

# Comparing Neural Networks and Kriging for Fitness Approximation in Evolutionary Optimization

Lars Willmes

NuTech Solutions GmbH  
Martin-Schmeißer-Weg 15  
44227 Dortmund, GERMANY  
Email: lars.willmes@nutechsolutions.de

Yaochu Jin

Honda Research Institute Europe  
Carl-Legien-Str. 30  
63073 Offenbach/Main, GERMANY  
Email: yaochu.jin@honda-ri.de

**Abstract-** Neural networks and the Kriging method are compared for constructing fitness approximation models in evolutionary optimization algorithms. The two models are applied in an identical framework to the optimization of a number of well known test functions. In addition, two different ways of training the approximators are evaluated: In one setting the models are built off-line using data from previous optimization runs and in the other setting the models are built online from the data available from the current optimization.

## 1 Introduction

This study focuses on the use of meta-modeling techniques in evolutionary optimization algorithms. The driving idea is to substitute computationally expensive objective function evaluations with a computationally cheap, data-driven approximation of the original objective function [8].

Following [9], the optimization algorithm used for this study is based on the Evolution Strategy with covariance matrix adaptation (CMA) [6]. The approximation model is used according to the *population control* model introduced in [9]. That means, that, for a given number of consecutive generations  $c$  (called a "cycle"), the offspring individuals of  $f \leq c$  generations are evaluated by the original objective function, while the offspring individuals of the remaining  $c - f$  generations of a cycle are evaluated by the approximation model. It appears to be realistic to set a fixed budget of 2000 evaluations of the original objective function. Thus, with a given cycle-length  $c$ , the number of generations  $g$  the  $(\mu, \lambda)$  evolution strategy may run depends on the frequency  $f$  through

$$g = \frac{c}{\lambda \cdot f} \cdot 2000.$$

In this study, two different meta-modeling techniques are compared: one is feed-forward neural networks implemented in [9] and the other is the kriging method adopted in [1, 16].

Two different approaches to the adaptation of meta-models for optimization problems are demonstrated. First, the models are built before the optimization starts, i.e. the models are adapted to a data set that was generated before optimization. This is called *off-line learning*. It is assumed that a data set is available from previous optimization runs. In this study, the training data sets were generated by optimizing the test problems with an ordinary (1,10)-Evolution

Strategy with mutative step size adaptation [15]. The resulting data sets had to be truncated in case of fast convergence to a (possibly local) optimum, since otherwise the meta-models are likely to be over-trained in the region close to the (local) optimum.

The second approach to meta-model adaptation involves learning from the data that is generated in the course of optimization, i.e. that in the  $i$ th iteration of the optimization algorithm all data collected in the preceding  $i$  iterations may be used to adapt and refine the approximation model. This approach is called *online learning*.

The remainder of the paper is organized as follows. In Section 2, neural networks and kriging modeling techniques are reviewed very briefly. Three widely used test problems, the Ackley function, the Rosenbrock function and the Keane function are given in Section 3. Section 4 provides the experimental settings adopted in the comparative study followed by the presentation of the simulation results in Section 5. Discussions of the results are carried out in Section 6. Section 7 summarizes the paper and suggests some open questions for further study.

## 2 Neural Network and Kriging Models

### 2.1 Neural Network

Feed-forward neural network models have widely been used for function approximation [12]. In most general cases, layered and fully connected networks are used in combination with the back-propagation (BP) algorithm as the learning method. However, it has been shown that this standard structure and the BP algorithm are often very inefficient. To improve the performance of feedforward neural networks, it is very common to adopt a faster variation of the BP learning algorithm and to optimize the structure of the neural network for a given problem [13].

A Lamarckian framework of evolutionary algorithm is employed to generate the neural network model [7], where both the structure and the parameters are optimized. The available data is split into training data set and a test data set, and the fitness function for training the neural network is as follows:

$$J = 0.75 * E_{TR} + 0.25 * E_{TT}, \quad (1)$$

where  $E_{TR}$  and  $E_{TT}$  are the approximation error on the train-

ing data and test data, respectively. A maximum of 10 hidden neuron is specified and the population size for neural network training is 32. 75 generations are run. Three different runs have been conducted for each data set and the best neural network model is selected for fitness approximation.

## 2.2 Kriging

The kriging method [11, 3, 4, 1] models a system as a localized, stochastic Gaussian process with an expected value  $\mu$  and a covariance matrix  $\Sigma$ . For this study, the so called *Ordinary Kriging* with a Gaussian covariance structure is used, i.e. the system's response  $Z(\vec{x})$  is assumed to be of the form

$$Z(\vec{x}) = E(Z) + \delta(\vec{x}),$$

where  $E(Z)$  is the expected value of  $Z$  and  $\delta$  a stochastic component with a covariance structure

$$c(\vec{x}_i, \vec{x}_j) = \exp(-\theta|\vec{x}_i - \vec{x}_j|^2),$$

where  $\theta$  is to be adapted to the sample data by Maximum Likelihood and  $|\vec{x}_i - \vec{x}_j|$  is the Euclidean distance between  $\vec{x}_i$  and  $\vec{x}_j$ .

The prediction for  $\vec{x}$  is given as

$$\hat{Z}(\vec{x}) = \sum_{i=1}^n \lambda_i Z(\vec{x}_i),$$

where the  $\lambda_i$  are derived from the covariance matrix of the underlying Gaussian process and the sample data. Additionally the kriging method allows to estimate the mean square prediction error  $\sigma_p^2(\vec{x})$ , so that prediction intervals can be constructed as

$$s = (\hat{Z}(\vec{x}) - 1.96\sigma_p(\vec{x}), \hat{Z}(\vec{x}) + 1.96\sigma_p(\vec{x})),$$

where  $s$  is a 95% confidence interval, in case the original process under study is indeed Gaussian.

## 3 Test Problems

The following well known test functions are used to compare the performance of the algorithms (cf. [2, B2.7.3]).

### Ackley-Function

$$f(\vec{x}) = -a \exp\left(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) + a \\ - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + \exp(1),$$

where  $a = 20$ ,  $b = 0.2$ ,  $c = 2\pi$  and  $-32.768 \leq x_i \leq 32.768$ . The minimum is at  $\vec{x}^* = \vec{0}$  with the minimal function value  $f(\vec{x}^*) = 0$

### Rosenbrock-Function

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

where  $-5.12 \leq x_i \leq 5.12$ . The minimum is at  $\vec{x}^* = \vec{1}$  with the minimal function value  $f(\vec{x}^*) = 0$ .

### Keane-Function

$$f(\vec{x}) = - \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right|,$$

subject to

$$0 < x_i < 10 \\ \prod_{i=1}^n x_i > 0.75 \\ \sum_{i=1}^n x_i < 7.5n$$

In all the test functions,  $n$  is the dimension.

## 4 Experimental Setup

Simulations have been conducted for the test functions with a dimension of 10 and 50. For each test function, 20 optimization runs with the help of a meta-model have been performed. In the framework for using meta-models, the control cycle  $c$  is set to 6 and the control frequency  $f$  is fixed to 4 in contrast to an adjustable one in [9]. The reason for a fixed frequency is to remove possible interference between factors from the meta-model used and those from the adaptation of the frequency, because the main purpose of this study is the investigation of different meta-modeling techniques.

### 4.1 Offline

For the 10-dimensional Ackley function a set of 500 learning data points has been used to construct the meta-models. For the 50-dimensional Ackley function a set of 1000 learning data points has been used.

For the 10-dimensional Rosenbrock function a set of 150 learning data points has been generated for building the meta-model. For the 50-dimensional Keane function a set of 700 learning data points has been used. The data are taken from previous optimization runs with standard evolution strategies.

For the 10-dimensional Keane function a set of 150 learning data points has been created. For the 50-dimensional Keane function a set of 700 learning data points has been used. All data were taken from previous optimization runs with standard evolution strategies.

## 4.2 Online

New data becomes available during the optimization. Obviously, these new data should be used for updating the meta-models. For the neural network model, the data generated during the optimization are used to tune the parameters of the neural network online using the RPROP learning method [14]. In other words, the structure of the neural network is determined based on the off-line training data and does not change during optimization.

There are some differences in the online adaptation of the kriging model. During the online learning, a separate kriging model is constructed for each new offspring individual whose fitness is to be predicted. To do so, a subset of all individuals that have already been evaluated with the original objective function  $\mathcal{S}$  is computed. A kriging model is then fitted to the data in  $\mathcal{S}$ . The resulting model is used for prediction of the new individual. In this study, the set  $\mathcal{S}$  consists of the 10 individuals that are closest to the individual to be predicted. In addition, the lower 95% confidence interval is adopted for predictions during the online construction of the kriging models, following a similar suggestion in [5].

## 5 Results

### 5.1 Off-line Learning

The comparison in this subsection will focus on the off-line learning situation with an evolution control frequency of  $f = 4$ . Figures 1-6 each show the convergence of the evolutionary optimization runs supported by meta-model ("Kriging" and "Neural Net") and those without the support of the meta-models ("Plain"). Notice that only the fitness of the individuals using the original fitness function is shown in the figures.

Table 1 summarizes the results. A strategy is assigned a number 1 if it shows the best performance on a test function, a -1 if it shows the worst performance. Otherwise, a 0 is assigned to the strategy. Comparing the sum of the values

|                    | Kriging | Neural Net | Plain |
|--------------------|---------|------------|-------|
| Ackley 10 dim.     | -1      | 0          | 1     |
| Ackley 50 dim.     | -1      | 0          | 1     |
| Rosenbrock 10 dim. | 1       | -1         | 1     |
| Rosenbrock 50 dim. | 0       | -1         | 1     |
| Keane 10 dim.      | 0       | 1          | -1    |
| Keane 50 dim.      | 0       | 0          | 0     |
| Sum                | -1      | -1         | 3     |

Table 1: The number 1 is assigned to case with the best performance, whereas -1 to the worst performance and 0 otherwise. A summary of the results obtained with the model assisted and the plain evolution strategy is provided. For the 50-dimensional Keane function all strategies are awarded a 0, since the results are extremely close.

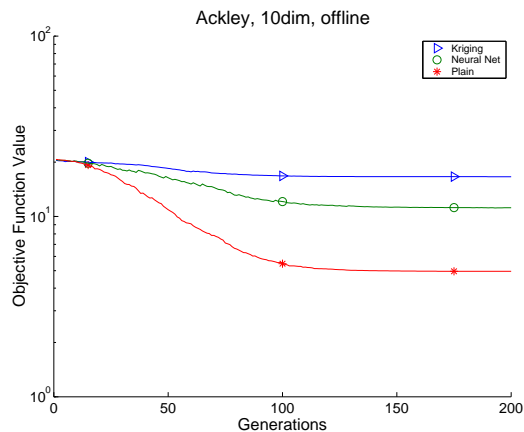


Figure 1: 10-dimensional Ackley function, off-line learning,  $f = 4$ . The kriging supported strategy is the worst, while the plain evolution strategy is better than the neural net supported strategy.

assigned in this way, we find that the plain strategy wins the highest points.

Especially for the Ackley function it is obvious from figures 1 and 2 that the evolutionary algorithm does not benefit from the introduction of the meta-models. In the 50 dimensional case there is hardly any convergence of the model assisted strategies visible, while the plain strategy still seems to improve even in the last iteration.

For the Rosenbrock function the results are slightly less clear, since in the 10 dimensional case (figure 3) at least the kriging supported strategy yields results that are as good as the results of the plain strategy. But in 50 dimensional cases, the plain strategy is clearly better than the supported strategies.

Different results can be seen for the Keane function in figures 5 and 6. In this case the plain strategy does not converge to a clearly better value than the other two strategies, for the 10 dimensional case it even yields the worst results. In 50 dimensions there is hardly a difference visible from the graphics.

### 5.2 Online Learning

The comparison in this subsection will focus on the online learning situation with a frequency of  $f = 4$ . Figures 7-12 each show the convergence of the meta-model supported ("Kriging" and "Neural Net") and the unsupported ("Plain") optimization runs.

The plain strategy performs best for the 50-dimensional Ackley function and the 50-dimensional Rosenbrock function. Only for the 50-dimensional Keane function it performs worst.

The kriging model yields best results for the 10-dimensional Rosenbrock function and the 50-dimensional Keane function. It is the worst for the 10-dimensional Keane function and the 10-dimensional Ackley function.

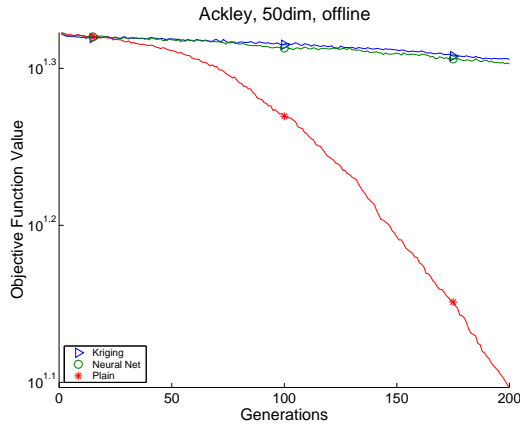


Figure 2: 50-dimensional Ackley function, off-line learning,  $f = 4$ . The plain, unsupported strategy is clearly the best. The meta-model supported strategies perform almost identically.

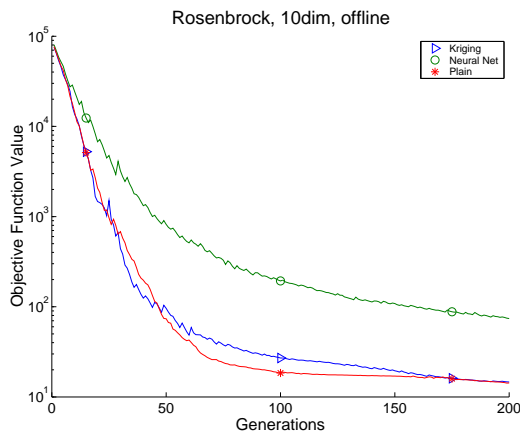


Figure 3: 10-dimensional Rosenbrock function, off-line learning,  $f = 4$ . The Kriging supported strategy and the plain strategy perform comparably well, with the plain strategy being slightly better-in the end. The neural net supported strategy exhibits worse results than the two other strategies.

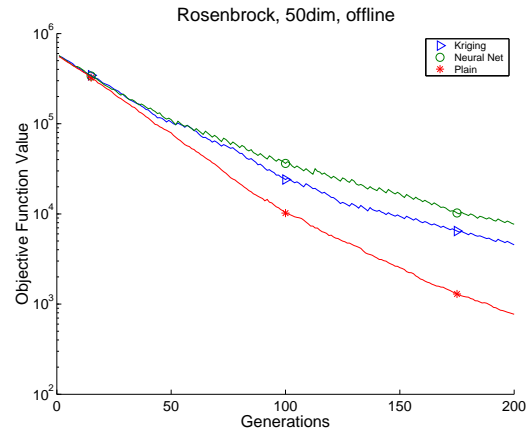


Figure 4: 50-dimensional Rosenbrock function, off-line learning,  $f = 4$ . The plain, unsupported strategy performs best. The kriging supported strategy is slightly better than the neural net supported strategy.

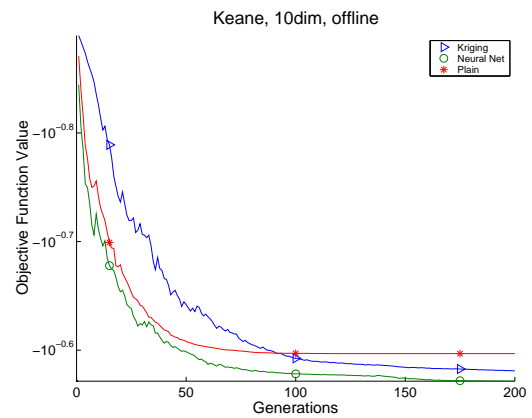


Figure 5: 10-dimensional Keane function, off-line learning,  $f = 4$ . The neural net supported strategy yields the best results, while the plain strategy is worse than the kriging supported strategy. The plain strategy seems to be get stuck quite soon.

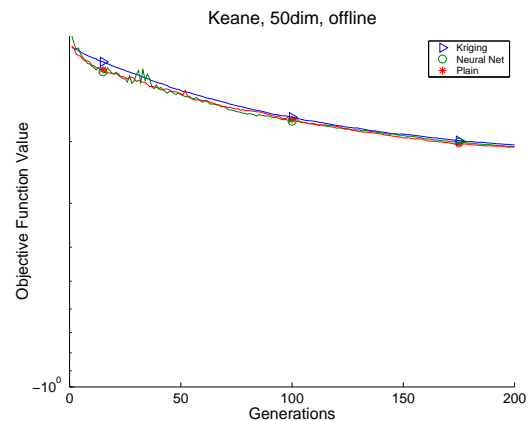


Figure 6: 50-dimensional Keane function, off-line learning,  $f = 4$ . All strategies show quite similar performances.

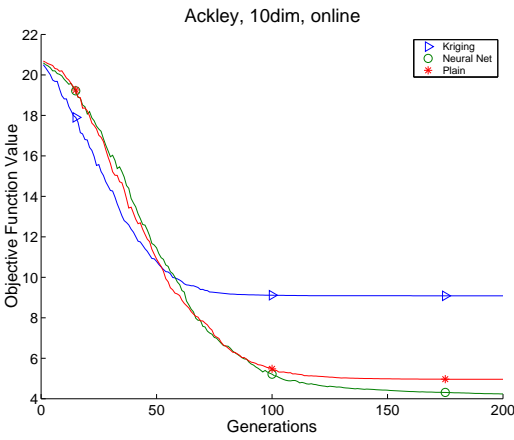


Figure 7: 10-dimensional Ackley function, online learning,  $f = 4$ . The kriging supported strategy is the worst, while the plain evolution strategy is slightly better than the neural net supported strategy.

The neural network wins on the 10-dimensional Ackley function and the 10-dimensional Keane function. It loses on the 50-dimensional Ackley function, the 10-dimensional Rosenbrock function and the 50-dimensional Rosenbrock function.

Table 2 summarizes the results: each strategy is awarded a 1, if it has better performance than the other two strategies on a test function, a  $-1$  if it shows the worst performance and a 0 otherwise. Comparing the sum of the values assigned in this way, we show that the plain strategy is the best strategy as a whole. Although these results are not sufficient to reach a clear conclusion on the performance of the strategies, it does imply that there is no obvious advantage in using the meta-models, when the generation-based evolution control is applied with a fixed frequency to call the approximate model.

## 6 Discussion

The results obtained in this study suggest that neither Kriging nor neural network approximations can achieve robust and

|                    | Kriging | Neural Net | Plain |
|--------------------|---------|------------|-------|
| Ackley 10 dim.     | -1      | 1          | 0     |
| Ackley 50 dim.     | 0       | -1         | 1     |
| Rosenbrock 10 dim. | 1       | -1         | 0     |
| Rosenbrock 50 dim. | 0       | -1         | 1     |
| Keane 10 dim.      | -1      | 1          | 0     |
| Keane 50 dim.      | 1       | 0          | -1    |
| Sum                | 0       | -1         | 1     |

Table 2: Assigning 1 for best performance, -1 for worst performance and 0 otherwise, the table summarizes the results obtained with the model assisted and the plain evolution strategy.

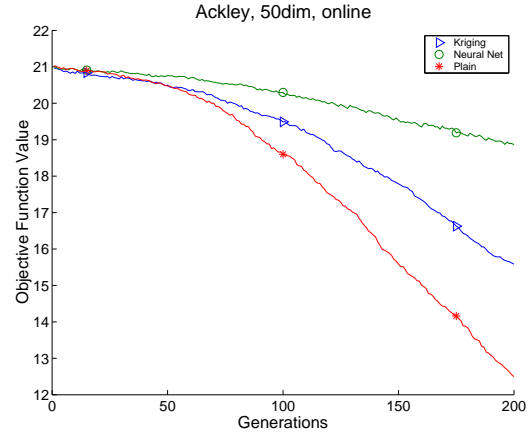


Figure 8: 50-dimensional Ackley function, online learning,  $f = 4$ . The neural net supported strategy obtains the worst, while the plain unsupported strategy achieves best.

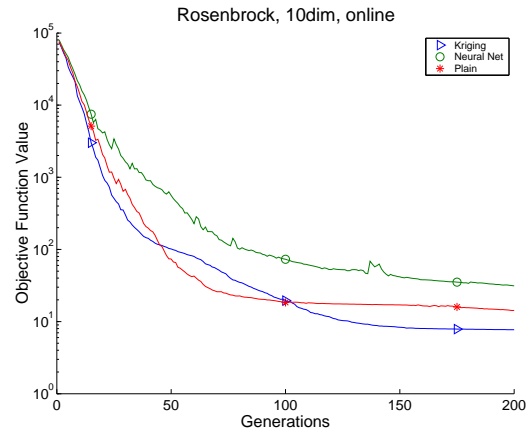


Figure 9: 10-dimensional Rosenbrock function, online learning,  $f = 4$ . The Kriging supported strategy starts very well, is beaten by the plain strategy between the 50th and 100th generation, but interestingly finishes best in the end. The neural net supported strategy turns out with the worst performance.

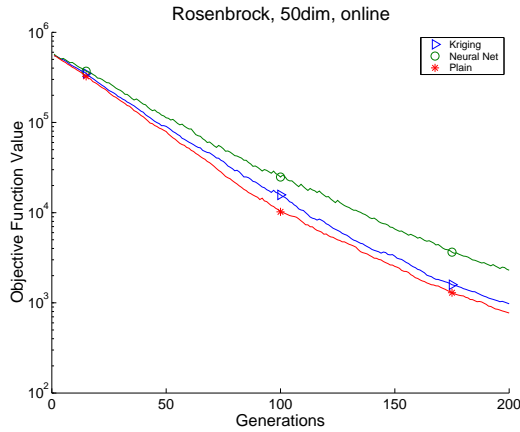


Figure 10: 50-dimensional Rosenbrock function, online learning,  $f = 4$ . The plain, unsupported strategy is slightly better than the kriging supported strategy, which itself is slightly better than the neural net supported strategy.

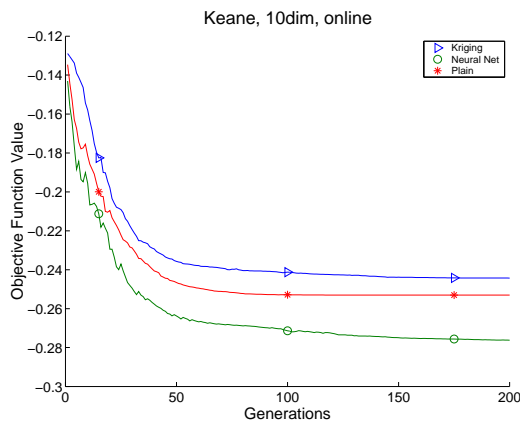


Figure 11: 10-dimensional Keane function, online learning,  $f = 4$ . The neural net supported strategy yields the best results, while the plain strategy is better than the kriging supported strategy.

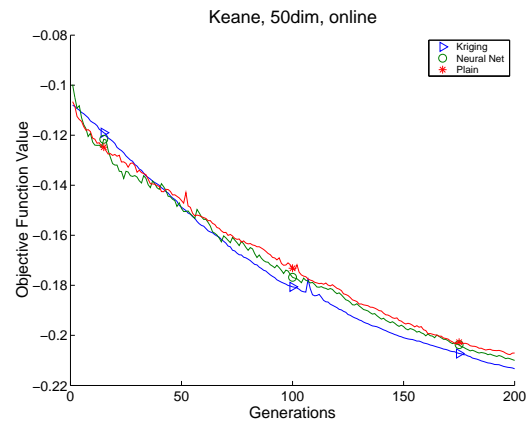


Figure 12: 50-dimensional Keane function, online learning,  $f = 4$ . The kriging supported strategy achieves the best results, while the neural net supported strategy is slightly better than the plain strategy. The results are very close though.

convincing acceleration of the optimization, when the generation based evolution control strategy is applied with a fixed control frequency.

It is not surprising to observe that the prediction quality differs depending on the learning schedule. Our results indicate that meta-models trained off-line could provide unexpected predictions on unseen data points while models trained online give better predictions, which suggests that the construction of local meta-models is more practical and reliable than global models. Nevertheless, evolutionary optimization supported by local models with relatively good prediction quality could still disturb the evolutionary search. From the results obtained in this study, no conclusion can be drawn on which meta-model is better for fitness approximation in evolutionary optimization.

Compared to the more promising results obtained in [10], it is believed that the online adaptation of the control frequency may be critical to the success of using meta-models for fitness evaluation in evolutionary computation. A straightforward explanation to this is the fact that if a meta-model is of poor quality, then this model should not be used often at the beginning of the search. Obviously, the model quality improves as the evolutionary search proceeds. Thus, a general rule is that the original function should be used more often at the beginning so that the quality of the meta-model can be improved significantly. After a certain number of generations, the model becomes much more reliable and can be employed more often for fitness evaluation.

In addition, the differences between the results found in this study and the results reported in [5] indicate that the way how the meta-model is used plays a crucial role for the optimization performance. It is speculated that when the quality of the meta-model is not of sufficient quality, individual-based evolution control could be more stable than the generation-based evolution control.

Figures 13 to 15 show some more details of the con-

vergence of the fitness values when only the approximation model is used. In these figures, both the real value of the 10-dimensional Rosenbrock function and that predicted by the kriging model in three consecutive generations are plotted. The real fitness values of the 10 offspring individuals of each generation are connected by a solid line. For each individual the real fitness value and the predicted fitness value are connected by a dashed line. The best individual of each generation based on the original fitness function is marked with a circle and the best one based on the predicted values are marked with a diamond.

Figure 13 shows that for the offspring individuals 990-999 and 1000-1009 only one parent is identified correctly. That means that due to the Kriging prediction some individuals are chosen as parents that would not have been chosen if the original fitness value would have been available. For individuals 1010-1019 no parent is identified correctly. It should be noted that for individuals 990-999 even the individual with the worst original function value is selected as a parent.

From Figure 14 it can be seen how the prediction quality degrades when the search proceeds on the approximation only. For the offspring individuals 3510-3519 the approximation is quite good and the parent individuals are selected correctly. For individuals 3520-3529 the approximate values are a bit more off the solid line which indicates slightly worse approximation quality. One parent is misclassified. For individuals 3530-3539 the approximation is quite bad and only one parent is identified correctly.

Figure 15 shows something very annoying: The values of the predicted fitness of the best individual is getting better in the three generations plotted while the best value of the original function is getting worse. These results indicate that evolution on a meta-model only could lead to very unreliable results.

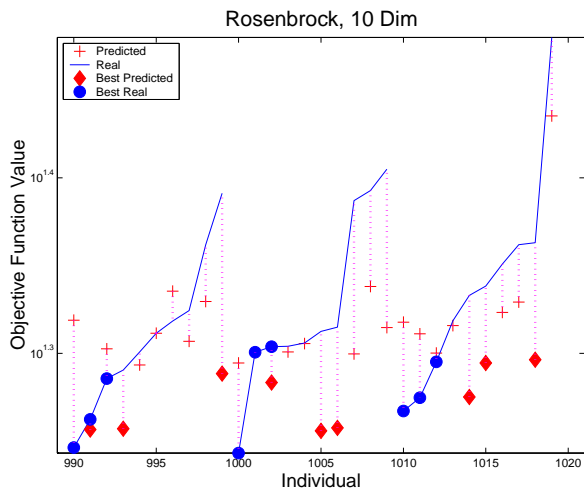


Figure 13: 10-dimensional Rosenbrock function,  $f = 3$ . For individuals 990-999 Kriging suggests the worst individual as one of the parents. For individuals 1010-1019 Kriging does not suggest one single parent correctly.

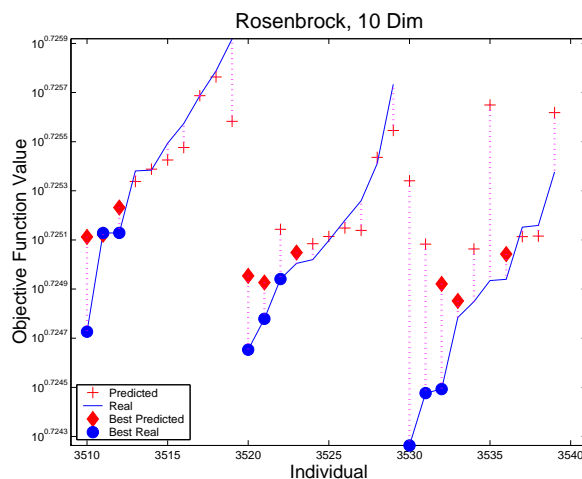


Figure 14: 10-dimensional Rosenbrock function,  $f = 3$ . For individuals 3510-3519, parents are identified correctly. For individuals 3520-3529, two out of three parents are identified correctly. For individuals 3530-3539 only one parent is identified correctly.

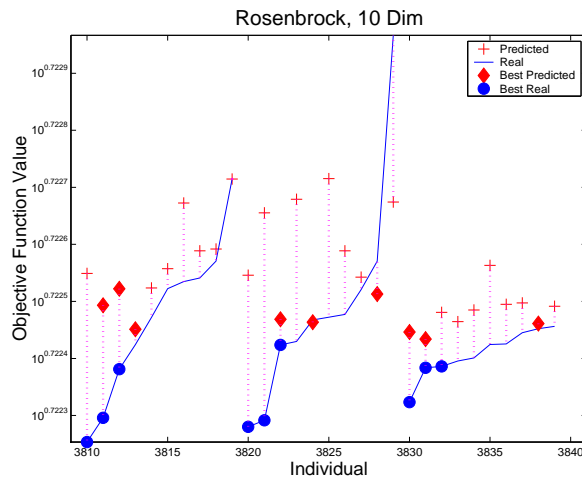


Figure 15: 10-dimensional Rosenbrock function,  $f = 3$ . During three generations the predicted performance improves, while the real performance degrades.

## 7 Summary and Conclusion

In this study three different test functions, namely the Ackley function, the Rosenbrock function, and the Keane function have been used to compare the optimization performance using evolution strategies assisted by meta-models. As meta-models the kriging and the neural network based techniques are used.

For both techniques, two different modes of model construction have been considered. In the off-line learning mode, data from previous optimization runs are used to build a

model before it is applied in optimization. In the online learning mode, the meta-model is updated repeatedly with new data generated from the optimization. In all cases the online learning mode showed significantly better results than the offline learning mode.

During optimization the meta-model is used according to the generation-based evolution control. Neither the kriging model nor the neural network could clearly demonstrate an advantageous performance over an evolutionary optimization without meta-models. Very often, optimization assisted with a meta-model could lead to a degraded performance.

This study leaves a number of open questions:

- What are the main reasons for performance degradation of optimization, when the number of "model only" iterations is increased?
- How far can the adaptation of strategy parameters of evolution strategies be influenced by using the meta-model?
- Why are the results found in [5] so much better than the results found in this study?

Answering these questions will be subject to future research.

## Bibliography

- [1] M. Armstrong. *Basic Linear Geostatistics*. Springer-Verlag, Berlin, 1998.
- [2] Th. Bäck, D. B. Fogel, and Zbigniew M., editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, Bristol/New York, 1997.
- [3] I. Clark. *Practical Geostatistics*. Applied Science Publishers, Essex, 1979.
- [4] N. A.C. Cressie. *Statistics for Spatial Data – Revised Edition*. Wiley series in probability and mathematical statistics. John Wiley & Sons, Inc., New York, 1993.
- [5] M. Emmerich, A. Giotis, M. Özdemir, Th. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In V. Merelo Guervós, P. Adamidis, H.-G. Beyer, J.-L. Fernández-Villacañas, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VII, 7th International Conference, Granada, Spain, September 7-11, 2002. Proceedings*, number 2439 in Lecture Notes in Computer Science, LNCS, page 361 ff. Springer-Verlag, 2002.
- [6] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized selfadaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [7] M. Hüskens and B. Sendhoff. Evolutionary optimization for problem classes with Lamarckian inheritance. In Soo-Young Lee, editor, *Seventh International Conference on Neural Information Processing (ICONIP 2000) – Proceedings*, volume 2, pages 897–902, 2000.
- [8] Y. Jin. Fitness approximation in evolutionary computation: A survey. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1105–1112, New York, USA, 2002. Morgan Kaufmann.
- [9] Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, 2002.
- [10] Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, 2002.
- [11] A.G. Journel and Ch.J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978.
- [12] S. Lawrence, A. C. Tsoi, and A. D. Back. Function approximation with neural networks and local methods: Bias, variance and smoothness. In Peter Bartlett, Anthony Burkitt, and Robert Williamson, editors, *Australian Conference on Neural Networks*, pages 16–21. Australian National University, 1996.
- [13] R. D. Reed and R. J. Marks II. *Neural Smoothing: Supervised Learning in Feedforward Artificial Neural Networks*. The MIT Press, Cambridge, MA, 1999.
- [14] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591. IEEE Press, 1993.
- [15] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
- [16] H. Wackernagel. *Multivariate Geostatistics – An Introduction with Applications*. Springer Verlag, Berlin Heidelberg, second, completely rev. edition, 1998.