From Expected Improvement to Investment Portfolio Improvement: Spreading the Risk in Kriging-based Optimization

Rasmus K. Ursem

Research & Technology, Grundfos Management A/S,
Poul Due Jensens Vej 7, DK-8850 Bjerringbro, Denmark
ursem@cs.au.dk

Abstract. The increasing use of time-consuming simulations in the industry has spawned a growing interest in coupling optimization algorithms with fast-to-compute surrogate models. A major challenge in this approach is to select the approximated solutions to evaluate on the real problem. To address this, the Kriging meta-model offers both an estimate of the mean value and the standard error in an unknown point. This feature has been exploited in a number of so-called prescreening utility functions that seek to maximize the outcome of an expensive evaluation. The most widely used are the Probability of Improvement (PoI) and Expected Improvement (ExI) functions.

This paper studies this challenge from an investment portfolio point-of-view. In short, the PoI favors low risk investments whereas the ExI promotes high risk investments. The paper introduces the investment portfolio improvement (IPI) approach as a strategy mixing the two extremes. The novel approach is applied to seven benchmark problems and two real world examples from the pump industry.

Keywords: expected improvement, prescreening methods, Kriging

1 Introduction

During the last couple of decades, the increasing use of time-consuming simulations in engineering-related industries poses a serious challenge to optimization algorithms. To address this challenge, researchers have studied a number of surrogate models that allow fast evaluation of an approximation of the real problem. This approximation is typically build from a low number of sample points of the real problem. Approximation models can be evaluated in a few hundred milliseconds, which is significantly faster than, e.g., a 5 hour flow simulation of a centrifugal pump. However, the result is only an approximation and there are usually differences between the real problem and the approximation of it. Thus, the optimization specialist trades evaluation accuracy for evaluation speed.

The Kriging meta-model has become increasingly popular as it provides both a mean value and standard error of the approximation of an unknown point. By viewing the standard error as a confidence interval for the approximation, authors have proposed to maximize a *prescreening* function expressing the potential improvement gained by performing the expensive/time-consuming evaluation of the real function. Three widely used prescreening functions are the Lower Bound (LB) approach by Dennis and Torczon [1], the Probability of Improvement (PoI) introduced by Ulmer et al. [2], and the Expected Improvement (ExI) popularized as the Efficient Global Optimization (EGO) algorithm by Jones et al. [3]. The ExI is the most popular and it has been extensively used for solving numerous real-world problems, e.g., [4–6]. The focus in this paper is on single-objective problems, but the ExI function has also been adapted to multi-objective problems, e.g., [5, 7].

The aim of this paper is twofold. First, to study and discuss LB, PoI, and ExI in relation to their theoretical ability to provide a return-of-investment, i.e., an improved solution to the problem. Second, to suggest an algorithm for optimizing a portfolio of solutions that spread the risk of investment by using the novel prescreening approach called *Investment Portfolio Improvement.* The algorithm is based on the Differential Evolution (DE) algorithm [10, 11].

The paper is structured as follows. Section 2 introduces the optimization methodology, in particular the Differential Evolution algorithm and the Kriging meta-model. Subsection 2.3 elaborates on LB, PoI, and ExI in relation to the investment strategy these prescreening functions implement. Subsection 2.4 describes the novel algorithm. Following this, section 3 introduces the experimental setup and the optimization problems. Section 4 contains the results and a discussion of these. Finally, section 5 concludes the paper.

2 Kriging-based prescreening optimization

Successful application of a Kriging-based prescreening optimization algorithm to a time-consuming or costly real-world problem involves three main decisions. First, the choice of the optimization algorithm. Second, the choice of the Kriging variant. Third, the choice of the prescreening function.

Regarding optimization algorithms, a good choice is the differential evolution (DE) algorithm suggested by Storn and Price in 1995 [10, 11]. Since then, it has become widely accepted as one of the best algorithms for numerical optimization as it has proven its worth on numerous problems, e.g., [4, 8, 9].

Concerning Kriging, numerous variants have been described and tested in the literature and choosing the best variant can be a challenge on its own [12, 13]. From an optimization perspective, the main choice is a trade-off between accuracy and ability to find a new solution that is better than the best known point. The use of prescreening methods may reduce the disadvantages of choosing a sub-optimal Kriging variant substantially as the methods allow the algorithm to employ an explorative search behavior. Thus, the prescreening method allows a choice of Kriging variant that does not return large overshoots of the best known point. For this reason, this paper uses simple Kriging with a “conservative” kernel function.

2.1 Differential evolution

The algorithm presented in this paper is based on the $rand/1/bin$ standard DE scheme. However, it is out of the scope to provide a detailed description of DE. Instead, see the original work of Storn and Price [10, 11] or refer to Ursem [9] for a shorter version.

2.2 Kriging

As mentioned, a large number of Kriging variants exists. Kriging has been described many times in the literature and a full mathematical description is beyond the scope of this paper. Instead, see e.g. [14]. In short, Kriging predicts a normal distribution $Y(\mathbf{x}) \sim N(\hat{y}, \hat{s})$ as an interpolation based on a so-called kernel function of the distances to a number of known points.

The used Kriging model is based on the DACE Matlab toolbox by Lophaven et al. [15] with the Exp kernel function. This kernel function was tested in preliminary runs and showed the desired absence of extreme prediction values.

2.3 Prescreening procedures

The main idea behind prescreening functions is to utilize the standard error of Kriging to assess the potential improvement achieved by evaluating an unknown point $\mathbf{x}$. The most widely used prescreening functions are the Lower Bound (LB), the Probability of Improvement (PoI), and the Expected Improvement (ExI).

$$\text{LB}(\mathbf{x}) = \hat{y}(\mathbf{x}) - w \cdot \hat{s}(\mathbf{x}) \tag{1}$$

$$\text{PoI}(\mathbf{x}) = P(Y(\mathbf{x}) \leq f_{min}) = \int_{-\infty}^{f_{min}} \phi(Y(\mathbf{x}))dY = \Phi\left(\frac{f_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) \tag{2}$$

$$\begin{aligned}\text{ExI}(\mathbf{x}) &= \int_{-\infty}^{f_{min}} (f_{min} - y)\phi\left(\frac{y - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) dy & (3)\\ &= (f_{min} - \hat{y}(\mathbf{x}))\Phi\left(\frac{f_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x})\phi\left(\frac{f_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) & (4)\end{aligned}$$

In this, assume we are using a Kriging model to minimize a function $f(\mathbf{x})$ with the best known function value f_{min}. Here, $\hat{y}(\mathbf{x})$ is the approximated value with the corresponding standard error $\hat{s}(\mathbf{x})$, and for LB is w a user-defined weight. Furthermore, $\phi(\cdot)$ is the probability density function of the normal distribution, and $\Phi(\cdot)$ is the cumulative distribution function.

Over the years, several papers have been published that extend these prescreening functions. To control the balance between global and local search, Schonlau et al. [16] introduced the g parameter and the Generalized Expected Improvement (GEI). Following this, Sasena et al. [17] suggested to use an annealing scheme to control the g parameter of the GEI prescreening function. Authors have also suggested to parallelize the use of ExI. For example, Janusevskis et al. [18] suggested the q-EI as a way to generate multiple points to evaluate in each optimization run thereby lowering the number of runs. In a similar direction, Ponweiser et al. [19] introduced the MGEI and the CMGEI criteria and compared these with the annealing of Sasena et al. [17]. In this comparison, GEI turned out to be the best approach. Although the parallelization of ExI is interesting, saving optimization run-time is less relevant as 90-95% of the computation time is typically spent on the actual evaluation of new solutions.

At this point, it may be worthwhile to take a step back and study the functions from a more mathematical perspective. In the following, assume (WLOG) that $f_{min} = 0.0$ and that both function values and standard errors have been normalized. Thus, a negative $\hat{y}$ corresponds to a value that is better than the best known solution. Figure 1 displays the surface plot of PoI and ExI for different values of $\hat{y}$ and $\hat{s}$. The plot for LB is omitted as it resembles the plot for ExI. The contour lines illustrate solutions that are considered equal by the plotted prescreening function.

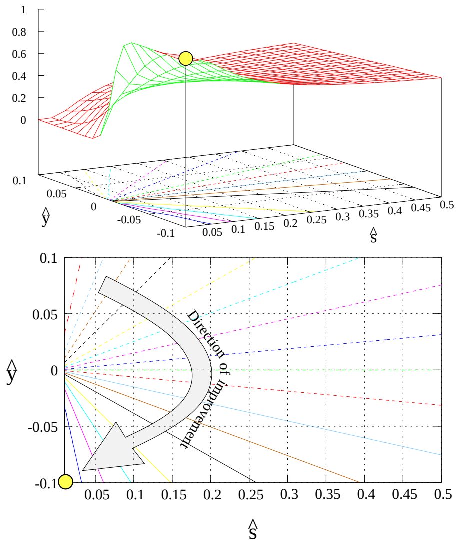

Fig. 1. Surface and contour plots of PoI (left) and ExI (right). Yellow circle is max (in the given range).

As seen in the figure, PoI has maximum when (or rather if) the optimization algorithm achieves a solution that outcompetes the current best known, i.e., $\hat{y} < 0$, *and* the standard deviation $\hat{s}$ is small. In case the evaluated solution is worse than the best known, i.e., $\hat{y} > 0$, then PoI favors solutions with large standard deviation $\hat{s}$. Furthermore, the PoI function is 0.5 for all values of $\hat{s}$ when $\hat{y} = 0$. In contrast to PoI, the ExI function clearly favors solutions with the largest possible $\hat{s}$ regardless of the approximated mean $\hat{y}$.

In an investment perspective, PoI implements a *low risk strategy* as it promotes a low standard deviation when a solution with a potential improvement ($\hat{y} < 0$) is found. For example, given two solutions x_1 and x_2 both with $\hat{y} = -0.05$ then PoI will favor the solution with lowest $\hat{s}$. In contrast, ExI represents a *high risk strategy* as ExI (and also LB) prefer unknown points that maximize the standard deviation. Considering financial investment as another field involving risk strategies, one general recommendation is to employ a *risk spreading strategy*. Thus, the strategies implemented by LB, PoI, and ExI are in conflict with this general recommendation.

2.4 Investment portfolio improvement prescreening

The main idea in investment portfolio improvement (IPI) prescreening is to optimize toward a target standard deviation t. This idea can be represented by numerous IPI functions. In this study, a number of functions were tested in preliminary runs. The IPI function defined in equation 5 turned out to have the best performance on the standard benchmark functions introduced in section 3.

$$IPI(\mathbf{x}) = 0.5 \cdot \Phi\left(\frac{f_{min} - \hat{y}}{1.05 - t}\right) + \Phi\left(\frac{-(\hat{s} - t)^2}{0.05}\right) \tag{5}$$

Figure 2 shows the function for $t = 0.4$ and $t = 0.8$. For a low t, the function promotes a search toward local improvements of the best known point. For a high t, the function primarily induces a search for a solution with the desired standard deviation and secondly an improvement over best known solution. The functions studied in preliminary runs did not impose a sufficiently strong selection pressure for improvement, i.e., the found solutions had the desired standard deviation, but with suboptimal performance.

To perform actual *portfolio optimization*, the algorithm needs to search with multiple values for t simultaneously. Naturally, this can be done in numerous ways. For instance, one may use an island model [20], a cellular EA [20], a multinational model [21], or other variants of diversity maintaining techniques. In this study, the target values t are set from the individual's population index i in the differential evolution algorithm. Thus, the IPI function for individual i at iteration j is defined as in equation 6.

$$IPI(\mathbf{x}_i, j) = 0.5 \cdot \Phi\left(\frac{f_{min} - \hat{y}(\mathbf{x}_i)}{1.05 - t_i}\right) + \Phi\left(\frac{-(\hat{s}(\mathbf{x}_i)/n_f - t_i)^2}{0.05}\right) \tag{6}$$

$$t_i = \frac{i}{popsize - 1} \qquad \text{(target for individual i)} \tag{7}$$

$$n_f = \max_i(\hat{s}(p_{i,j-1})) \qquad \text{(normalization factor)} \tag{8}$$

In this, the calculation of the normalization factor is based on measurements from the previous generation $j - 1$ where $p_{i,j-1}$ denotes parent i. Optimizing under this scheme results in a population where the low-index individuals seek a low standard deviation and high-index individuals have a high standard deviation.

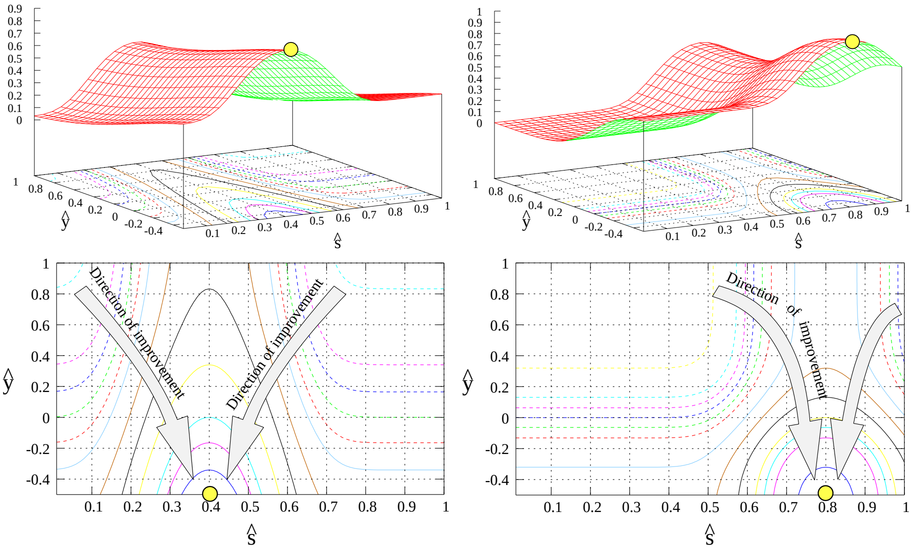

Fig. 2. Surface and contour plots for IPI with $t = 0.4$ and $t = 0.8$. Yellow circle is max.

In a real-world application, the resources for evaluating found solutions are often limited by the time and costs required to perform the full evaluation. The optimization specialist is therefore often only interested in 3-5 distinct candidate solutions per round [22]. The IPI algorithm therefore returns a user-defined number of candidates K_{NC} as follows. After the stopping criterion is met, the algorithm divides the population into K_{NC} segments and return the highest IPI scoring individual from each segment. For example, setting $K_{NC} = 3$ will result in a low-risk, a mid-risk, and a high-risk solution taken from the first third, the middle third, and the last third of the population.

3 Experimental setup

The experiments focus on comparing the performance of the novel IPI algorithm with the established PoI and ExI prescreening functions, and a DE version of the annealing GEI algorithm [17]. To complete the picture, the DE algorithm is also optimizing only the mean value (OMV), thereby allowing comparison with the traditional approach. The GEI prescreening function is defined according to equation 9 and g values are set from table 1 as in [17].

$$E(I^0) = P(y < f_{min}) = \Phi\left(\frac{f_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) = \Phi(u) \quad g = 0 \tag{9}$$

$$E(I^g) = \hat{s}(\mathbf{x})^g \sum_{k=0}^{g} (-1)^k \binom{g}{k} u^{g-k} T_k \qquad g = 1, 2, \ldots$$

$$T_0 = \Phi(u),\; T_1 = -\Phi(u),\; T_k = -u^{k-1}\phi(u) + (k-1)T_{k-2} \quad k = 2, 3, \ldots$$

Table 1. Annealing values for g depending on number of iterations.

Iteration	1-4	5-9	10-19	20-24	25-34	$\geq$35
g	20	10	5	2	1	0

The DE algorithm was run with 100 individuals, $CR = 0.2$, $F = 0.35$, and 1000 iterations to ensure convergence. In the IPI algorithm, the number of candidates was set to $K_{NC} = 3$ as this represent a typical number of simulations that can be performed per day for an industrial problem. Thus, the algorithm returned a low-risk, a medium-risk, and a high-risk solution.

The test suite includes seven benchmark problems and two model-calculated pump design problems. The tested benchmark problems are the well-known[1] Branin function, the six hump camel back function, the Hartmann 3D function, the Hartmann 6D function, the Colville function, the Rastrigin 2D function, and the less known Sasena "mystery" function [23], which is defined in equation 10.

$$\begin{aligned} \min f(x_1, x_2) = {} & 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 + \\ & 7\sin(0.5x_1) \cdot \sin(0.7x_1x_2) \\ & x_i \in [0:5], i = 1, 2 \end{aligned} \tag{10}$$

The benchmark problems are chosen to represent engineering-like problems that typically have a few local optima and a single global optimum. However, the Rastrigin function does not fulfill this selection criterion as it has 11 optima per dimension (Rastrigin 2D has 120 local optima and one global). Nevertheless, it is included to investigate the performance on a simple problem often used in traditional tests of optimization algorithms.

The two model-calculated pump design problems are based on classic pump textbook theory [24] coupled with in-house loss models for modeling the Grundfos pumps. The details of the two pump design problems cannot be revealed as it would violate the need for business confidentiality. However, the first problem has 6 design parameters and the objective is to maximize the hydraulic efficiency in the design point. The second problem has 12 design parameters and the objective is also to maximize the hydraulic efficiency in the design point.

All nine problems are fast to calculate and allow a statistical comparison of the methods based on 20 repetitions each executed as follows:

1. Generate 20 random solutions and evaluate them.
2. While (Total number of new solutions$\leq$50)
   (a) Train Kriging approximator on database.
   (b) Run the DE algorithm with the prescreening function.
   (c) Add 1-3 new solution(s) to database (IPI adds 3, others add 1).
3. Report the best found solution.

[1] Details are omitted due to space limitations.

The initial database of 20 random solutions and the following 50 samples represents a typical setup in the industry. For example, a computational fluid dynamics (CFD) simulation of a full pump curve can take up to 2-3 hours in a steady-state setup and up to 4-5 days for a full transient simulation. The number of initial solutions was deliberately kept at 20 individuals to stress the algorithms as the problem dimensionality increased.

4 Results and discussion

The results of the experiments are listed in table 2. In the table, a number marked in **bold** denote the algorithm with the best mean. Furthermore, a dagger (†) indicates that the algorithm is best wrt. Mann-Whitney rank sum test, i.e., the null-hypothesis[2] H_0 is rejected at 5% confidence level and a double dagger (‡) at 1% confidence level.

Table 2. Mean and standard deviation for the seven benchmark problems and the two pump problems.

Function	**OMV**	**PoI**	**ExI**	**GEI**	**IPI**
Branin 2D	2.57±1.795	2.95±2.371	0.41±0.017	0.42±0.019	**0.40±0.003**‡
Sasena 2D	1.29±1.780	1.51±2.297	-1.23±0.965	-1.41±0.117	**-1.45±0.022**
Six hump 2D	0.73±3.387	0.00±0.907	-0.90±0.110	-0.75±0.209	**-0.92±0.116**
Rastrigin2D	10.36±5.256	11.23±4.413	**2.69±2.440**	7.14±5.173	3.15±2.470
Hartmann 3D	-2.96±0.486	-3.26±0.482	-3.77±0.106	-3.71±0.170	**-3.82±0.056**†
Colville 4D	8519±17993	4480±6256	749±1153	**472±620**	829±1108
Hartmann 6D	-1.39±0.500	-1.23±0.576	-1.72±0.660	-1.99±0.707	**-2.77±0.472**†
Pump 6D	45.97±3.137	46.54±2.407	49.39±1.484	48.85±1.167	**49.53±1.205**
Pump 12D	59.57±1.806	59.57±1.477	61.28±1.133	60.52±1.243	**61.34±1.690**

As seen, OMV and PoI clearly have the worst performance on all problems. Comparing ExI, GEI, and IPI, the IPI achieves a better mean on seven of the nine tested problems and three of these seven are further supported by the Mann-Whitney rank sum tests. Interestingly, the GEI does not seem to be significantly better than traditional ExI. One possible explanation is that a g value higher than 1 actually induces an even stronger focus on finding solutions with high standard deviation. Stepping from traditional ExI ($g = 1$) to PoI ($g = 0$) occurs rather late in the annealing process and this step represents a rather large change in search strategy, i.e., from high-risk to low-risk as discussed earlier. Thus, the annealing approach could probably have benefited from smaller steps in the g value from, e.g., 1.0 to 0.9, and gradually toward 0.0. However, this is not possible with the current formulation of GEI.

Scrutinizing the IPI data, a typical run benefits from the portfolio optimization as follows (recall that $K_{NC} = 3$ and the algorithm adds 50 new solutions).

[2] The null-hypothesis states that the samples are drawn from the same distribution.

In the beginning of the run, the medium or high-risk solutions often locate a new best point, which are further improved by the low risk solution in following rounds. Towards the end, the high-risk solutions often explore regions with suboptimal performance as these parts have not yet been explored. In a few runs, the high-risk solution discovers a new best point towards the end of the run. Thus, the algorithm clearly benefits from implementing the portfolio strategy.

5 Conclusions

This paper presents the investment portfolio improvement prescreening approach and demonstrates its performance on seven benchmark problems and two real-world pump design problems. The experiments show that the suggested technique yields a better mean value in seven of the nine tested functions. Thus, the experiments support the initial analysis of the search strategies employed by established prescreening functions like Probability of Improvement (low risk) and Expected Improvement (high risk). Hence, it is an advantage to use a risk-spreading strategy as this simultaneously allows search in the vicinity of the best known solution and explorative search in new regions of the search space.

Regarding future work, this initial paper only presents the main idea and demonstrates its potential. The scheme is easy to extend and some next steps could be to investigate an annealing scheme where the target standard deviation t is gradually lowered or alternatively a self-adaptive version. Another idea is to implement an island version of the method.

References

1. Dennis, J., Torczon, V.: Managing approximate models in optimization. In: Multidisciplinary design optimization: State-of-the-art. SIAM (1997) 330–347
2. Ulmer, H., Streichert, F., Zell, A.: Evolution strategies assisted by gaussian processes with improved pre-selection criterion. In: Proceedings of the Fifth Congress on Evolutionary Computation (CEC-2003). 692–699
3. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**(4) (1998) 455–492
4. Sheng, N., Liao, C., Lin, W., Chang, L., Zhang, Q., Zhou, H.: A hybrid optimized algorithm based on ego and taguchi's method for solving expensive evaluation problems of antenna design. Progress In Electromagnetics Research C **17** (2010) 181–192
5. Jeong, S., Minemura, Y., Obayashi, S.: Optimization of combustion chamber for diesel engine using kriging model. Journal of Fluid Science and Technology **1**(2) (2006) 138–146
6. Couckuyt, I., Declercq, F., Dhaene, T., Rogier, H., Knockaert, L.: Surrogate-based infill optimization applied to electromagnetic problems. International Journal of RF and Microwave Computer-Aided Engineering **20**(5) (2010) 492–501
7. Emmerich, M.T., Giannakoglou, K.C., Naujoks, B.: Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. IEEE Transactions on Evolutionary Computation **10**(4) (2006) 421–439

8. Das, S., Mukherjee, R., Kundu, R., Vasilakos, T.: Multi-user detection in multi-carrier cdma wireless broadband system using a binary adaptive differential evolution algorithm. In: Proceeding of the fifteenth conference on Genetic and evolutionary computation conference (GECCO-2013). 1245–1252
9. Ursem, R.K., Vadstrup, P.: Parameter identification of induction motors using differential evolution. In: Proceedings of the Fifth Congress on Evolutionary Computation (CEC-2003). 790–796
10. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkley (1995)
11. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimisation over continuous spaces. Journal of Global Optimization **11** (1997) 341–359
12. Martin, J.D., Simpson, T.W.: Use of kriging models to approximate deterministic computer models. AIAA Journal **43**(4) (2005) 853–863
13. Laurenceau, J., Sagaut, P.: Building efficient response surfaces of aerodynamic functions with kriging and cokriging. AIAA Journal **46**(2) (2008) 498–507
14. Matheron, G.: Principles of geostatistics. Economic Geology **4**(4) (1963) 409–435
15. Lophaven, S.N., Nielsen, H.B., Søndergaard, J.: Dace – a matlab kriging toolbox. Technical report, IMM, Technical University of Denmark (2002)
16. Schonlau, M., Welch, W.J., Jones, D.R.: Global versus local search in constrained optimization of computer models. In: New Developments and Applications in Experimental Design. IMS Lecture Notes - Monograph Series (34). (1998) 11–25
17. Sasena, M.J., Papalambros, P., Goovaerts, P.: Exploration of metamodeling sampling criteria for constrained global optimization. Engineering Optimization **34** (2002) 263–278
18. Janusevskis, J., Riche, R., Ginsbourger, D., Girdziusas, R.: Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In: Learning and Intelligent Optimization. Lecture Notes in Computer Science, Springer (2012) 413–418
19. Ponweiser, W., Wagner, T., Vincze, M.: Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In: Proceedings of the Tenth Congress on Evolutionary Computation (CEC-2008).
20. Bäck, T., Fogel, D.B., Michalewicz, Z., et al., eds.: Handbook on Evolutionary Computation. IOP Publishing Ltd and Oxford University Press (1997)
21. Ursem, R.K.: Multinational evolutionary algorithms. In Proceedings of the Congress of Evolutionary Computation (CEC-99). 1633–1640
22. Ursem, R.K., Justesen, P.D.: Multi-objective distinct candidates optimization: Locating a few highly different solutions in a circuit component sizing problem. Applied Soft Computing **12**(1) (2012) 255–265
23. Sasena, M.J.: Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations. PhD thesis, University of Michigan (2002)
24. Gülich, J.: Centrifugal Pumps. Springer (2010)