

Evolution Strategies assisted by Gaussian Processes with improved Pre-Selection Criterion

Holger Ulmer, Felix Streichert, Andreas Zell

Center for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72076 Tübingen, Germany,
ulmerh@informatik.uni-tuebingen.de,
<http://www-ra.informatik.uni-tuebingen.de>

Abstract- In many engineering optimization problems, the number of fitness function evaluations is limited by time and cost. These problems pose a special challenge to the field of evolutionary computation, since existing evolutionary methods require a very large number of problem function evaluations. One popular way to address this challenge is the application of approximation models as a surrogate of the real fitness function. We propose a model assisted Evolution Strategy, which uses a Gaussian Process approximation model to pre-select the most promising solutions. To refine the pre-selection process we determine the likelihood of each individual to improve the overall best found solution. Due to this, the new algorithm has a much better convergence behavior and achieves better results than standard evolutionary optimization approaches with less fitness evaluations. Numerical results from extensive simulations on several high dimensional test functions including multimodal functions are presented.

1 Introduction

Evolution Strategies (ES) are excellent optimization tools for complex high dimensional multimodal real valued problems [10] [11]. However, like other population based algorithms they require a very high number of fitness function evaluations. In most real world applications the process of fitness evaluation is very expensive and time consuming. Therefore standard ES are not practical for such applications.

A promising approach to handle this problem is the application of modeling techniques, where a model evaluation is orders of magnitude cheaper than a real fitness function evaluation. A model is trained on already evaluated individuals and is used to guide the search for promising solutions. This approach decreases the number of expensive fitness evaluations, which results in a better convergence rate of the algorithm.

The application of modeling techniques in evolutionary computation receives increasing attention [7] [2] [3]. A survey on this research field can be found in [8] and [6]. Two major points have to be considered:

- **Model Selection:** The selection of an appropriate model to approximate the fitness function is a central point. Neural networks [7] [9] are widely used for function approximation and are therefore used for modeling in evolutionary optimization. Gaussian Processing [2] and Kriging [3] are statistical modeling techniques which are also used for fitness function approximation.
- **Model Management:** The coupling of the model used with the evolutionary algorithm 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 [7] controls the impact of the model on the evolutionary optimization process. The Metamodel-Assisted Evolution Strategy (MA-ES) [3] uses the estimation of the model to pre-select the most promising individuals before applying the expensive real fitness function. Another approach is to use the confidence criterion given by statistical models like Kriging [3] or Gaussian Processing [2] to control the interaction of the model with the evolutionary optimization process.

Concerning model management, in our opinion the key issue of using approximation models for evolutionary computation lies in the tradeoff between the exploitation of the approximation model by sampling where it is optimized and the need to improve the approximation model by sampling where the model confidence is low. In this work we try to address this tradeoff by introducing a model management approach, which takes both necessities into account.

The remainder of this paper is organized as follows: We first describe the modeling technique of Gaussian Process (GP) approximation, which is used to approximate the expensive real fitness function and provides also information about the model confidence in section 2. Section 3 introduces the synthesis of the approximation model with a standard ES. The GP model is utilized to assist the ES by selecting the most promising solutions of an offspring population to be evaluated by the real fitness function.

We analyse the impact of the GP model on the evolutionary optimization process and introduce two different search cri-

teria to identify the most promising solutions.

Numerical results from extensive simulations on several high dimensional test functions are presented and discussed in section 4. The paper closes with a brief conclusion and outlook on future work.

2 Fitness Approximation with Gaussian Processes

For the approximation of the fitness function we chose Gaussian Processes (GP), which are known to be general and proper real valued function approximators, also for noisy training data. A detailed description is given in [4]. Compared to other models like artificial neural networks or RBF-networks GP are probabilistic models, which have the advantage of providing a confidence value given by the standard deviation σ for the predicted fitness value t without additional computational cost. Moreover GP is stable against overfitting and has only a limited number of model parameters, which have to be chosen by the user.

Consider a d -dimensional real valued problem with scalar fitness function $f(\vec{x})$, which is to be minimized. As stated in the introduction, an approximation model is needed to predict the fitness of an individual \vec{x} .

Given a training data set \mathcal{D} consisting of N fitness cases $(\vec{x}_n, f(\vec{x}_n))$ for $n = 1, \dots, N$, we want to predict the target t_{N+1} at new data points $\vec{x}_{N+1} \notin \mathcal{D}$.

This is done in the GP framework by evaluating the conditional distribution of the prediction t_{N+1} . A Gaussian Process is a collection of samples $\vec{t}_N = (t_1, t_2, \dots, t_N)$, which has a joint distribution,

$$P(\vec{t}_N | \mathcal{D}) = \frac{1}{Z_N} \exp\left(-\frac{1}{2}(\vec{t}_N - \mu)^T C^{-1}(\vec{t}_N - \mu)\right) \quad (1)$$

with the covariance function,

$$\begin{aligned} C_{mn} &= C(\vec{x}_m, \vec{x}_n; \Theta) \\ &= \Theta_1 \exp\left(-\frac{1}{2} \sum_{l=1}^d \frac{(x_m^l - x_n^l)^2}{r_l^2}\right) + \delta_{mn} \Theta_2 \end{aligned} \quad (2)$$

which evaluates the components of the covariance matrix C .

The covariance function describes the correlation between the N already evaluated fitness cases. The parameter Θ_1 scales the correlation between two data points and Θ_2 sets the noise level. The radii r_l for $l = 1, \dots, d$ provide a distinct distance measure for each input dimension. For irrelevant inputs of dimension l , the corresponding r_l will be large and the GP will ignore this input dimension.

Let us assume vector \vec{t}_N of targets to be a Gaussian Process with covariance matrix C_N and mean $\mu = 0$ (for properly normalized data), then the conditional Gaussian distribution

over $\vec{t}_{N+1} = (\vec{t}_N, t_{N+1})$ can be expressed as:

$$P(\vec{t}_{N+1} | \mathcal{D}, \vec{x}_{N+1}) = \frac{1}{Z_N} \exp\left(-\frac{1}{2} \vec{t}_{N+1}^T C_{N+1}^{-1} \vec{t}_{N+1}\right) \quad (3)$$

$$\begin{aligned} P(t_{N+1} | \mathcal{D}, \vec{x}_{N+1}) &= \frac{P(\vec{t}_{N+1} | \mathcal{D}, \vec{x}_{N+1})}{P(\vec{t}_N | \mathcal{D}, \vec{x}_N)} \\ &= \frac{1}{Z} \exp\left(-\frac{(t_{N+1} - \hat{t}_{N+1})^2}{2\sigma_{N+1}^2}\right) \end{aligned} \quad (4)$$

$$\hat{t}_{N+1} = \vec{k}_{N+1}^T C_N^{-1} \vec{t}_N \quad (5)$$

$$\sigma_{N+1}^2 = \kappa - \vec{k}_{N+1}^T C_N^{-1} \vec{k}_{N+1} \quad (6)$$

With autocorrelation $\kappa = C(\vec{x}_{N+1}, \vec{x}_{N+1}; \Theta)$ of the new data point \vec{x}_{N+1} and correlation vector $\vec{k}_{N+1} = (C(\vec{x}_1, \vec{x}_{N+1}; \Theta), \dots, C(\vec{x}_N, \vec{x}_{N+1}; \Theta))$. \hat{t}_{N+1} is the mean of the prediction at \vec{x}_{N+1} and σ_{N+1} is the standard deviation of t_{N+1} and describes the model confidence at this point \vec{x}_{N+1} .

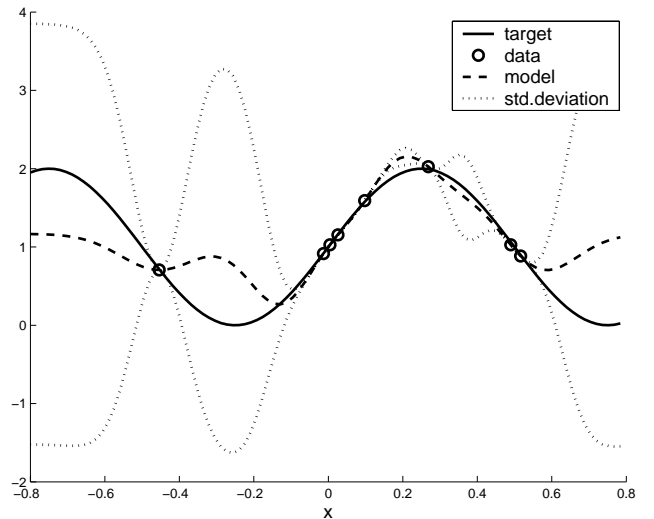


Figure 1: GP model trained on 8 data points given by a 1-dim. sinus target function (real fitness function). Unexplored areas of object space with less training data inputs have a bigger standard deviation than explored ones.

The $d + 2$ hyperparameter set $r_1, \dots, r_d, \Theta_1, \Theta_2$ are evaluated by maximisation of the posterior probability of the hyperparameter using Bayes's theorem. In general this is carried out using a gradient-based optimizer [4].

Figure 1 shows a 1-dimensional example of a trained GP model.

3 GP Model Assisted Evolution Strategy

We start our consideration with a standard (μ, λ) ES, which will be later coupled with the GP model.

3.1 Standard (μ, λ) Evolution Strategy ES

An ES works on a population of potential solutions \vec{x} (individuals) by manipulating these individuals with evolutionary operators [10] [11].

λ offspring individuals are generated from μ parents by

```

Procedure ES
Begin
  eval=0;
  Pop=CreateInitialPop();
  Pop.EvalRealFitness();
  while (eval<maxeval);
    Offspring=Pop.Reproduce( $\lambda$ );
    Offspring.Mutate();
    Offspring.EvalRealFitness();
    Pop=Offspring.SelectBest( $\mu$ );
    eval=eval+ $\lambda$ ;
  end while
End

```

Figure 2: Standard (μ, λ) Evolution Strategy (ES).

applying the evolutionary operators reproduction, recombination and mutation (see pseudocode in Figure 2).

After evaluating the fitness of the λ offspring individuals, μ individuals with the best fitness are selected by a (μ, λ) strategy to build the parent population for the next generation.

On the other hand, a $(\mu + \lambda)$ strategy selects the best μ individuals from the aggregation of parents and offspring individuals.

The algorithm terminates when a maximum number of fitness function evaluations have been performed.

3.2 Model Assisted Evolution Strategy MAES

To incorporate the approximation model into the ES we use a pre-selection concept similar to the one described by Emerich et al. [3]. Compared to the standard ES $\lambda_{Pre} > \lambda$ new offspring individuals are created from μ parents (see pseudocode in Figure 3). These λ_{Pre} individuals have to be pre-selected to generate the offspring of λ individuals, which will be evaluated with the real fitness function. The model is trained at the beginning with a randomly created initial population and is updated after each generation step with λ new fitness cases.

```

Procedure MAES
Begin
  eval=0;
  Pop=CreateInitialPop();
  Pop.EvalRealFitness();
  GPModel.update(Pop);
  while (eval<maxeval);
    Offspring=Pop.Reproduce( $\lambda_{Pre}$ );
    Offspring.Mutate();
    Offspring.SetPreSelectionCriterion();
    Offspring=Offspring.SelectBest( $\lambda$ );
    Offspring.EvalRealFitness();
    GPModel.update(Offspring);
    Pop=Offspring.SelectBest( $\mu$ );
    eval=eval+ $\lambda$ ;
  end while
End

```

Figure 3: GP Model Assisted Evolution Strategy (GP MAES). The MAES can be easily developed by expanding the standard ES with the bold marked lines.

The pre-selection concept is comparable with the task of Active Learning, which has been widely studied in the field of neural network computing. The basic idea is to select an individual or sample in such a way that an objective function is optimized. Several objective functions as selection criterion are known. But all approaches are based on the same ideas, that this criterion should represent the quality of the individual and only the most promising individuals should be selected to be later evaluated with the fitness function. We investigate here two different selection criteria:

3.2.1 Mean of Model Prediction (MMP)

The motivation of this criterion is straightforward, very simple and often used in other work. The λ_{Pre} possible candidates are pre-evaluated using the mean of the prediction given by the GP model. The individuals $\vec{x}_i, i = 1, \dots, \lambda_{Pre}$ with the best mean predicted fitness $\hat{t}(\vec{x})$ are pre-selected to generate the new offspring. The approximation model is used directly as a surrogate of the real fitness function, which should lead to a decrease in the number of fitness evaluations [3].

3.2.2 Probability of Improvement (POI)

Using only the mean of model prediction \hat{t} to identify the most promising individuals comes along with one major drawback.

On multimodal problems with many misleading local minima MMP leads to premature and suboptimal convergence, because individuals with a better model prediction are

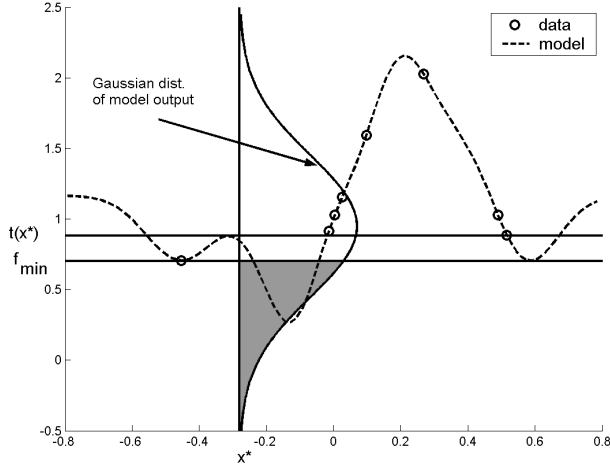


Figure 4: Illustration of the POI concept based on the same data as in Figure 1. The gray filled area represents the probability that a model output value t is sampled at point x^* , which is smaller than f_{min} .

preferred to others and therefore have a lower probability to escape from these minima.

One approach to address this problem is to use the standard deviation σ of the model output to prefer unexplored areas by defining the objective function $f_o = \hat{t}(\vec{x}) - \alpha\sigma(\vec{x})$, which has to be minimized [12]. α scales the impact of the standard deviation. The major drawback of this idea is to find an appropriate α , because the performance of this approach is highly dependent on α especially for high dimensional problems.

For this reason we introduce a new pre-selection criterion, which utilizes also the model confidence given by the GP model and has no parameter selection problem. The idea is not new in the field of global optimization [1], but new in the context of evolutionary optimization.

The concept is illustrated in Figure 4. At any given point \vec{x} , we model the uncertainty about the model value prediction by considering this value to be like the realization of a random variable $Y(\vec{x})$ with mean $\hat{t}(\vec{x})$ and standard deviation $\sigma(\vec{x})$.

Let $f_{min} = \min(t_1, \dots, t_N)$ be the current best until now sampled fitness value, then the target value for the improvement will be some number $T \leq f_{min}$. The probability of improvement (POI) is simply the probability that $Y(\vec{x}) \leq T$. Assuming the random variable is normal distributed, this probability is given by

$$POI(\vec{x}) = \Phi\left(\frac{T - \hat{t}(\vec{x})}{\sigma(\vec{x})}\right) \quad (7)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function.

Figure 5 shows the characteristics of the POI selection criterion. Areas with a high POI have a high probability to sample a data point with a target value smaller than f_{min} and are therefore more promising. Areas with model prediction $\hat{t}(\vec{x}) \gg f_{min}$ have a low probability of improvement $POI \approx 0$. As the function is sampled more and more around the current best point, the standard deviation in this area decreases. The term $\frac{T - \hat{t}(\vec{x})}{\sigma(\vec{x})}$ becomes extremely negative and POI will be so small that the algorithm is driven to search elsewhere in unexplored areas where the standard deviation is higher. Therefore POI prefers unexplored areas of object space and has a multimodal characteristic. Note, that the maximal POI value has another location in object space than the minimal model output value. The individuals \vec{x}_i , $i = 1, \dots, \lambda_{Pre}$ with the highest $POI(\vec{x})$ are pre-selected to build the new offspring.

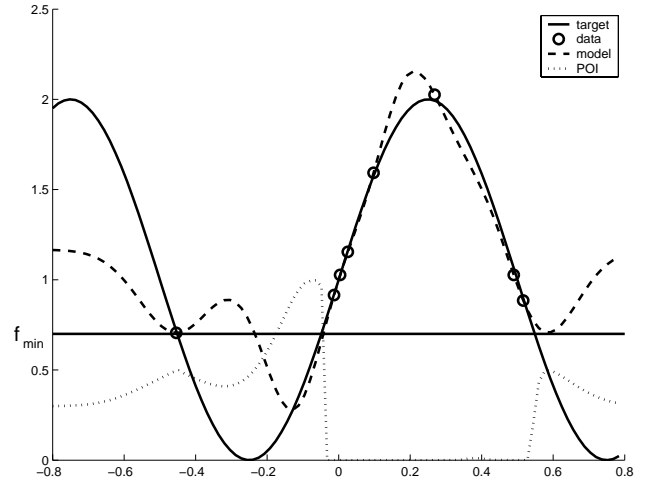


Figure 5: Characteristics of the POI criterion based on the same data as in the former figures. Areas with a higher POI have a higher probability to sample a data point with a target value smaller than f_{min} .

The size of the pre-selected population λ_{Pre} controls the impact of the model on the evolutionary optimization process. For $\lambda_{Pre} = \lambda$ the algorithm performs like a standard (μ, λ) ES. Increasing λ_{Pre} results in a bigger selection pressure in the pre-selection and in a stronger impact of the model on the convergence behavior of the optimization process.

4 Experimental results and discussion

To analyze the performance of the algorithms extensive simulations were performed for several well known real valued unimodal and multimodal test functions. For each test function the following algorithms were compared:

- Standard (μ, λ) -ES
- (μ, λ) -MAES with MMP pre-selection
- (μ, λ) -MAES with POI pre-selection

For the model assisted algorithms the size of the pre-selected population λ_{Pre} was set to 3λ . Throughout our study we used Covariance Matrix Adaption (CMA) developed by Hansen et al. [5], which is still the most powerful method for adaption of the mutation step size. For all simulations no recombination was used and the initial population size was set to 10.

The model was built in all cases by the GP as in section 2 described. The training data for the GP model consisted of the 2λ most recently performed fitness evaluations. For this reason the model is a local model of the individual's neighborhood in object space. Using more training data improves the performance only slightly but results in with much higher computational costs for model training.

The values are always evaluated as the mean of 100 repeated runs with different seed values for random number generation.

4.1 Unimodal Test Functions

The Sphere function (A.1) is a nonlinear, continuous, convex, smooth function, which is an easy test for the self-adaptation mechanism of ES.

The standard $(1, 10)$ ES has the worst performance (see Figure 6). Both model assisted approaches improve the convergence speed of the standard ES. MMP has the highest convergence speed and performs better than POI. This can be explained due to the higher tendency of POI to sample in unexplored areas (see Figure 5), which bothers the convergence for unimodal problems with constant search direction. However POI outperforms the standard ES clearly.

The same observations are obtained with several other unimodal functions. We present the results in Figure 7 for one more unimodal function, which is given by Schwefel's function 1.2 (A.2)

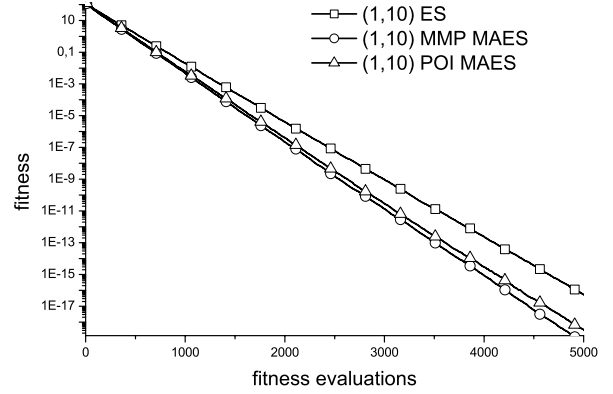


Figure 6: 20-dim. Sphere function: fitness of best individual ($\mu = 1, \lambda = 10$).

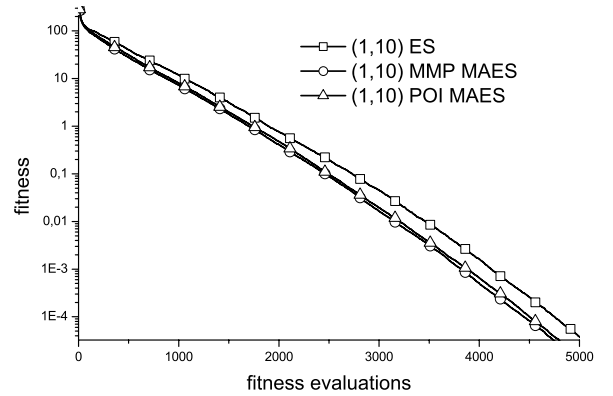


Figure 7: 20-dim. Schwefel's function 1.2: fitness of best individual ($\mu = 1, \lambda = 10$).

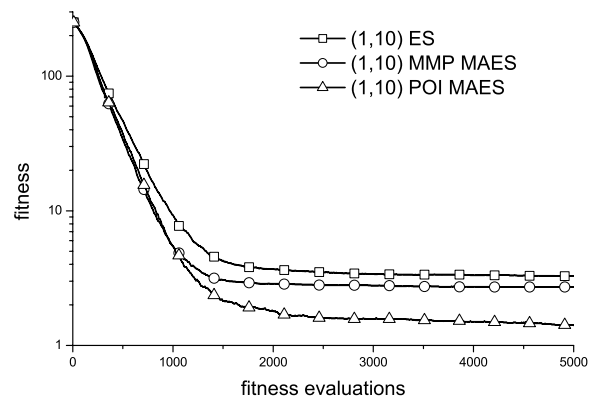


Figure 8: 20-dim. Step function: fitness of best individual ($\mu = 1, \lambda = 10$).

4.2 Multimodal Test Functions

Multimodal functions evoke hills and valleys, which are misleading local optima. A simple optimization algorithm

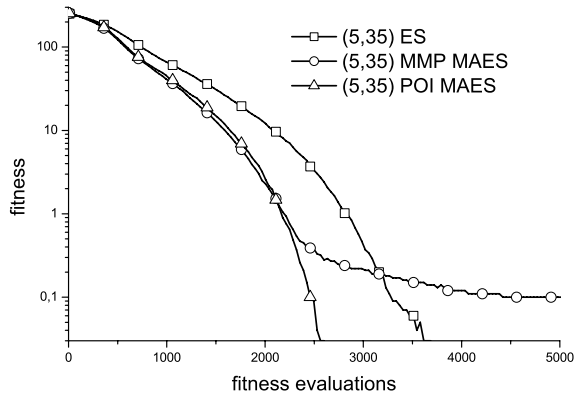


Figure 9: 20-dim. Step function: fitness of best individual ($\mu = 5, \lambda = 35$).

like hill-climbing would get stuck in a local minimum. For such problems evolutionary algorithms are much more appropriate. In general an ES needs a bigger population size to manage multimodal problems compared with unimodal problems. Therefore we present results for (1,10) and (5,35) strategies.

The Step function (A.3) 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. Therefore no conclusions about the search direction are possible.

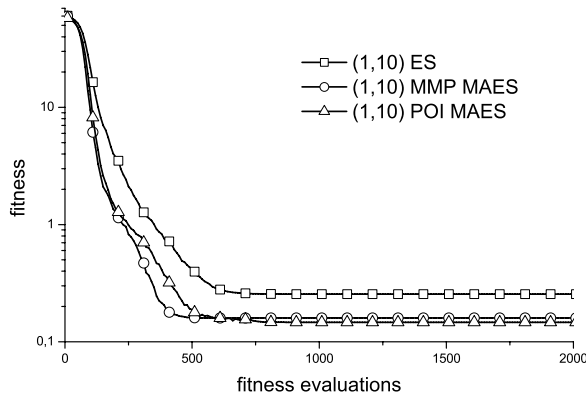


Figure 10: 5-dim Griewank function: fitness of best individual ($\mu = 1, \lambda = 10$).

Here the application of modeling results in an acceleration of the convergence velocity at the beginning of the optimization process (Figure 8 and 9). For the (1,10) algorithms no strategy reliably reaches the global optimum (note, that the figures show the mean from 100 runs of the best individual). But POI achieves clearly the best results.

Due to the bigger population size the (5,35) algorithms are more stable against premature convergence (Figure 9). It is remarkable that POI and standard ES are always reaching the global optimum, but not the MMP algorithm. Here the usage of MMP as pre-selection criterion increases the probability of premature convergence. This observation justifies the motivation of POI to sample also in unexplored areas of the objective space.

Figure 10 and 11 shows the results for the Griewank function (A.4). Note, that compared to the other functions the Griewank function has a stronger multimodal characteristic for low dimensions than for high dimensions. POI also reaches the best fitness values, but is not stable against premature convergence.

Ackley's test function (A.5) is symmetric and very bumpy. Its number of local minima increases exponentially with the problem dimension. Ackley's function has a global optimum with very strong local features. All (1,10) algorithms stagnate in local minima (Figure 12) like for the Step function. POI achieves again here the best result and MMP performs worse than the standard ES. When applying a bigger population size (5,35) on Ackley's function (Figure 13) the standard ES and POI get stable against premature convergence but the low performance of MMP remains. POI has the best convergence speed.

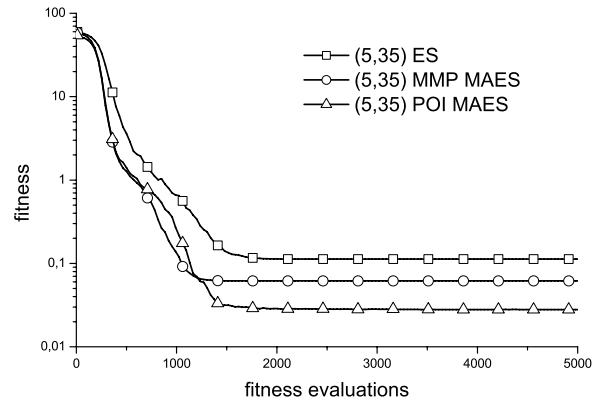


Figure 11: 5-dim Griewank function: fitness of best individual ($\mu = 5, \lambda = 35$).

The obtained results show, that the application of model assistance by GP using only the simple pre-selection criterion MMP on multimodal problems only worsens the performance of the optimization algorithm. However the usage of the more sophisticated POI pre-selection criterion leads to a statistically significant performance enhancement.

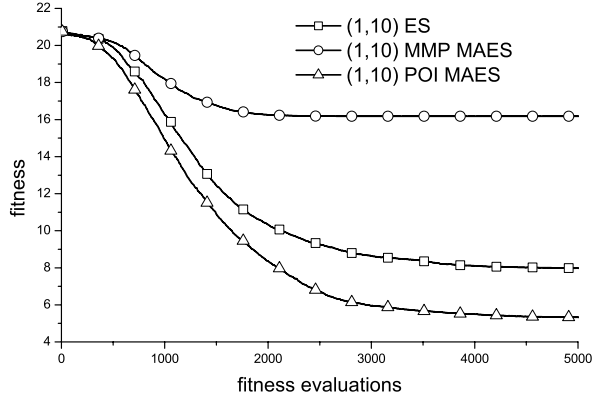


Figure 12: 20-dim Ackley's function: fitness of best individual ($\mu = 1, \lambda = 10$).

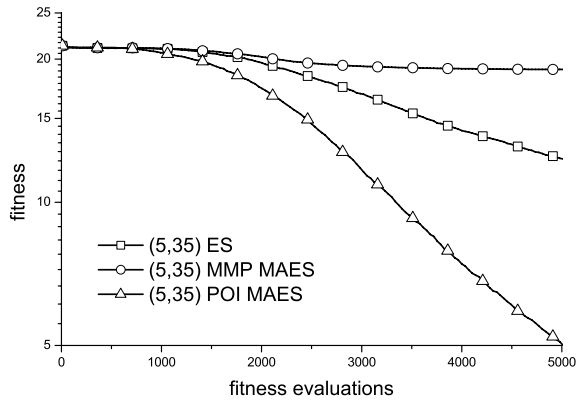


Figure 13: 20-dim Ackley's function: fitness of best individual ($\mu = 5, \lambda = 35$).

5 Conclusions

We applied a Gaussian Process as an approximation model to assist a standard ES by using the GP to pre-select the most promising individuals to be evaluated by the real fitness function. We identified the tradeoff between exploitation and exploration, which was investigated using mean of model prediction (MMP) and the probability of

improvement (POI) as pre-selection criterion.

Extensive simulations showed that both approaches enhance the performance of a standard ES on unimodal problems and MMP performs slightly better than POI, due to the higher tendency of POI to sample in unexplored areas.

On the other hand, for multimodal problems the usage of the simple MMP pre-selection fails and leads to a decrease in performance. In such cases the more sophisticated POI pre-selection criterion succeeds, shows higher convergence speed and is much more stable against premature convergence. This is reasonable, because the POI addresses the tradeoff between exploitation and exploration by utilizing the probabilistic interpretation of the GP model. These encouraging results of POI justify its application in the field of model assisted ES.

For further work it is planned to develop a mechanism which controls the impact of the approximation model on the optimization process by controlling λ_{Pre} . This can be carried out by using the confidence of the approximation model.

Acknowledgments

This research has been funded by the German federal ministry of research and education (BMBF) in the project "Entwicklung eines Systems zur automatisierten Herstellung und Charakterisierung von kristallinen Festkörpern in hohem Durchsatz" under contract No. 03C0309E.

A Test Functions

A.1 Sphere function

$$f_{Sphere}(\vec{x}) = \sum_{i=1}^n x_i^2 \quad (8)$$

$$-5.12 \leq x_i \leq 5.12 ; n = 20;$$

$$\min(f_{Sphere}) = f_{Sphere}(0, \dots, 0) = 0$$

A.2 Schwefel's function 1.2

$$f_{Schwefel}(\vec{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (9)$$

$$-5.12 \leq x_i \leq 5.12 ; n = 20;$$

$$\min(f_{Schwefel}) = f_{Schwefel}(0, \dots, 0) = 0$$

A.3 Step function

$$f_{Step}(\vec{x}) = \sum_{i=1}^n \lfloor x_i \rfloor \quad (10)$$

$$-5.12 \leq x_i \leq 5.12; n = 20;$$

$$\min(f_{Step}) = f_{Step}(0, \dots, 0) = 0$$

A.4 Griewank function

$$f_{Griewank}(\vec{x}) = 1 + \frac{1}{200} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (11)$$

$$-100 \leq x_i \leq 100; n = 20;$$

$$\min(f_{Griewank}) = f_{Griewank}(0, \dots, 0) = 0$$

A.5 Ackley's function

$$f_{Ackley}(\vec{x}) = 20 + e - 20 \exp\left(-0.2 \cdot \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) \quad (12)$$

$$-32.768 \leq x_i \leq 32.768; n = 20;$$

$$\min(f_{Ackley}) = f_{Ackley}(0, \dots, 0) = 0$$

Bibliography

- [1] M. Schonlau D. Jones and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- [2] M. A. El-Beltagy and A. J. Keane. Evolutionary optimization for computationally expensive problems using gaussian processes. In CSREA Press Hamid Arabia, editor, *Proc. Int. Conf. on Artificial Intelligence IC-AI'2001*, pages 708–714, 2001.
- [3] 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.
- [4] M. Gibbs and D. MacKay. Efficient implementation of gaussian processes. Technical report, Cavendish Laboratory, Cambridge, UK, 1997.
- [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. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 2003. In press.
- [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. Accelerating 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*. frommannholzboog, Stuttgart, 1994.
- [11] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser, Basel, 1977.
- [12] M. W. Trosset and V. Torczon. Numerical optimization using computer experiments. Technical Report TR-97-38, 1997.