_{19}(\vec{x}) = -\sum_{i=1}^{m} \frac{1}{(\vec{x}-A(i))(\vec{x}-A(i))^T + c_i}$

$m = 7$

$0 \le x_i \le 10\,; \quad dim = 4$

i	$A(i)$				c_i	$f_{19}(\vec{x} = A(i))$
1	4	4	4	4	0.1	-10.4028
2	1	1	1	1	0.2	-5.08767
3	8	8	8	8	0.2	-5.1288
4	6	6	6	6	0.4	-2.75186
5	3	7	3	7	0.4	-2.76589
6	2	9	2	9	0.6	-1.83708
7	5	5	3	3	0.3	-3.72275

$\min(f_{19}) = f_{19}(0,\ldots,0) = -10.4029$

A.10 f_{20}: Shekel-10

[Törn and Žilinskas, 1989]

$f_{20}(\vec{x}) = -\sum_{i=1}^{m} \frac{1}{(\vec{x}-A(i))(\vec{x}-A(i))^T + c_i}$

$m = 10$

$0 \le x_i \le 10\,; \quad dim = 4$

i	$A(i)$				c_i	$f_{20}(\vec{x} = A(i))$
1	4	4	4	4	0.1	-10.5363
2	1	1	1	1	0.2	-5.12847
3	8	8	8	8	0.2	-5.17562
4	6	6	6	6	0.4	-2.871
5	3	7	3	7	0.4	-2.80662
6	2	9	2	9	0.6	-1.85892
7	5	5	3	3	0.3	-3.83364
8	8	1	8	1	0.7	-1.67525
9	6	2	6	2	0.5	-2.42083
10	7	3.6	7	3.6	0.5	-2.42652

$\min(f_{20}) = f_{20}(0,\ldots,0) = -10.5364$

A.11 f_{21}: Griewangk

[Törn and Žilinskas, 1989]

$f_{21}(\vec{x}) = 1 + \frac{1}{d} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$

$d = 200$

$-100.0 \le x_i \le 100.0\,; \quad dim = 10$

$\min(f_{21}) = f_{21}(0,\ldots,0) = 0$

A.12 f_{23}: Galar

[Galar, 1991]

$f_{23}(\vec{x}) = (\exp(-5x_1^2) + 2\exp(-5(1-x_1)^2))$

$\cdot \exp\left(-5 \sum_{i=2}^{n} x_i^2\right)$

$-5.0 \le x_i \le 5.0\,; \quad dim = 10$

$\min(f_{23}) = f_{23}(0.9965, 0, \ldots, 0) \approx 2.00686$

A.13 f_{24}: Kowalik

[Schwefel, 1977, Schwefel, 1995]

$$f_{24}(\vec{x}) = \sum_{i=1}^{11} \left(a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4} \right)^2$$

$$-5.0 \leq x_i \leq 5.0\,; \quad dim = 4$$

i	a_i	b_i^{-1}
1	0.1957	0.25
2	0.1947	0.5
3	0.1735	1
4	0.1600	2
5	0.0844	4
6	0.0627	6
7	0.0456	8
8	0.0342	10
9	0.0323	12
10	0.0235	14
11	0.0246	16

$$\min(f_{24}) \approx f_{24}(0.1928, 0.1908, 0.1231, 0.1358) \approx 0.0003075$$

References

[Ackley, 1987] Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing.* Kluwer Academic Publishers, Boston.

[Bäck, 1992] Bäck, T. (1992). A user's guide to genesys 1.0. Technical report, University of Dortmund, Department of Computer Science

[Bäck et al., 1997] Bäck, T., Hammel, U., and Schwefel, H.-P. (1997). Evolutionary computation. Comments on the history and current state. In *IEEE Transactions on Evolutionary Computation*, volume 1, pages 3–17.

[de Jong, 1975] de Jong, K. (1975). An analysis of the behaviour of a class of genetic adaptive systems. Master's thesis, University of Michigan.

[Fletcher and Powell, 1963] Fletcher, R. and Powell, M. J. D. (1963). A rapidly convergent descent method for minimization. *Comp. J. 6*, pages 163–168.

[Galar, 1991] Galar, R. (1991). Simulation of local evolutionary dynamics of small populations. *Biological Cybernetics*, 65:37–45.

[Hansen and Ostermeier, 1996] Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adatation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC '96)*, pages 312–317, Nagoya, Japan.

[Huhse and Zell, 2000] Huhse, J. and Zell, A. (2000). Evolution strategy with neighborhood attraction. In *Proceedings of the Second ICSC Symposium on Neural Computation (NC'2000)*, pages 363–369.

[Kohonen, 1995] Kohonen, T. (1995). *Self-Organizing Maps.* Springer Verlag, Berlin.

[Ostermeier et al., 1993] Ostermeier, A., Gawelczyk, A., and Hansen, N. (1993). A derandomized approach to self adaptation of evolution strategies. Technical report, Technische Universität Berlin.

[Price and Storn, 1995] Price, K. and Storn, R. (1995). Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute (ICSI), University of California at Berkeley.

[Rechenberg, 1994] Rechenberg, I. (1994). *Evolutionsstrategie '94.* frommann-holzboog, Stuttgart.

[Schwefel, 1977] Schwefel, H.-P. (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, volume 26 of *Interdisciplinary systems research.* Birkhäuser, Basel.

[Schwefel, 1981] Schwefel, H.-P. (1981). *Numerical Optimization of Computer Models.* Wiley, Chichester.

[Schwefel, 1988] Schwefel, H.-P. (1988). Evolutionary learning optimum–seeking on parallel computer architectures. In Sydow, A., Tzafestas, S. G., and Vichnevetsky, R., editors, *Proceedings of the International Symposium on Systems Analysis and Simulation 1988, I: Theory and Foundations*, pages 217–225. Akademie der Wissenschaften der DDR, Akademie-Verlag, Berlin.

[Schwefel, 1995] Schwefel, H.-P. (1995). *Evolution and Optimum Seeking.* John Wiley and Sons, New York.

[Törn and Žilinskas, 1989] Törn, A. and Žilinskas, A. (1989). *Global Optimization*, volume 350 of *Lecture Notes in Computer Science.* Springer, Berlin.

[Wakunda and Zell, 1997] Wakunda, J. and Zell, A. (1997). Eva - a tool for optimization with evolutionary algorithms. In *Proceedings of the 23rd EUROMICRO Conference*, Budapest, Hungary.