# Model-Assisted Steady-State Evolution Strategies

Holger Ulmer, Felix Streichert, and Andreas Zell

Center for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72074 Tübingen, Germany,
{ulmerh,streiche,zell}@informatik.uni-tuebingen.de,
http://www-ra.informatik.uni-tuebingen.de

**Abstract.** The task of speeding up the optimization process on problems with very time consuming fitness functions is a central point in evolutionary computation. Applying models as a surrogate of the real fitness function is a quite popular idea. The performance of this approach is highly dependent on the frequency of how often the model is updated with data from new fitness evaluations. However, in generation based algorithms this is only done every $\lambda$-th fitness evaluation. To overcome this problem we use a steady-state strategy, which updates the model immediately after each fitness evaluation. We present a new model assisted steady-state Evolution Strategy (ES), which uses Radial-Basis-Function networks as a model. To support self-adaption in the steady-state algorithm a median selection scheme is applied. The convergence behavior of the new algorithm is examined with numerical results from extensive simulations on several high dimensional test functions. It achieves better results than standard ES, steady-state ES or model assisted ES.

## 1 INTRODUCTION

Evolution Strategies (ES) were developed in the late 60s by Rechenberg and Schwefel [10] [11]. They are known as excellent optimization tools for complex high dimensional multimodal real valued problems. However, they require a very high number of fitness function evaluations. In many real world applications, like high throughput material science or design optimization, the fitness evaluation is very expensive and time consuming. Therefore standard ES methods are not practical for such applications.

One approach to overcome this problem is the application of modeling techniques: To decrease the number of expensive fitness evaluations, approximation models, also known as metamodels, are used instead of the exact fitness function. These models are orders of magnitude cheaper to evaluate than the real fitness function.

The use of metamodeling techniques in evolutionary computation receives increasing attention [9] [7] [3] [4]. A survey on this research field can be found in [8].

E. Cant'u-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 610–621, 2003.
© Springer-Verlag Berlin Heidelberg 2003

The selection of an appropriate model to approximate the fitness function is a central point. Neural networks are widely used for function approximation and are therefore used for modeling in evolutionary optimization [7] [6]. Gaussian processing and kriging are statistical modeling techniques which are also used for modeling [9] [2] [3].
The coupling of the metamodel with the evolutionary optimization process controls how the optimization process is affected by replacing the expensive real fitness evaluation by the approximation of the model. The adaptive evolution control concept [6] [7] controls the impact of the model on the evolutionary optimization process. The Metamodel-Assisted Evolution Strategy (MA-ES) [4] uses the estimation of the model to preselect the most promising individuals before applying the expensive real fitness function. Another approach is to use the criterion confidence given by statistical models like kriging [4] or gaussian processing [3] to control the interaction of the metamodel with the evolutionary optimization process.
Our work aims at improving the coupling between the metamodel and the ES by using a steady-state strategy. The remainder of this paper is organized as follows: We first describe the modeling technique of Radial-Basis-Function (RBF) networks, which is used in this work as a surrogate for the expensive real fitness function, in section 2. Section 3 introduces the synthesis of the metamodel with a standard generation based ES. We propose the new Model Assisted Steady-State ES (MASS-ES) with median selection scheme in section 4.
Numerical results from extensive simulations on several high dimensional test functions are presented in section 5. The paper closes with a brief conclusion and outlook on future work.

## 2 RBF NETWORK MODEL

Consider a $d$-dimensional real valued problem with fitness function $f(\boldsymbol{x})$, which has to be minimized. As stated in the introduction, a model is needed to predict the fitness $\hat{f}(\boldsymbol{x})$ of an individual to save time consuming and expensive evaluations of the real fitness function $f(\boldsymbol{x})$.

We chose in our study Radial-Basis-Function (RBF) networks, which are known to be general and proper real valued function approximators. A comprehensive description of RBF-networks is given in [1].
The RBF network maps the objective variable $\boldsymbol{x}$ of an individual to its corresponding fitness value $\hat{f}(\boldsymbol{x})$. The network consists of an input layer of $d$ units (one for each input dimension), a single hidden layer of $h$ nonlinear processing units and an output layer of linear weights $\boldsymbol{\omega_i}$ (see figure 1).
The output $\hat{f}$ of the RBF network is given as a linear combination of a set of radial basis functions:

$$\hat{f}\,(x) = \omega_0 + \sum_{i=1}^{h} \omega_i \phi_i(\|\boldsymbol{x} - \boldsymbol{c_i}\|) \tag{1}$$

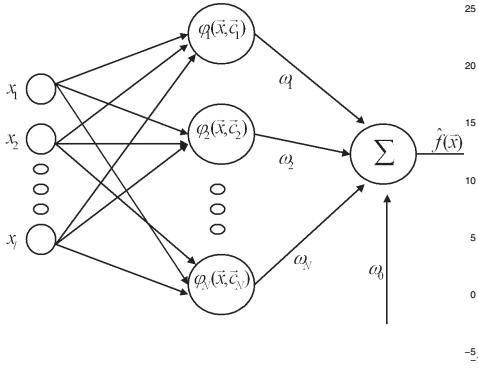


**Fig. 1.** Structure of a RBF network. Left side input layer of $d$ units (one for each input dimension), hidden layer and output unit.

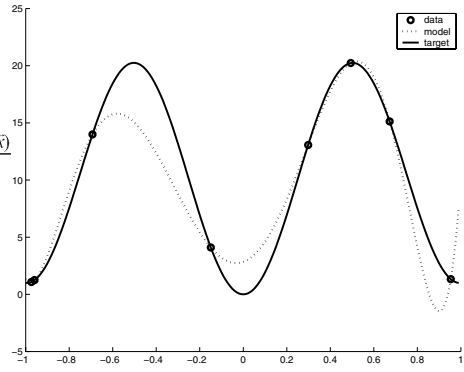


**Fig. 2.** Model output of RBF network for an one dimensional Rastrigin's test function(see A.5).

The term $\phi_i(\|\boldsymbol{x} - \boldsymbol{c_i}\|)$ represents the $i$-th radial basis function which evaluates the distance between the input $\boldsymbol{x}$ and the center $\boldsymbol{c_i}$. For $\phi_i$ we use the gaussian kernel, which is the most common used in practice.

$$\phi_i\left(\|\boldsymbol{x} - \boldsymbol{c_i}\|\right) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c_i}\|^2}{2\sigma^2}\right) \tag{2}$$

The centers $\boldsymbol{c_i}$ are determined by a simple k-means clustering algorithm. The parameter $\sigma$ represents the width of the radial basis functions, which is given here by the mean euclidian distance of all nearest neighbor pairs on the data inputs $\boldsymbol{x_j}$.
Inserting the fitness cases $(\boldsymbol{x}_j, f(\boldsymbol{x}_j))$ from previous fitness evaluations into equation 1 leads to a linear equation system from which the weight parameter vector $\boldsymbol{\omega}$ can be computed by least squares.
We used the RBF network without regularisation. In figure 2 a RBF network output can be seen for a model which is trained with 10 data points from a one dimensional test function (see A.5).

# 3 MODEL ASSISTED STANDARD EVOLUTION STRATEGY

The start of our research is based on a standard $(\mu, \lambda)$ ES which will be later coupled with the model.

## 3.1 STANDARD $(\mu, \lambda)$ EVOLUTION STRATEGY

An ES works on a population of potential solutions $\boldsymbol{x}$ (individuals) by manipulating these individuals with evolutionary operators. $\lambda$ offspring individuals are

generated from $\mu$ parents by applying the evolutionary operators reproduction, recombination and mutation (see pseudocode in figure 3). After evaluating the fitness of the offspring, $\mu$ individuals with the best fitness are selected to built the parent population for the next generation.
The most important evolutionary operator for an ES is the mutation of the objective variables representing the solution of the problem, which is responsible for the self-adaptation capability of the ES. Throughout our study we use Covariance Matrix Adaption (CMA) developed by Hansen et al. [5], which is still the most powerful method, without recombination. The algorithm terminates when a maximum number of fitness function evaluations have been performed.

```
Procedure ES
Begin
 eval=0;
 Pop=CreateInitialPop();
 Pop.EvaluateRealFitness();

 while (eval<maxeval);
  Offspring=Pop.Reproduce(λ);
  Offspring.Mutate();


  Offspring.EvaluateRealFitness();

  Pop=Offspring.SelectBest(μ);
  eval=eval+lambda;
 end while
End
```

**Fig. 3.** Standard $(\mu, \lambda)$ Evolution Strategy (ES).

```
Procedure MA-ES
Begin
 eval=0;
 Pop=CreateInitialPop();
 Pop.EvaluateRealFitness();
 RBFModel.update(Pop);
 while (eval<maxeval);
  PrePop=Pop.Reproduce(λ_Pre);
  PrePop.Mutate();
  PrePop.EvaluateWithRBFModel();
  Offspring=PrePop.SelectBest(λ);
  Offspring.EvaluateRealFitness();
  RBFModel.update(Offspring);
  Pop=Offspring.SelectBest(μ);
  eval=eval+lambda;
 end while
End
```

**Fig. 4.** Model Assisted Evolution Strategy (MA-ES).

### 3.2 MODEL ASSISTED EVOLUTION STRATEGY

To couple the ES with the model we use a preselection concept similar to the one described by Emmerich et al. [4] (see pseudocode in figure 4).

Compared to the standard ES $\lambda_{Pre} > \lambda$ new offspring individuals are reproduced and mutated from $\mu$ parents. These offspring individuals are evaluated by using the prediction $\hat{f}(\boldsymbol{x})$ of the model. Out of these $\lambda_{Pre}$ individuals $\lambda$ individuals with the best predicted fitness are preselected to build the new offspring population. Finally the offspring were evaluated with the real fitness function. The model is updated after each generation step with $\lambda$ new fitness cases.
The idea behind this preselection approach is that only the most promising individuals with a good fitness prediction are evaluated with the real fitness function.

We use a comma selection strategy which selects the $\mu$ best individuals out of the $\lambda$ offspring individuals instead of a plus selection strategy which selects the best $\mu$ out of the $\lambda$ offspring individuals plus its $\mu$ parents. The comma selection strategy has the advantage of a better support of the self-adaption mechanism for the mutation step size control [12], which is necessary to solve complex problems.

The size of the preselected population $\lambda_{Pre}$ controls the impact of the model on the evolutionary optimization process. For $\lambda_{Pre} = \lambda$ the algorithm performs like a standard $(\mu, \lambda)$ ES. Increasing $\lambda_{Pre}$ results in a stronger impact of the model on the convergence behavior of the optimization process.

# 4 MODEL ASSISTED STEADY-STATE EVOLUTION STRATEGY

It is obvious that the performance of a metamodeling approach for an evolutionary optimization process is highly dependent on the quality of the model used. The more often the model is updated with already evaluated fitness cases the better is the quality of the model.
The MA-ES described in the last section has hereby a problem. It is generation based and its model is updated only after $\lambda$ fitness evaluations. This problem could be fixed by dividing the preselection of the $\lambda$ offspring individuals into $\lambda$ steps. Within each step the model should be updated by only one preselected and evaluated individual. But this method leads to a much more complicated algorithm. Therefore we discuss an alternative approach of applying steady-state strategies:
A standard steady-state ES is equivalent to a $(\mu + 1)$ ES [10]. Only one individual is generated and evaluated at each step and gets immediately integrated into the population. Compared to generation based algorithms the information of new evaluated individuals can be integrated directly into the optimization process. But standard steady-state strategies have one big disadvantage. They are missing the ability of self-adaption, which is the most important property of ES. As a result of this the application of standard steady-state ES is not very promising. However, this problem can be solved by the median selection scheme, suggested by Wakunda [13].
The idea is to approximate the selection mechanism of a standard $(\mu, \lambda)$ ES, by using a fitness buffer containing fitness values of the last $n_P$ evaluations. Given a relative rate of acceptance $r_P \equiv \frac{\mu}{\lambda}$. A newly evaluated individual substitutes the worst individual of the population, if it has a better fitness than the $r_P \cdot n_P$ best individuals in the buffer. The pseudocode for the steady-state ES is given in figure 5.
The model is coupled with the steady-state ES in the same way as MA-ES, which results in the new Model Assisted Steady-State ES (MASS-ES) (see pseudocode figure 6). The median selection scheme is again used. The main difference as mentioned before is, that the model will be updated after each fitness evaluation.

```
Procedure SS-ES
Begin
 eval=0;
 Pop=CreateInitialPop();
 Pop.EvaluateRealFitness();

 while (eval<maxeval);
  Individual=Pop.Reproduce(1);
  Individual.Mutate();


  Individual.EvaluateRealFitness();

  Pop = Pop.SelectMedian(Individual);
  eval=eval+1;
 end while
End
```

**Fig. 5.** Steady-State Evolution Strategy (SS-ES) with median selection scheme.

```
Procedure MASS-ES
Begin
 eval=0;
 Pop=CreateInitialPop();
 Pop.EvaluateRealFitness();
 RBFModel.update(Pop);
 while (eval<maxeval);
  PrePop=Pop.Reproduce(λ_Pre);
  PrePop.Mutate();
  PrePop.EvaluateWithRBFModel();
  Individual=PrePop.SelectBest(1);
  Individual.EvaluateRealFitness();
  RBFModel.update(Individual);
  Pop=Pop.SelectMedian(Individual);
  eval=eval+1;
 end while
End
```

**Fig. 6.** Model Assisted Steady-State Evolution Strategy (MASS-ES) with median selection scheme.

## 5 EXPERIMENTAL RESULTS AND DISCUSSION

To analyze and compare the performance of the algorithms extensive simulations were performed for several well-known-real valued 10-dimensional test functions. For each test function the following strategies were compared:

- Standard $(\mu, \lambda)$-ES ($\mu = 2$, $\lambda = 8$),
- RBF network model assisted ES ($\mu = 2$, $\lambda = 8$, $\lambda_{Pre} = 30$),
- Steady-state ES with median selection ($n_P = 10$, $r_P = 0.15$),
- RBF network model assisted steady-state ES with median selection ($n_P = 10$, $r_P = 0.15$, $\lambda_{Pre} = 10$).

All four strategies use covariance matrix adaptation (CMA) for the adaptation of the strategy parameters. For each test function two figures are presented, first the best individual for each generation, and second the mean squared error (MSE) of all $N$ fitness predictions done by the model, both as a function of the number of evaluations of the real fitness function.

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^{N} \left( f(x) - \hat{f}(x) \right)^2 \tag{3}$$

The MSE represents the quality of the RBF network model during the optimization process. The values are always evaluated as the mean from 100 repeated runs with different seed values for random number generation.
The model was built in all cases by the RBF network with 10 hidden neurons,

in section 2 described. The training data for the network consists of the 30 most recently performed fitness evaluations. For this reason the RBF-network is a local model of the individual's neighborhood in object space. Using more training data would improve the performance only slightly but comes along with much higher computational costs for model training.

## 5.1 UNIMODAL TEST FUNCTIONS

The sphere function (A.1) and the weighted sphere function (A.2) are continuously, convex, smooth functions, which are appropriate tests for the self-adaptation mechanism of ES.

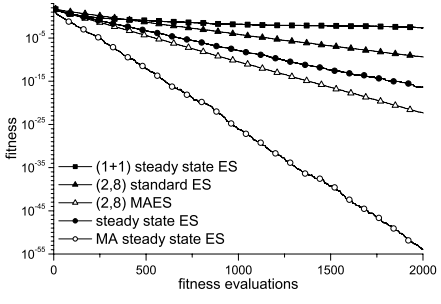


**Fig. 7.** Result for 10-dim. sphere function: fitness of best individual.

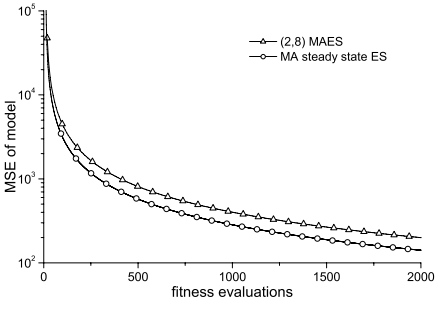


**Fig. 8.** MSE of model, for MA-ES vs MASS-ES.

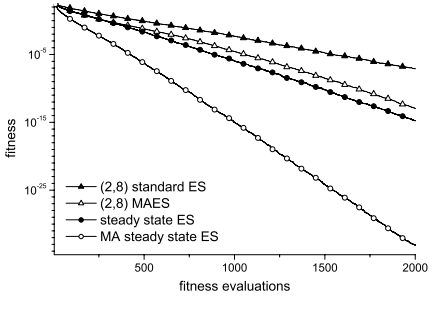


**Fig. 9.** Result for 10-dim. weighted sphere function: fitness of best individual.

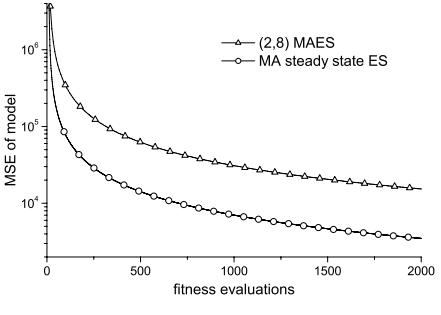


**Fig. 10.** MSE of model, for MA-ES vs MASS-ES.

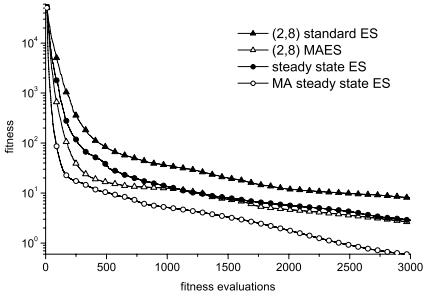


**Fig. 11.** Result for 10-dim. Rosenbrock function: fitness of best individual.

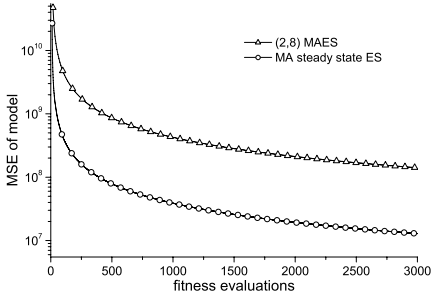


**Fig. 12.** MSE of model, for MA-ES vs MASS-ES.

In Figure 7 the standard (1+1) steady-state ES without median selection has the worst performance. In contrast steady-state ES ($\mu = 1$) with median selection performs much better and also better than the generation based standard ES due to the median selection.
This proves the effectiveness of median selection scheme by providing a self-adaption mechanism comparable to the one of a standard generation based $(\mu, \lambda)$ ES. For this reason standard (1+1) steady-state ES results are not compared later.
The support by the model results in an improvement of convergence both for standard ES and steady-state ES. But the relative enhancement due to the model is in case of the steady-state strategy much higher than in the standard ES case. MASS-ES yields orders of magnitude better solutions than all other strategies (see figure 7 and 9).
This amazing result can be explained by the smaller model error (MSE) of the model for the steady-state ES compared to standard generation based ES (see figure 8, 10), due to the higher model update frequency of the steady-state strategy. The MSE has a hyperbolic appearance, because of the decreasing absolute prediction error of the model with increasing $N$.
The Rosenbrock function (A.3) is nonlinear, continuous and not symmetric. It is a very popular test function and has a very hard to find global optimum. Figure 11 shows again the superiority of the MASS-ES to the other strategies. It reaches the best fitness value and has a remarkably higher convergence speed at the beginning of the optimization process. Also the MSE of the steady-state strategy is smaller than the standard ES one and is responsible for the better performance of MASS-ES.

## 5.2 MULTIMODAL TEST FUNCTIONS

Multimodal functions evoke hills and valleys which are misleading local optima. A simple optimization algorithm like hill-climbing would get stuck in a local minimum. For multimodal functions a population size of $\mu = 10$ for steady state

is selected, in order not to converge too fast into a local optimum.
The Step function (A.4) consists of flat plateaus with slope=0 in an underlying continuous function. It is hard to find the global optimum because minor changes of the object variables do not affect the fitness.

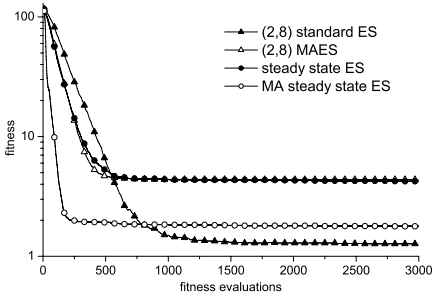


**Fig. 13.** Result for 10-dim. Step function: fitness of best individual.

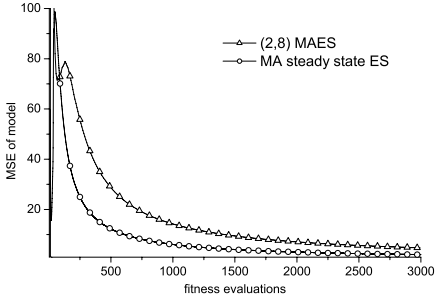


**Fig. 14.** MSE of model, for MA-ES vs MASS-ES.

Here the application of modeling results in an acceleration of the convergence velocity at the beginning (first 500 fitness evaluations) of the optimization process (figure 13). Again the benefit for the steady-state case is bigger and the MSE lower (figure 14). However, both modeling approaches stall in local optima and standard ES achieves a little better fitness values than MASS-ES. Rastrigin's (A.5) and Ackley's (A.6) are very bumpy symmetric test functions. Ackley's function has a global optimum with very strong local features. It is the only here analyzed test function, whose MSE increases temporary and is for MASS-ES higher than for MA-ES (figure 18). This observation could be explained by the very high complexity of Ackley's function. However, MASS-ES reaches the best fitness values (figure 15 and 17), although it has a lower convergence speed at the beginning than the standard ES strategy for Ackley's function. This observation is in contrast to results of Keane [3], whose modeling approach stalls in situations where the optimum has strong local features.

## 6 CONCLUSIONS

We applied metamodeling with RBF networks to standard generation based ES and a steady-state ES with median selection scheme. The strategies were tested in extensive simulations on 6 high-dimensional test functions. Metamodeling enhances the performance in both cases, due to the effective preselection of only very promising individuals by fitness value prediction.
However, the improvement for steady-state is much better than for the standard

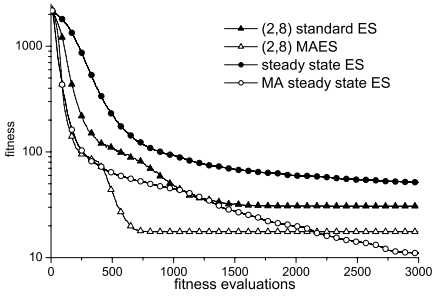


**Fig. 15.** Result for 10-dim. Rastrigin function: fitness of best individual.

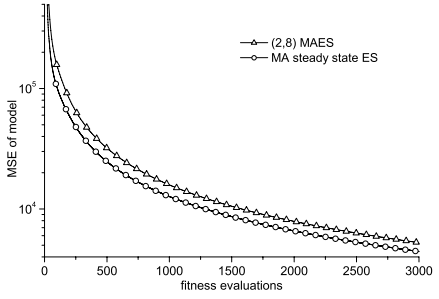


**Fig. 16.** MSE of model, for MA-ES vs MASS-ES.

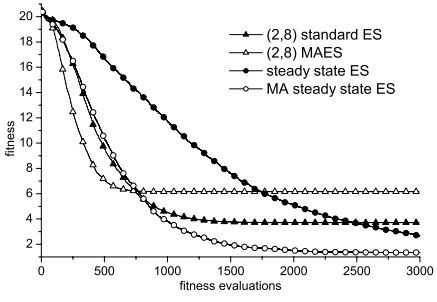


**Fig. 17.** Result for 10-dim Ackley's function: fitness of best individual.

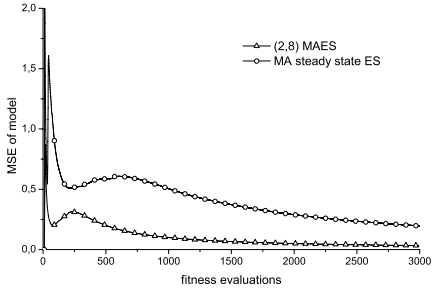


**Fig. 18.** MSE of model, for MA-ES vs MASS-ES.

ES strategies. This can be explained by its superior model quality due to its higher model update frequency. Therefore steady-state strategies are more suitable for metamodeling assistance than generation based strategies.
The new model assisted steady-state ES with median selection achieves equally good or much better results compared to standard ES, MA-ES or standard steady-state ES. It outperforms other strategies especially at unimodal smooth test functions, whereby it yields orders of magnitude better solutions. For multimodal test functions it converges faster at the beginning and achieves equally good or better fitness values compared to the other strategies.

For further work it is planned to develop a mechanism which controls the impact of the model on the ES. The optimization time can be shortened by asynchronous parallel fitness evaluation in a multiprocessor environment. Moreover a comparison of different modeling techniques would be interesting.

# A TEST FUNCTIONS

## A.1 Sphere function

$$f_{Sphere}(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 \tag{4}$$

$-5.12 \leq x_i \leq 5.12$ ; $n = 10$ ; $min(f_{Sphere}) = f_{Sphere}(0,..,0) = 0$

## A.2 Weighted Sphere function

$$f_{WSphere}(\boldsymbol{x}) = \sum_{i=1}^{n} i \cdot x_i^2 \tag{5}$$

$-5.12 \leq x_i \leq 5.12$ ; $n = 10$ ; $min(f_{WSphere}) = f_{WSphere}(0,..,0) = 0$

## A.3 Generalized Rosenbrock's function

$$f_{Rosen}(\boldsymbol{x}) = \sum_{i=1}^{n} (100 \cdot (x_{i+1} - x_i)^2 + (x_i - 1)^2) \tag{6}$$

$-5.12 \leq x_i \leq 5.12$ ; $n = 10$ ; $min(f_{Rosen}) = f_{Rosen}(1,..,1) = 0$

## A.4 Step function

$$f_{Step}(\boldsymbol{x}) = \sum_{i=1}^{n} \lfloor x_i \rfloor \tag{7}$$

$-5.12 \leq x_i \leq 5.12$ ; $n = 10$ ; $min(f_{Step}) = f_{Step}(0,..,0) = 0$

## A.5 Rastrigin's function

$$f_{Rastrigin}(\boldsymbol{x}) = 10 \cdot n + \sum_{i=1}^{n} (x_i^2 - \cos(2\pi x_i)) \tag{8}$$

$-32.768 \leq x_i \leq 32.768$ ; $n = 10$ ; $min(f_{Rastrigin}) = f_{Rastrigin}(0,..,0) = 0$

## A.6 Ackley's function

$$f_{Ackley}(\boldsymbol{x}) = 20 + e - 20 \exp\left(-0.2 \cdot \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) \tag{9}$$

$-32.768 \leq x_i \leq 32.768$ ; $n = 10$ ; $min(f_{Ackley}) = f_{Ackley}(0,..,0) = 0$

## References


1. C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
2. M. EL-Beltagy, P. Nair, and A.Keane. Metamodeling techniques for evolutionary optimization of computationally expensive problems: promises and limitations. In *GECCO 2000 Proceedings of Genetic and Evolutionary Computation Conference*, pages 196–203, 1999.
3. M. A. El-Beltagy and A. J. Keane. Evolutionary optimization for computationally expensive problems using gaussian processes. In CSREA Press Hamid Arabnia, editor, *Proc. Int. Conf. on Artificial Intelligence IC-AI'2001*, pages 708–714, 2001.
4. M. Emmerich, A. Giotis, M.Multlu Özdemir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *Parallel Problem Solving from Nature VII*, pages 362–370, 2002.
5. N. Hansen and A. Ostermeier. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_i, \lambda)$-cma-es. In *5th European Congress on Intelligent Techniques and Soft Computing*, pages 650–654, 1997.
6. Y. Jin, M. Olhofer, and B. Sendhoff. On evolutionary optimisation with approximate fitness functions. In *GECCO 2000 Proceedings of Genetic and Evolutionary Computation Conference*, pages 786–793, 2000.
7. Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation. March 2002 (in press). (ISSN: 1089-778X)*, 2002.
8. Y. Jin and B. Sendhoff. Fitness approximation in evolutionary computing - a survey. In *GECCO 2002 Proceedings of Genetic and Evolutionary Computation Conference*, pages 1105–1111, 2002.
9. A. Ratle. Accelearating the convergence of evolutionary algorithms by fitness landscape approximation. In A. Eiben et al, editor, *Parallel Problem Solving from Nature V*, pages 87–96, 1998.
10. I. Rechenberg. *Evolutionsstrategie '94.* frommann-holzboog, Stuttgart, 1994.
11. H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhäuser, Basel, 1977.
12. H.-P. Schwefel. Natural evolution and collective optimum–seeking. In PA. Sydow, editor, *Computational Systems Analysis: Topics and Trends*, pages 5–14, 1992.
13. J. Wakunda and A. Zell. A new selection scheme for steady-state evolution strategies. In Darell Whitley, David Goldberg, Erick Cantú-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2000*, pages 794–801, Las Vegas, Nevada, July 10-12 2000. Morgan Kaufmann Publishers, San Francisco, California.