Evolutionary optimization with active learning of surrogate models and fixed evaluation batch size*

Viktor Charypar[1] and Martin Holeňa[2]

[1] Czech Technical University
Faculty of Nuclear Sciences and Physical Engineering
Břehová 7, 115 19 Praha 1, Czech Republic
charyvik@fjfi.cvut.cz
[2] Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Praha, Czech Republic
martin@cs.cas.cz

Abstract. *Evolutionary optimization is often applied to problems, where simulations or experiments used as the fitness function are expensive to run. In such cases, surrogate models are used to reduce the number of fitness evaluations. Some of the problems also require a fixed size batch of solutions to be evaluated at a time. Traditional methods of selecting individuals for true evaluation to improve the surrogate model either require individual points to be evaluated, or couple the batch size with the EA generation size. We propose a queue based method for individual selection based on active learning of a kriging model. Individuals are selected using the confidence intervals predicted by the model, added to a queue and evaluated once the queue length reaches the batch size. The method was tested on several standard benchmark problems. Results show that the proposed algorithm is able to achieve a solution using significantly less evaluations of the true fitness function. The effect of the batch size as well as other parameters is discussed.*

1 Introduction

Evolutionary optimization algorithms are a popular class of optimization techniques suitable for various optimization problems. One of their main advantages is the ability to find optima of black-box functions – functions that are not explicitly defined and only their input/output behavior is known from previous evaluations of a finite number of points in the input space. This is typical for applications in engineering, chemistry or biology, where the evaluation is performed in a form of computer simulation or physical experiment.

The main disadvantage for such applications is the very high number of evaluations of the objective function (called fitness function in the evolutionary optimization context) needed for an evolutionary algorithm (EA) to reach the optimum. Even if the simulation used as the objective function takes minutes to finish, the traditional approach becomes impractical. When the objective function is evaluated using a physical experiment, in the evolutionary optimization of catalytic materials [1] for example, an evaluation for one generation of the algorithm takes between several days and several weeks and costs thousands of euros.

The typical solution to this problem is performing only a part of all evaluations using the true fitness function and using a response-surface model as its replacement for the rest. This approach is called surrogate modeling. When using a surrogate model, only a small portion of all the points that need to be evaluated is evaluated using the true objective function (simulation or experiment) and for the rest, the model prediction is assigned as the fitness value. The model is built using the information from the true fitness evaluations.

Since the fitness function is assumed to be highly non-linear the modeling methods used are non-linear as well. Some of the commonly used methods include artificial neural networks, radial basis functions, regression trees, support vector machines or Gaussian processes [3].

Furthermore, some experiments require a fixed number of samples to be processed at one time. This presents its own set of challenges for adaptive sampling and is the main concern of this paper. We present an evolutionary optimization method assisted by a variant of a Gaussian-process-based interpolating model called kriging. In order to best use the evaluation budget, our approach uses active learning methods in selecting individuals to evaluate using the true fitness function. A key feature of the approach is support for online and offline batch evaluation with arbitrary batch size independent of the generation size of the EA.

The rest of the paper is organized as follows: in the following section we introduce the kriging surrogate model and its properties, in section 2 the methods of

* This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS12/196/OHK3/3T/14 as well as the Czech Science Foundation grant 201/08/0802.

coupling a model to the evolutionary optimization are discussed, section 4 provides details of the proposed method and finally, the results of testing the method are presented and discussed in section 5.

2 Model-assisted evolutionary optimization

Since the surrogate model used as a replacement for the fitness function in the EA is built using the results of the true fitness function evaluations, there are two competing objectives. First, we need to get the most information about the underlying relations in the data, in order to build a precise model of the fitness function. If the model does not capture the features of the fitness function correctly, the optimization can get stuck in a fake optimum or generally fail to converge to a global one. Second, we have a limited budget for the true fitness function evaluations. Using many points from the input space to build a perfect model can require more true fitness evaluations than not employing a model at all.

In the general use of surrogate modeling, such as design space exploration, the process of selecting points from the input space to evaluate and build the model upon is called sampling [3]. Traditionally, the points to sample are selected upfront. Upfront sampling schemes are based on the theory of design of experiments (DoE), e.g. a Latin hypercube design. When we don't know anything about the function we are trying to model, it is better to use a small set of points as a base for an initial model, which is then iteratively improved using new samples, selected based on the information from previous function evaluations and the model itself. This approach is called adaptive sampling [3].

Using the surrogate model in an evolutionary optimization algorithm, the adaptive sampling decisions change from selecting which points of the input space to evaluate in order to improve the model to whether to evaluate a given point (selected by the EA) with the true fitness function or not. There are two general approaches to this choice: the generation-based approach and the individual-based approach. We will discuss both, with emphasis on the latter, a variant of which is used in the method we propose in section 4.

2.1 Generation-based approach

In the generation-based approach the decision whether to evaluate an individual point with the true fitness function is made for the whole generation of the evolutionary algorithm. The optimization takes the following steps.

1. An initial N_i generations of the EA is performed, yielding sets $\mathcal{G}_1, \ldots, \mathcal{G}_{N_i}$ of individuals $(\mathbf{x}, f_t(\mathbf{x}))$, f_t being the true fitness function.
2. The model M is trained on the individuals $(\mathbf{x}, f_t(\mathbf{x})) \in \bigcup_{i=1}^{N_i} \mathcal{G}_i$.
3. The fitness function f_t is replaced by a model prediction f_M.
4. T generations are performed evaluating f_M as the fitness function.
5. One generation is performed using f_t yielding a set $\mathcal{G}_j$ of individuals. (initially $j = N_i + 1$)
6. The model is retrained on the individuals $(\mathbf{x}, f_t(\mathbf{x})) \in \bigcup_{i=1}^{j} \mathcal{G}_i$
7. Steps 4–6 are repeated until the optimum is reached.

The amount of true fitness evaluations in this approach is dependent on the population size of the EA and the frequency of control generations T, which can be fixed or adaptively changed during the course of the optimization [6]. For problems requiring batched evaluation this approach has the advantage of evaluating the whole generation, the size of which can be set to the size of the evaluation batch. The main disadvantage of the generation-based strategy is that not all individuals in the control generation are necessarily beneficial to the model quality and the expensive true fitness evaluations are wasted.

2.2 Individual-based approach

As opposed to the generation-based approach, in the individual-based strategy, the decision whether to evaluate a given point using the true fitness function or the surrogate model is made for each individual separately.

In model-based optimization in general, there are several possible approaches to individual-based sampling. The most used approach in the evolutionary optimization is pre-selection. In each generation of the EA, number of points, which is a multiple of the population size, is generated and evaluated using the model prediction. The best of these individuals form the next generation of the algorithm. The optimization is performed as follows.

1. An initial set of points $\mathcal{S}$ is chosen and evaluated using the true fitness function f_t.
2. Model M is trained using the pairs $(\mathbf{x}, f_t(\mathbf{x})) \in \mathcal{S}$
3. A generation of the EA is run with the fitness function replaced by the model prediction f_M and a population $\mathcal{O}_i$ of size qp is generated and evaluated with f_M, where p is the desired population size for the EA and q is the pre-screening ratio. Initially, $i = 1$.

4. A subset $\mathcal{P} \subset \mathcal{O}$ is selected according to a selection criterion.
5. Individuals from $\mathcal{P}$ are evaluated using the true fitness function f_t.
6. The model M is retrained using $\mathcal{S} \cup \mathcal{P}$, the set $\mathcal{S}$ is replaced with $\mathcal{S} \cup \mathcal{P}$, and the EA resumes from step 3.

Another possibility, called the best strategy [5], is to replace $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{O}$ instead of just $\mathcal{P}$ in step 6 after re-evaluating the set $\mathcal{O} \setminus \mathcal{P}$ with f_M (after the model M has been re-trained). This also means using the population size qp in the EA.

The key piece of this approach is the selection criterion (or criteria) used to determine which individuals from set $\mathcal{O}$ should be used in the following generation of the algorithm. There are a number of possibilities, let us discuss the most common.

An obvious choice is selecting the best individuals based on the fitness value. This results in the region of the optimum being sampled thoroughly, which helps finding the true optimum. On the other hand, the regions far from the current optimum are neglected and a possible better optimum can be missed. To sample the areas of the fitness landscape that were not explored yet, space-filling criteria are used, either alone or in combination with the best fitness selection or other criteria.

All the previous criteria have the fact that they are concerned with the optimization itself in common. A different approach is to use the information about the model, most importantly its accuracy, to decide which points of the input space to evaluate with the true fitness function in order to most improve it. This approach is sometimes called active learning.

2.3 Active learning

Active learning is an approach that tries to maximize the amount of insight about the modeled function gained from its evaluation while minimizing the number of evaluations necessary. The methods are used in the general field of surrogate modeling as an efficient adaptive sampling strategy. The terms adaptive sampling and active learning are often used interchangeably. We will use the term active learning for the methods based on the characteristics of the surrogate model itself, such as accuracy, with the goal of minimizing the model prediction error either globally or, more importantly, in the area of the input space the EA is exploring.

The active learning methods are most often based on the local model prediction error, such as cross-validation error. Although some methods are independent of the model, for example the LOLA-Voronoi method [2], most of them depend on the model used. The kriging model used in our proposed method offers a good estimate of the local model accuracy by giving an error estimate of its prediction. It is possible to use the estimate itself as a measure of the model's confidence in the prediction, or base a more complex measure on the variance estimate. The measures that were tested for use in our method will be described in detail in section 4.1.

3 Kriging meta-models

The kriging method is an interpolation method originating in geostatistics [9], based on modeling the function as a realization of a stochastic process [11].

In the ordinary kriging, which we use, the function is modeled as a realization of a stochastic process

$$Y(\mathbf{x}) = \mu_0 + Z(\mathbf{x}) \tag{1}$$

where $Z(\mathbf{x})$ is a stochastic process with mean 0 and covariance function $\sigma^2\psi$ given by

$$cov\{Y(\mathbf{x}+\mathbf{h}), Y(\mathbf{x})\} = \sigma^2\psi(\mathbf{h}), \tag{2}$$

where σ^2 is the process variance for all $\mathbf{x}$. The correlation function $\psi(\mathbf{h})$ is then assumed to have the form

$$\psi(\mathbf{h}) = \exp\left[-\sum_{l=1}^{d} \theta_l |\mathbf{h}_l|^{p_l}\right], \tag{3}$$

where $\theta_l, l = 1, \ldots, d$, where d is the number of dimensions, are the correlation parameters. The correlation function depends on the difference of the two points and has the intuitive property of being equal to 1 if $\mathbf{h} = \mathbf{0}$ and tending to 0 when $\mathbf{h} \to \infty$. The θ_l parameters determine how fast the correlation tends to zero in each coordinate direction and the p_l determines the smoothness of the function.

The ordinary kriging predictor based on n sample points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with values $\mathbf{y} = (y_1, \ldots, y_n)'$ is then given by

$$\hat{y}(\mathbf{x}) = \hat{\mu_0} + \psi(\mathbf{x})'\mathbf{\Psi}^{-1}(\mathbf{y} - \hat{\mu_0}\mathbf{1}), \tag{4}$$

where $\psi(\mathbf{x})' = (\psi(\mathbf{x}-\mathbf{x}_1), \ldots, \psi(\mathbf{x}-\mathbf{x}_n))$, $\mathbf{\Psi}$ is an $n \times n$ matrix with elements $\psi(\mathbf{x}_i - \mathbf{x}_j)$, and

$$\hat{\mu_0} = \frac{\mathbf{1}'\mathbf{\Psi}^{-1}\mathbf{y}}{\mathbf{1}'\mathbf{\Psi}^{-1}\mathbf{1}}. \tag{5}$$

An important feature of the kriging model is that apart from the prediction value it can estimate the prediction error as well. The kriging predictor error in point $\mathbf{x}$ is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2\left[1 - \psi'\mathbf{\Psi}^{-1}\psi + \frac{(1 - \psi'\mathbf{\Psi}^{-1}\psi)^2}{\mathbf{1}'\mathbf{\Psi}^{-1}\mathbf{1}}\right] \tag{6}$$

where the kriging variance is estimated as

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \hat{\mu_0}\mathbf{1})\mathbf{\Psi}^{-1}(\mathbf{y} - \hat{\mu_0}\mathbf{1})}{n}. \quad (7)$$

The parameters θ_l and p_l can be estimated by maximizing the likelihood function of the observed data.

For the derivation of the equations 4 - 7 as well as the MLE estimation of the parameters the reader may consult a standard stochastic process based derivation by Sacks et al. in [11] or a different approach given by Jones in [7].

4 Method description

In this section we will describe the proposed method for kriging-model-assisted evolutionary optimization with batch fitness evaluation. Our main goal was to decouple the true fitness function sampling from the EA iterations based on an assumption that requiring a specific number of true fitness evaluations in every generations of the EA forces unnecessary sampling.

In the generation-based approach, some of the points may be unnecessary to evaluate, as they will not bring any new information to the surrogate model. The individual-based approach is better suited for the task, as it chooses those points from each generation, which are estimated to be the most valuable for the model. There is still the problem of performing a given number of evaluations in every generation, although there might not be enough valuable points to select from.

The method we propose achieves the desired decoupling by introducing an evaluation queue. The evolutionary algorithm uses the model prediction at all times and when a point, in which the model's confidence in its prediction is low, is encountered, it is added to the evaluation queue. Once there are enough points in the queue, all the points in it are evaluated and the model is re-trained using the results. The optimization takes the following course.

1. Initial set $\mathcal{S}$ of b samples is selected using a chosen initial design strategy and evaluated using the true fitness function f_t.
2. An initial kriging model M is trained using pairs $(\mathbf{x}, f_t(\mathbf{x})) \in \mathcal{S}$.
3. The evolutionary algorithm is started, with the model prediction f_M as the fitness function.
4. For every prediction $f_M(\mathbf{x}) = \hat{y}_M(\mathbf{x})$, an estimated improvement measure $c(s^2_M(\mathbf{x}))$ is computed from the error estimate $s^2_M(\mathbf{x})$. If $c(s^2_M(\mathbf{x})) > t$, an improvement threshold, the point is added to the evaluation queue $\mathcal{Q}$.
5. If the queue size $|\mathcal{Q}| \geq b$, the batch size, all points $\mathbf{x} \in \mathcal{Q}$ are evaluated, the set $\mathcal{S}$ is replaced by $\mathcal{S} \cup \{(\mathbf{x}, f_t(\mathbf{x})\}$ and the EA is resumed.
6. Steps 4 and 5 are repeated until the goal is reached, or a stall condition is fulfilled.

The b and t parameters, as well as the function $c(s^2)$, are chosen before running the optimization. Note that the evaluation in step 5 can be performed either immediately, i.e. online, or offline. In offline evaluation, after filling the evaluation queue, the EA is stopped when the current iteration is finished and the control is returned to the user. After obtaining the fitness values for the samples in the sample queue (e.g. by performing an experiment), the user can manually add the samples and resume the EA from the last generation.

While the choice of the parameters will be discussed in section 5, let us introduce three different measures of estimated improvement in the model prediction $c(s^2(\mathbf{x}))$ which we tested – the standard deviation, the probability of improvement and the expected improvement.

4.1 Measures of estimated improvement

To estimate the improvement, which evaluation of a given point will bring, we can use several measures. The three measures introduced here are all based on the prediction error estimate of the kriging model. The goal of these measures is to prefer the points that help improve the model in regions explored by the EA.

Each of the measure's results for a given point are compared with a threshold and when the estimated improvement is above the threshold, the point is evaluated using the true fitness function.

Standard deviation (STD) is the simplest measure we tested. It is computed directly from the error as its square root

$$STD(x) = \sqrt{\hat{s^2_M}(\mathbf{x})}. \quad (8)$$

The STD captures only the model's estimate of the error of its own prediction (based on the distance from the known samples). As such, it does not take into account the value of the prediction itself and can be considered a measure of the model accuracy.

Probability of improvement (POI) [7] uses the fact, that the kriging prediction is a Gaussian process and the prediction in a single point is therefore a normally distributed random variable $Y(\mathbf{x})$ with a mean and variance given by the kriging predictor. If we choose a target T (based on the goal of the optimization), we can estimate the probability that a given point will have a value $y(\mathbf{x}) \leq T$ as a probability that $Y(\mathbf{x}) \leq T$. The probability of improvement is therefore defined as

$$POI(x) = \Phi\left(\frac{T - \hat{y}_M(\mathbf{x})}{s^2_M(\mathbf{x})}\right), \quad (9)$$

where Φ is the cumulative distribution function of the standard normal distribution. As opposed to the STD, the POI takes into account the prediction mean (value) as well as its variance (error estimate). The area of the current optimum is therefore preferred over the rest of the input space. When the area of the current optimum is sampled enough, the variance becomes very small and the term $\frac{T-\hat{y}_M(x)}{s^2_M(x)}$ becomes extremely negative, encouraging the sampling of less explored areas.

Expected improvement (EI) [7, 8] is based on estimating, as the name suggests, the improvement we expect to achieve over the current minimum $f_{\min}$, if a given point is evaluated. As before, we assume the model prediction in point $\mathbf{x}$ to be a normally distributed random variable $Y(\mathbf{x})$ with a mean and variance given by the kriging predictor. We achieve an improvement I over $f_{\min}$ if $Y(\mathbf{x}) = f_{\min} - I$. As shown in [7] the expected value of I can be obtained using the likelihood of achieving the improvement

$$\frac{1}{\sqrt{2\pi}s^2_M(\mathbf{x})}\int_{I=0}^{I=\infty}\exp\left[-\frac{(f_{\min}-I-\hat{y}_M(\mathbf{x})^2}{2s^2_M(\mathbf{x})}\right] \quad (10)$$

Expected improvement is the expected value of the improvement found by integrating over this density. The resulting measure EI is defined as

$$EI(\mathbf{x}) = E(I) = s^2_M(\mathbf{x})[u\Phi(u)+\phi(u)],\phi(u)], \quad (11)$$

where

$$u = \frac{f_{\min}-\hat{y}_M(\mathbf{x})}{s^2_M(\mathbf{x})} \quad (12)$$

and Φ and ϕ are the cumulative distribution function and the probability distribution functions of the normal distribution respectively. The expected improvement has an important advantage over the POI: it does not require a preset target T, which can be detrimental to the POI's successful sample selection when set too high or too low.

All three measures have an important weakness of being based on the model prediction. If the modeled function is deceptive, the model can be very inaccurate while estimating a low variance. A good initial sampling of the fitness function is therefore very important. The success of the whole method is dependent on the model's ability to capture the response surface correctly and thus on the function itself.

5 Results and discussion

The proposed method was tested using simulations on three standard benchmark functions. We studied the model evolution during the course of the optimization, the effect of the parameters and also investigated the optimal choice of batch size for problems where an upfront choice is possible. In this section we discuss the tests performed and their results.

For testing, we used the genetic algorithm implementation from the global optimization toolbox for the Matlab environment and the implementation of an ordinary kriging model from the SUMO Toolbox [4]. The parameters of the supporting methods, e.g. the genetic algorithm itself, were kept on their default values provided by the implementation.

Because the EA itself is not deterministic, each test was performed 20 times and the results we present are statistical measures of this sample. As a performance measure we use the number of true fitness evaluations used to reach a set goal in all tests. The main reason to use this measure is that in model-assisted optimization the computational cost of everything except the true fitness evaluation is minimal in comparison. We also track the proportion of the 20 runs that reached the goal before various limits (time, stall, etc.) took effect.

5.1 Benchmark functions

Since the evolutionary algorithms and optimization heuristics in general are often used on black-box optimization, where the properties of the objective function are unknown, it is not straightforward to asses their quality on real world problems. It has therefore become a standard practice to test optimization algorithms and their modifications on specially designed testing problems.

These benchmark functions are explicitly defined and their properties and optima are known. They are often designed to exploit typical weaknesses of optimization algorithms in finding the global optimum. We used three functions found in literature [10]. Although we performed our tests in two dimensions we give general multi-dimensional definitions of the functions.

First of the functions used is the De Jong's function. It is one of the simplest benchmarks, it is continuous, convex and unimodal and is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \quad (13)$$

The domain is restricted to a hypercube $-10 \leq x_i \leq 10, i = 1, \ldots, n$. The function has one global optimum $f(\mathbf{x}) = 0$ in point $\mathbf{x} = \mathbf{0}$. The De Jong's function was primarily used as a proof of concept test.

As a second benchmark, we used the Rosenbrock's function, also called Rosenbrock's valley. The global optimum is inside a long parabolic shaped valley, which is easy to find. Finding the global optimum in

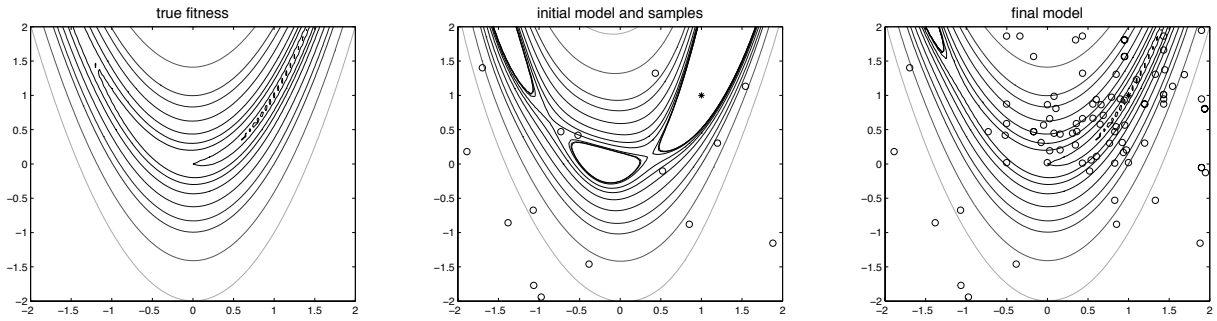


Fig. 1. The original fitness function, the initial model and the final model.

that valley however is difficult [10]. The function has the following definition

$$f(\mathbf{x}) = \sum_{i=1}^{n} [100(x_{i+1} + x_i^2)^2 + (1 - x_i)^2] \qquad (14)$$

The domain of the function is restricted to a hypercube $-2 \leq x_i \leq 2, i = 1, \ldots, n$. It has one global optimum $f(\mathbf{x}) = 0$ in $\mathbf{x} = \mathbf{1}$.

Finally, the third function used as a benchmark is the Rastrigin's function. It is based on the De Jong's function with addition of cosine modulation, which produces a high number of regularly distributed local minima and makes the function highly multimodal. The function is defined as

$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)] \qquad (15)$$

The domain is restricted to $-5 \leq x_i \leq -5$, $i = 1, \ldots, n$. The global optimum $f(\mathbf{x}) = 0$ is in $\mathbf{x} = \mathbf{1}$.

5.2 Model evolution

As the basic illustration of how the model evolves during the course of the EA, let us consider an example test run using the Rosenbrock's function. For this experiment we set the batch size of 15, used the STD measure of estimated improvement with a threshold of 0.001 and set the target fitness value of 0.001 as well. The target was reached at the point $(0.9909, 0.9824)$ using 90 true fitness evaluations. A genetic algorithm without a surrogate model needed approximately 3000 evaluations to reach the goal in several test runs.

The model evolution is shown in figure 1. The true fitness function is shown on the left, the initial model is in the middle and the final model on the right. The points where the true fitness function was sampled are denoted with circles an the optimum is marked with a star.

function	ev (1q)	ev (med)	ev (3q)	goal	reached
De Jong	60	60	120	0.01	1
Rosenbrock	60	125	310	0.1	1
Rastrigin	260	370	580	0.1	0.85

Table 1. GA performance on benchmark functions without a model.

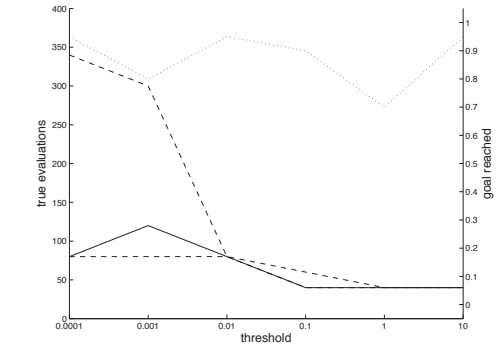
5.3 Measures of estimated improvement comparison

In order to compare the measures of estimated improvement, we performed simulations on each benchmark using each improvement estimate measure with different values of the threshold. The batch size was set to 40 – generally found to be the ideal batch size – for these experiments. For comparison, we also performed tests with the standard genetic algorithm without a model. Results of these simulations are shown in the table 1.

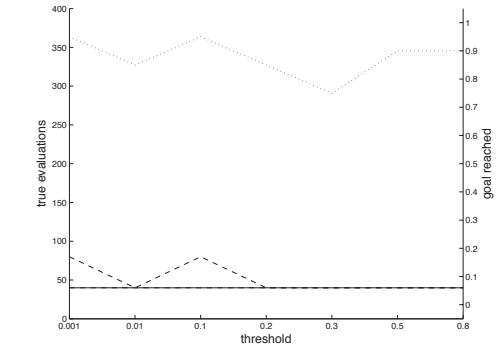
The De Jong's function proved to be simple to optimize and the threshold setting did not have almost any effect. Only when using the standard deviation, setting the threshold too low lead to an increase in the number of evaluations, as too many points were evaluated, although the model prediction in those points was accurate enough.

The same is true for the STD measure used on the Rosenbrock's function, where setting the threshold too low leads to a big increase in variance of the results. Interestingly, setting the threshold too high leads to a decrease in the number of evaluations, but also in the success rate of reaching the goal. The POI and EI are more stable in terms of true fitness evaluations, but have worse overall success rate. The results are shown in figure 2

The Rastrigin's function proved difficult to optimize. This is probably due to the locality of the kriging model and the high number of local minima of the function. Overall the STD measure is the most suc-



(a) STD



(b) POI

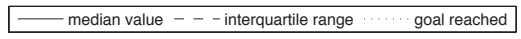


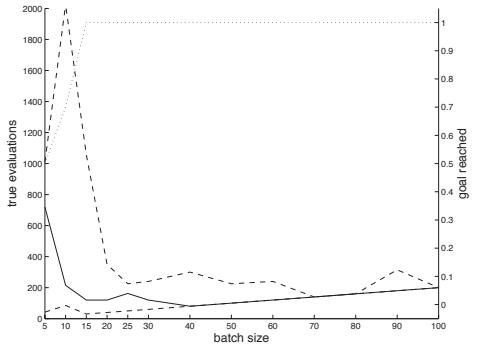
Fig. 2. STD measure on the Rosenbrock's function - true fitness evaluations and proportion of runs reaching goal.

cessful. POI and EI lead to bad sampling of the model and failure to reach the optimum.

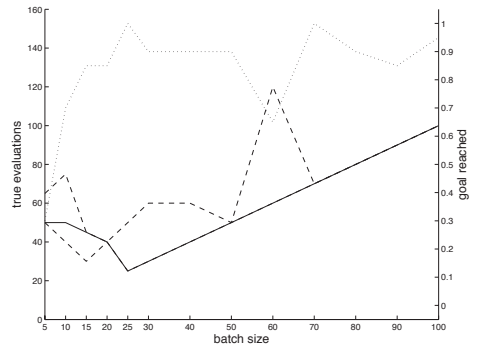
An interesting general result is that the more complex measures of estimated improvement perform worse than the simple standard deviation estimate. This indicates that the goal of active learning selection criteria in the evolutionary optimization should be the best possible sampling for overall model accuracy, as opposed to trying to improve the accuracy in the best regions of the input space. Both the POI and EI are design to select next best points to reach the optimum. Since in our case, this is handled by the EA itself, the measures bring an unnecessary noise to the estimate of the model accuracy. The results also show that the best measure selection is dependent on the optimized function.

5.4 Batch size

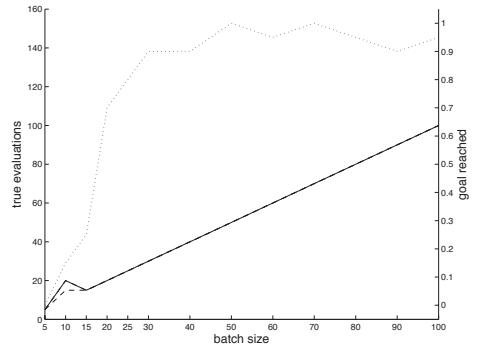
In order to study the batch size effect on the optimization, a number of experiments were performed with different batch sizes. The only option to achieve



(a) No model



(b) STD



(c) POI

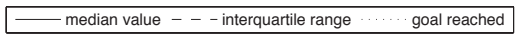


Fig. 3. Batch size effect on Rosenbrock's function optimization - true fitness evaluations and proportion of runs raching the goal.

a given batch size is to set the population size in a standard GA, in our method however, the settings are independent so a population size of 30, which proved efficient, was used in all of the tests.

The results on the De Jong's functions show that apart from small batch sizes (up to 10), the optimization is successful in all runs. Our method helps stabilize the EA for small batch sizes and for batch sizes above 15 the algorithm finds the optimum using

a single batch. For a standard GA this strong dependence arises for batch sizes above 40 and the algorithm reaches the goal in the second generation, evaluating twice as many points.

For the Rosenbrock's function we get the intuitive result that setting the batch size too low leads to more evaluations or a failure to reach the goal, while large batch size do not improve the results and waste true fitness evaluations. For this function the POI proved to be the most efficient measure. The comparison is shown in figure 3. Overall the method reduces the number of true evaluations from hundreds to tens for the Rosenbrock's function, while slightly reducing the success rate of the computation.

The Rastrigin's function proved difficult to optimize even without a surrogate model. With the model, the STD achieved the best results reducing the number of true fitness evaluations approximately three times in the area of the highest success rate with batch size of 70. The other two measures were ineffective. We attribute the method's difficulty optimizing the Rastrigin's function to the fact that the kriging model is local and thus it requires a large number of samples to capture the function's complicated behavior in the whole input space. When the initial sampling is misleading, which is more likely for the Rastrigin's function, both the model prediction and estimated improvement are wrong.

The results suggest that best batch size and best estimated improvement measure are highly problem-dependent. The proposed method is also very sensitive to good initial sample selection, which is the most usual reason for it to fail to find the optimum. The experimental results support the intuition that batches too small are bad for the initial sampling of the model and batches too large slow down the model improvement by evaluating points that it would not be necessary to evaluate with smaller batches. This suggests using a larger initial sample and a small batch for the rest of the optimization.

6 Conclusions

In this paper we presented a method for model-assisted evolutionary optimization with a fixed batch size requirement. To decouple the sampling from the EA iterations and support an individual-based approach while keeping a fixed evaluation batch size, the method uses an evaluation queue. The candidates for true fitness evaluations are selected by an active learning method using a measure of estimated improvement of the model quality based on the model prediction error estimate.

The results suggest using simple methods for improvement estimate in active learning, which only capture information about the model accuracy improvement expected by sampling a given point. In the experiments with the batch size we found that small batch sizes perform better when the objective function is simple, while causing bad initial sampling of more complex functions, suggesting using a larger initial sample. The future development of this work should include experiments using different batch sizes for initial sampling and comparison of the method with other ways of employing a surrogate model in the optimization as well as other model-assisted optimization methods.

The method brings promising results, reducing the number of true fitness evaluations to a large degree for some of the benchmark functions. On the other hand, its success is highly dependent on the optimized function and its initial sampling.

References

1. M. Baerns, M. Holeňa: *Combinatorial development of solid catalytic materials: design of high-throughput experiments, data analysis, data mining.* Catalytic Science Series. Imperial College Press, 2009.
2. K. Crombecq, L. De Tommasi, D. Gorissen, T. Dhaene: *A novel sequential design strategy for global surrogate modeling.* In Winter Simulation Conference, WSC '09, Winter Simulation Conference, 2009, 731–742.
3. D. Gorissen: *Grid-enabled adaptive surrogate modeling for computer aided engineering.* PhD Thesis, Ghent University, University of Antwerp, 2009.
4. D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, K. Crombecq: *A surrogate modeling and adaptive sampling toolbox for computer based design.* The Journal of Machine Learning Research 11, 2010, 2051–2055.
5. L. Gräning, Y. Jin, B. Sendhoff: *Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study.* In ESANN, 2005, 273–278.
6. Y. Jin, M. Olhofer, B. Sendhoff: *Managing approximate models in evolutionary aerodynamic design optimization.* In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, vol. 1, IEEE, 2001, 592–599.
7. D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.
8. D. R. Jones, M. Schonlau, W.J. Welch: *Efficient global optimization of expensive black-box functions.* Journal of Global Optimization 13, 1998, 455–492.
9. G. Matheron: *Principles of geostatistics.* Economic Geology 58(8), 1963, 1246–1266.
10. M. Molga, C. Smutnicki: *Test functions for optimization needs.* Test Functions for Optimization Needs, 2005.
11. J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn: *Design and analysis of computer experiments.* Statistical Science 4(4), 1989, 409–423.