UNIVERSITY OF SOUTHAMPTON

Faculty of Physical Sciences & Engineering

Electronics and Computer Science

ROBUST MULTIOBJECTIVE OPTIMISATION IN ELECTROMAGNETIC DESIGN

by

Yinjiang Li

Thesis for the degree of Doctor of Philosophy

August 2017

`

UNIVERSITY OF SOUTHAMPTON

Abstract

Faculty of Physical Sciences & Engineering

Electronics and Computer Science

Doctor of Philosophy

ROBUST MULTIOBJECTIVE OPTIMISATION IN ELECTROMAGNETIC DESIGN

by Yinjiang Li

In electromagnetic design, optimisation often involves evaluating the finite element method (FEM) – repetitive evaluation of the objective function may require hours or days of computation, making the use of standard direct search methods (e.g. genetic algorithm and particle swarm) impractical. Surrogate modelling techniques are helpful tools in these scenarios. Indeed, their applications can be found in many aspects of engineering design in which a computationally expensive model is involved.

Kriging, one of the most widely used surrogate modelling techniques, has become an increasingly active research subject in recent decades. This thesis focuses on four interesting research topics in surrogate-based optimisation: infill sampling efficiency, robust optimisation, and the memory problem encountered in large datasets and multi-objective optimisation. This thesis briefly provides relevant background information and introduces a number of independent novel approaches for each topic, with the aim of increasing efficiency of optimisation process and ability to handle larger datasets.

`

`

Contents

`

`

`

Declaration of Authorship

I, Yinjiang Li, declare that the thesis entitled “Robust Multiobjective Optimisation in Electromagnetic Design” and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:

[1] Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2016. A Dual Kriging Approach With Improved Points Selection Algorithm for Memory Efficient Surrogate Optimization in Electromagnetics. *IEEE Transactions on Magnetics*, *52*(3), pp.1-4.

[2] Li, Y., Rotaru, M. and Sykulski, J.K., 2016. Kriging based robust optimisation algorithm for minimax problems in electromagnetics. *Archives of Electrical Engineering*, *65*(4), pp.843-854.

[3] Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2017. A Kriging-Based Optimization Approach for Large Data Sets Exploiting Points Aggregation Techniques. *IEEE Transactions on Magnetics*, *53*(6), pp.1-4.

[4] Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2017. Localized probability of improvement for kriging based multi-objective optimization. *18th International Symposium on Electromagnetic Fields in Mechatronics, September 2017, Lodz, Poland*.

Signed:_____

Date:_____

`

`

Acknowledgements

Foremost, I would like to take this opportunity to express my highest esteem for Professor Jan Sykulski, my supervisor. I still vividly remember his first lecture about high voltage transformers back in the first year of my undergraduate education. Since then, his exquisite attitude toward work and life has been a strong and positive influence in my life during the research program and will continue to be an example in my future career. I could not feel more fortunate to have been one of his students; it was his professional guidance, exceptional patience and invaluable encouragement that made this possible.

I would also like to thank Dr. Mihai Rotaru and Dr. Roger, my second supervisor and a colleague of mine, respectively, for providing some of the FEM models for algorithms testing, their invaluable suggestions, discussions about and contributions to this publication.

I also need to thank Professor Alun Vaughan and Dr. Igor Golosnoy for their constructive advice and precious insight during my first-year examination and second year transfer. The thoughtful discussions we had certainly had a positive impact on the progress of this program.

`

`

Abbreviations

CAD	Computer aided design
EAs	Evolutionary algorithms
ED	Expected deterioration
EHVI	Expected hypervolume improvement
EI	Expected Improvement
FEM	Finite element model
HCA	Hierarchical cluster analysis
HV	Hypervolume
HVI	Hypervolume improvement
LPoI	Localised-probability of improvement
MOO	Multi-objective optimisation
MOOPs	Multi-objective optimisation problems
MSE	Mean square error
PoI	Probability of improvement
RMSD	Root-mean-square deviation
T.E.A.M.	Testing Electromagnetic Analysis Methods
WCEI	Worst-case expected improvement

`

`

Chapter 1. Introduction

1.1 Optimisation design

The engineering industry has witnessed an explosive advance in computer technology in recent decades. Engineers are currently able to solve complicated optimisation design problems with the assistance of powerful computers. Research in the field of optimisation, from linear programming (1939) [1] to evolutionary algorithms ([2] which have attracted significant attention lately), has exhibited successful application in engineering design problems [1]. Various optimisation methods have been published, with their efficiency and capability to solve different types of problems being consistently improved over time. The advantage of the current heuristic methods is that the algorithm is computationally fast; global optimisation methods like the genetic algorithm and the particle swarm algorithm can be naturally incorporated with parallel computing and can take advantage of multi-threaded computers.

1.2 Global optimisation

The purpose of global optimisation is to find the solution $\hat{x}$, in a feasible region that optimises the objective function $f(x)$. Depending on the specific application, the design can be one of a maximisation problem (to maximise the durability of a product) or a minimisation problem (to minimise the energy consumption of a device). Without loss of generality, we assume that they are all minimisation problems:

$$\text{maximise}\,\{f(x)\} = \text{minimise}\,\{-f(x)\} \tag{1.1}$$

Problem definition:

$$\begin{aligned} &\text{Minimise}\,\{f(\boldsymbol{x})\} \\ &\text{Subject to } x \in \boldsymbol{D} \\ &g_i(\boldsymbol{x}) \leq 0, \\ &h_i(\boldsymbol{x}) = 0, \qquad i = 1, 2 \ldots, n. \end{aligned} \tag{1.2}$$

where $f(x)$ is the objective function, $\boldsymbol{D}$ is a non-empty set of feasible design points, $g_i(x)$ is the inequality constraints and $h_i(\boldsymbol{x})$ is the equality constraints.

The local minimum in a convex problem is also the global minimum; this type of problem can be solved using local optimisation techniques efficiently. A number of well-known local optimisation methods include: the Simplex Method, Newton's methods, Quasi-Newton Methods and sequential quadratic programming.

`

Practical engineering design problems often possess a non-convex objective function or constraints. These problems are likely to contain multiple local minima and are often difficult to solve [3]. When multiple local minima are present in the search space (also called the feasible region) [3], new methodologies are necessary for locating the global minimum without being entrapped in a local minimum. The simplest approach is to start the local optimiser at multiple locations within the search space. Other examples of more sophisticated global optimisation techniques include: branch and bound, pattern search, simulated annealing, genetic algorithm and particle swarm. An in-depth review of recent developments in global optimisation methods can be found in [3], [4] and [5].

1.3 Previous work on kriging at ECS

This project will build on the previous work of Song Xiao (Roger), who recently completed his PhD. His results were published in several journal papers [6], [7], [8], [11], [12] and conference contributions [9], [10]. The main outcomes of his work are summarised as follows:

- The importance of balancing exploration and exploitation to effectively achieve convergence on the global optimum was confirmed after a series of tests.
- Techniques from reinforcement learning were employed to automatically introduce tuning weights to balance exploration and exploitation in response to the feedback produced by a kriging surrogate model. A novel method named "adaptive weighted expected improvement with rewards" was demonstrated to be able to learn from the experience of attempting the exploration and exploitation separately and then, determine the distribution of weights accordingly.
- A pre-test utilising only a combination of predicted results and the mean square error, which is computationally cheap, was developed and proven to be helpful for long-term decisions. Another novel method, called the "surrogate model based weighted expected improvement approach with rewards", which applies reinforcement learning based on the improved pre-test strategy, was proposed; it attempts to capture the optimal weights combination at each iteration of the optimisation process.
- To mitigate the issues caused by the accumulation of data by the correlation matrices due to an increase in the updated sampling process, an adaptive partitioning scheme for these matrices was introduced to the kriging surrogate model, especially in high-dimensional tasks.
- Several methods were investigated with regard to the robustness of the design. First, the gradient index method was evaluated. However, due to its limitations,

`

a modified method was developed to evaluate the sensitivity of each solution obtained by the kriging surrogate model. The worst case optimisation method was also explored and an assessment of the average performance was added to the algorithm to increase reliability.

The aim of this thesis is to enhance the methods and techniques already developed and thus, further advance the kriging technique, particularly within the context of multivariable optimisation and robust design. The focus is on handling the large amounts of data created by the relevant models.

1.4 Objectives

The use of kriging for design optimisation proved very promising and this provided the main motivation behind this work. The aim of this thesis is to advance the relevant techniques further, with focus on multi-objective and robust optimisation, while addressing some of the previously identified issues related to the algorithm efficiency.

The specific objectives may be formulated as follows:

1. Review and critically assess the state of the art in surrogate-based optimisation, with emphasis on kriging assisted techniques.
2. To consider and suggest new and better 'infill sampling criteria' based on the notion of expected improvement.
3. To enhance the application of kriging assisted methods to robust optimisation and suggest the most efficient approach.
4. To investigate the limitations of the kriging approach related to the size of correlation matrices and suggest possible solutions.
5. To extend the application of the developed techniques to multi-objective cases.
6. To illustrate the performance of the proposed algorithms using carefully selected test functions and practical applications, especially the T.E.A.M. problems*.

Note: T.E.A.M. (Testing Electromagnetic Analysis Methods) workshop problems [40] consist of a set of practical electromagnetic optimisation design problems for benchmarking the performance of optimisation algorithms. A list of benchmark problems can be found on the International Compumag Society website (http://www.compumag.org/jsite/).*

Topics not addressed in this thesis:

1. The issue of discontinuity in the objective function respond surface. Ordinary kriging is an exact interpolation method that generally does not work well on

`

problems with discontinued objective functions [13]; there are other methods that have been proven more effective in such type of problems, see details and comparisons in [15]. However, if the discontinuity was due to noisy data, stochastic kriging (one of the kriging variations) may be implemented [14].

2. Discontinuity in design parameters is not consider separately in the thesis, since all parameters are normalised for the kriging model – the only difference for problems with discontinued inputs is that the prediction and new infill points need to be sampled at feasible locations due to the discontinuity.

3. Design problems in this thesis are assumed to be global optimisation problems, i.e. obtaining the global optimum is the objective of the design. In some practical design cases, the manufacturer only considers local improvement over the current design due to practical limitations. This type of problem may be transferred into a global design problem by providing the desired search space for design parameters, and equality and/or inequality constraints if necessary.

1.5 Thesis structure

This thesis consists of 7 chapters. **Chapter 1** briefly introduces the field of optimisation design and provides an overview of the thesis. **Chapter 2** provides a background review for surrogate-based optimisation algorithms. Since kriging is one of the most popular surrogate modelling approaches, it is studied in detail.

Chapter 3 focuses on infill sampling criteria, particularly popular infill criterion expected improvement (EI). Furthermore, an efficient infill sampling approach utilising the automatic weight adjustment is proposed in combination with a fast gradient-based infill criterion search method.

Robust optimisation is studied in **Chapter 4**, which briefly introduces robust optimisation and problem classifications. A kriging-based approach for worst-case design problems is proposed for greater sampling efficiency.

Chapter 5 addresses potential bottle-necks in the current kriging method when handling large datasets. Three methods from the literature are briefly discussed in this chapter, two of which (dual-kriging and aggregation kriging) are used for improving memory efficiency.

In **Chapter 6**, multi-objective optimisation is studied in detail and existing approaches are described. Furthermore, a kriging-based localised probability of improvement approach is proposed.

Finally the research results are summarised in **Chapter 7**. Chapter 7 also discusses potential future work.

`

Chapter 2. Background review

2.1 Introduction

Rapid advances in electronic information technologies in recent decades have made large datasets more common in the field of computer modelling. Consequently, the demand for high quality computer models and solvers has made optimisation design problems both more complicated and more time-consuming to solve. Thus, surrogate model-based optimisation approaches are now drawing more attention since they provide more efficient solutions to expensive computer model designs. The efficiency of a modelling approach and its ability to handle complicated problems have been an important research objective in the optimisation community.

2.2 Surrogate-based optimisation

Computer aided design (CAD), modern engineering design processes and physical engineering experiments are restricted by cost, time or available resources. Thus, extensive sampling is not possible and complicated finite element models (FEM) may consume hours or even days per evaluation; in these circumstances, modelling techniques can improve design efficiency and quality.

Surrogate models need to be of good quality to be useful and fast when evaluating the underlying design problem. Generally, the evaluation time of the surrogate model is negligible compared to the sampling cost of the original problem. Furthermore, the model's quality is closely related to the amount of sampling data available. Therefore, when building a surrogate model, a trade-off between the quality of the model and the time required to build the model (including evaluation cost of the underlying problem) needs to be balanced.

The procedure of building a surrogate model involves the initial sampling, choosing the right model and fitting it to the initial available data, adding new sampling points to the initial model and then upgrading the previous model (the choice of added points are determined by infill criteria). This process is repeated until the termination criterion is achieved (i.e. the model is accurate enough, the maximum running time is reached, the approximate optimal point is found, etc.).

`

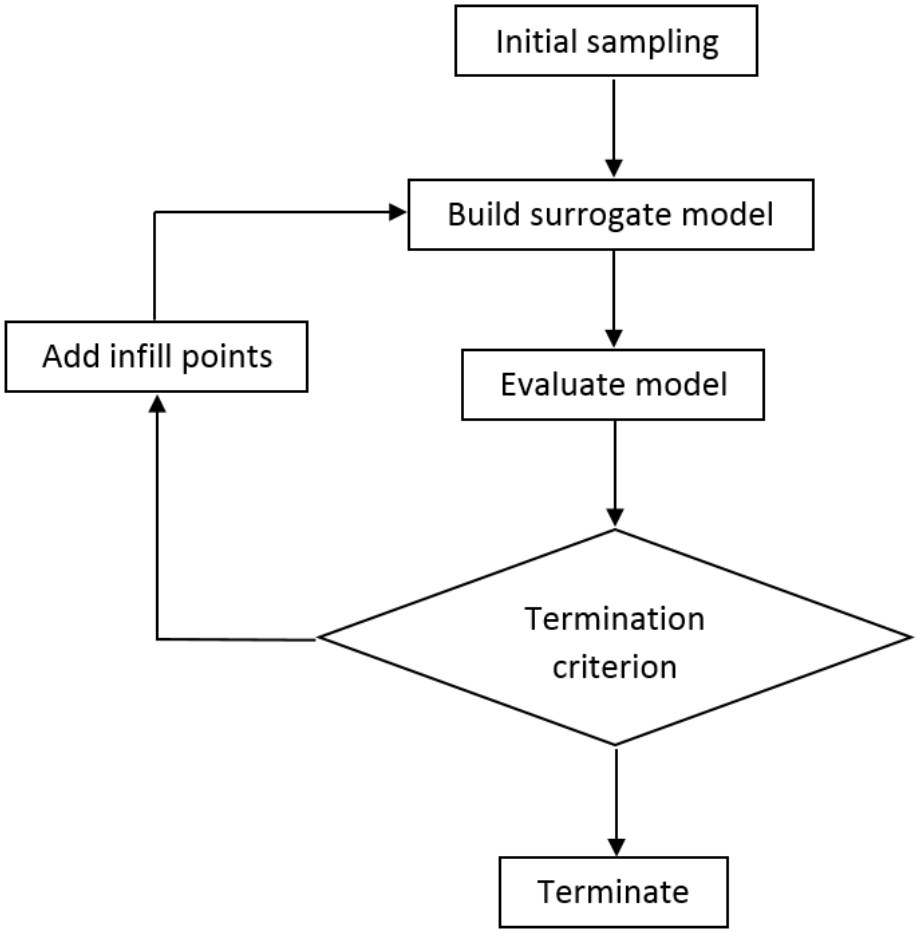


Figure 2.1. Surrogate-based optimisation workflow.

As discussed earlier in this chapter, a number of engineering applications are too expensive to apply optimisation routines directly to the problem. In these optimisation designs, a surrogate is built based on the sampled points obtained from the underlying problem. The model is then updated by adding additional infill sampling points depending on the infill strategy. The optimisation routine is subsequently applied to the surrogate model, instead of the underlying problem, at each iteration or at the end of the model updating process. Because the computation time of the surrogate model is much faster, the model can be searched thoroughly using population-based optimisation approaches, which require a large number of evaluations. At this stage, the global optimum point is only the “approximate” optimum point. The algorithm can validate it by evaluating the underlying problem and the newly sampled point can be added to the surrogate model during the updating process. This process repeats until the stop criterion is met.

`

2.3 Kriging theory

Kriging is an interpolation method originally developed by Danie Gerhardus Krige (1919-2013) to predict the distribution of gold, based on samples obtained from a small set of boreholes. Kriging was originally used in geostatistics to determine the mineral distribution in a field, due to its ability to plot the response surface of a particular objective function. It has also been widely implemented in the field of optimisation.

The kriging model is a linear combination of a global model and a local departure:

$$Y(x) = f(x) + Z(x) \tag{2.1}$$

where $Y(x)$ is the unknown function to be approximated;

$f(x)$ is the polynomial that interpolates x: "the function f(x) is similar to a polynomial type response surface, providing a global model of the design space."

$Z(x)$ is a realisation of the stochastic process. It follows the Gaussian distribution with a mean of 0, variance of σ^2 and non-zero covariance. The covariance is given by:

$$\boldsymbol{V}\left(x_i, x_j \right) = \sigma^2 \boldsymbol{R}\left(\boldsymbol{\theta}, x_i, x_j\right) \tag{2.2}$$

where σ^2 is the process variance, $\boldsymbol{R}$ is the spatial correlation model (function), $\boldsymbol{\theta}$ is the correlation function parameter and x_i and x_j are two different design points.

The likelihood formula is:

$$\frac{1}{(2\pi)^{\frac{\pi}{2}}(\sigma^2)^{\frac{\pi}{2}}|R|^{\frac{1}{2}}} exp\left[-\frac{y - 1u'R^{-1}(y - 1u)}{2\sigma^2}\right] \tag{2.3}$$

The stationary points of u and σ^2 can be found by calculating its partial derivatives with respect to u and σ^2, respectively, and making them equivalent to 0.

The optimal values of the likelihood function in terms of R are:

$$\hat{\mu} = \frac{1'R^{-1}y}{1'R^{-1}1} \tag{2.4}$$

$$\hat{\sigma}^2 = \frac{(1 - 1\hat{u})'R^{-1}(y - 1\hat{\mu})}{n} \tag{2.5}$$

Substituting the above two equations back into the likelihood function, the following concentrated log-likelihood function is obtained:

$$-\frac{n}{2}\log(\hat{\sigma}^2 - \frac{1}{2}\log(|R|) \tag{2.6}$$

`

The value of y that maximises the concentrated log-likelihood function is the kriging prediction at the corresponding design point x.

Therefore, a kriging prediction at point x within the design field is as follows:

$$\hat{y} = \hat{\mu} + rR^{-1}(y - 1\hat{\mu}) \tag{2.7}$$

One of the greatest advantages of kriging approximation in optimisation design is that it provides an error estimation of the predicted point - the mean square error (MSE):

$$s^2 = \hat{\sigma}^2 \left[1 - r'R^{-1}r + \frac{(1 - r'R^{-1}r)^2}{1'R^{-1}1}\right] \tag{2.8}$$

The correlation function [16], [17] is in the form of:

$$R(\theta, x_i, x_j) = \prod_{n=1}^{m} R_n(\theta, x_{i_n} - x_{j_n}) \tag{2.9}$$

The correlation function is a function of Euclidean distance between any two points. There are different correlation functions:

Exponential

$$R_n\left(\theta, x_{i_n}, x_{j_n}\right) = \exp(-\theta_n, \left|x_{i_n} - x_{j_n}\right|) \tag{2.10}$$

General Exponential

$$R_n\left(\theta_n, x_{i_n}, x_{j_n}\right) = \exp(-\theta_n, \left|x_{i_n} - x_{j_n}\right|)^{p_n}, \qquad 0 < p_n \leq 2 \tag{2.11}$$

Gaussian

$$R_n\left(\theta_n, x_{i_n}, x_{j_n}\right) = \exp(-\theta_n, \left|x_{i_n} - x_{j_n}\right|^2) \tag{2.12}$$

Linear

$$R_n\left(\theta_n, x_{i_n}, x_{j_n}\right) = \max\{0, 1 - \theta_n, \left|x_{i_n} - x_{j_n}\right|\} \tag{2.13}$$

Spherical

$$R_n\left(\theta_n, x_{i_n}, x_{j_n}\right) = 1 - 1.5\xi_n + 1.5\xi_n^3, \quad \xi_n = \min\left\{1 - \theta_n, \left|x_{i_n} - x_{j_n}\right|\right\} \tag{2.14}$$

Cubic

$$R_n\left(\theta_n, x_{i_n}, x_{j_n}\right) = 1 - 3\xi_n^2 + 2\xi_n^3, \quad \xi_n = \min\{1 - \theta_n, \left|x_{i_n} - x_{j_n}\right| \tag{2.15}$$

`

Spline

$$\begin{cases} 1 - 15\xi_n^2 + 30\xi_n^3, & for\ 0 \le \xi_n \le 0.2 \\ 1.25(1-\xi_n)^3, & for\ 0.2 < \xi_n < 1 \\ 0, & for\ \xi_n \ge 1 \end{cases}, \xi_n = \theta_n, \left| x_{i_n} - x_{j_n} \right| \qquad (2.16)$$

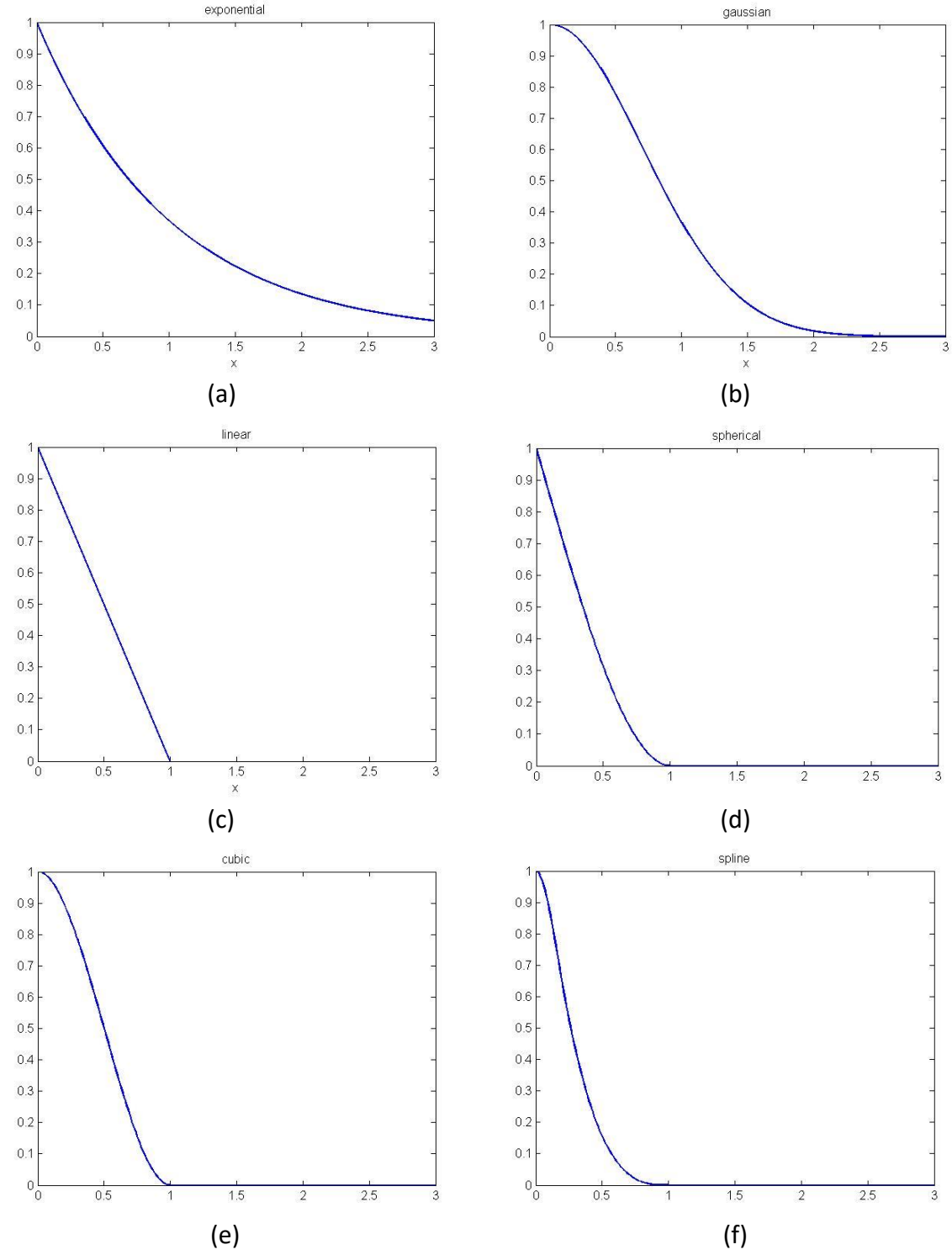


Figure 2.2. Correlation functions. (a) Exponential, (b) Gaussian, (c) Linear, (d) Spherical, (e) Cubic, (f) Spline.

`

The choice of the correlation function will have a direct impact on predicted values at all unknown locations, therefore choosing the right correlation function is not a trivial task. Kriging was originally invented to solve a geographical problem as mentioned earlier, for geospatial datasets; it is often assumed that the sampling locations nearby have similar response values, and their relationship is modelled by one of the correlation functions.

In the context of optimisation, design problems can be categorised into continuous optimisation problems and discrete optimisation problems in terms of continuity of the variables. When solving continuous optimisation problems, it is reasonable to assume that the respond surface is continuous, since a small variation in a parameter will produce a similar design and it is therefore likely to result in a small change in the output response. For an unknown problem, the general exponential function, and the Gaussian function, are reasonable choices of the correlation function, since they include information from all existing observations and provide smooth respond surfaces [18], [19]. The formula for general exponential correlation function is as follows:

$$R_n\left(\theta_n, x_{i_n}, x_{j_n}\right) = \exp(-\theta_n, \left|x_{i_n} - x_{j_n}\right|)^{p_n}, \qquad 0 < p_n \leq 2 \qquad (2.17)$$

where θ_n and p_n are the hyper parameters, which need to be optimised. θ_n controls the decreasing speed of the correlation between design sets as they move further apart and p_n controls the behaviour of the correlation function. $p_n = 1$ gives the exponential correlation function, which has a linear behaviour around the origin. It is suitable for approximating a response surface with low correlation. $p_n = 2$ gives the Gaussian correlation function. It has a parabolic behaviour around the origin and is suitable to approximate a smooth and differentiable response surface. Therefore, it has been widely accepted that the Gaussian correlation function is an appropriate choice for unknown continuous optimisation design problems ($p_n = 2$) [20], [21]. Figure 2.4 illustrates the kriging prediction (25 design points) of the following 2D-objective function (Figure 2.3) based on various correlation functions.

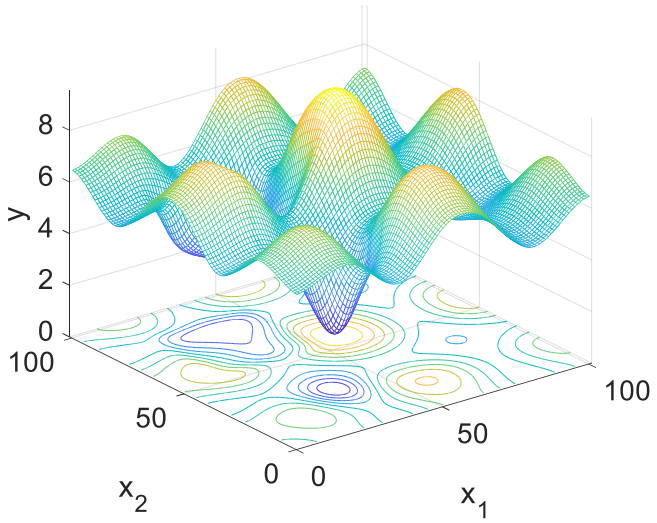


Figure 2.3. 2D-objective function.

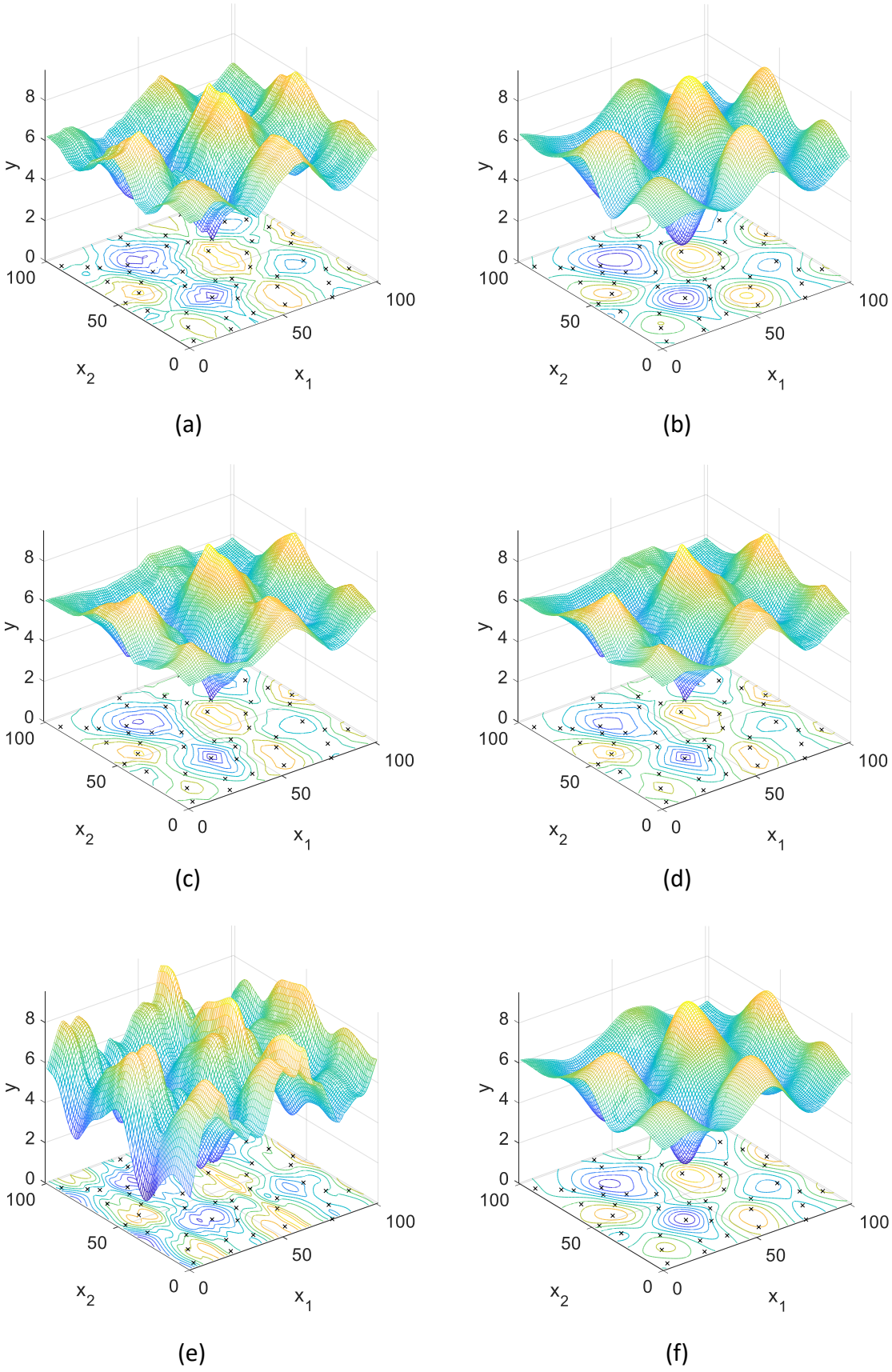


Figure 2.4. Kriging models based on various correlation functions. (a) Exponential, (b) Gaussian, (c) Linear, (d) Spherical, (e) Cubic, (f) Spline.

2.4 Sampling plan

In recent decades, more publications have focused on improving the sampling plan by comparing the development of surrogate model methodologies. Since surrogate-based optimisation approaches are often implemented for computationally time-consuming problems, improvement in sampling efficiency will directly affect the overall design efficiency. Depending on the design stage, the sampling plans can be divided into two categories: initial sampling and infill sampling.

2.4.1 Initial sampling

The goal of an initial sampling plan is to gain the maximum amount of information from the limited number of initial sampling points in the design. The design problems can be classified into: physical experiments (laboratory experiments) or computer experiments. Generally, physical experiments inevitably involve random error or noise and the experiment results are likely to be different even when the same sampling points are used each time. Computational experiments, on the other hand, are deterministic and repeated sampling from the same design site provides identical output results.

For the above reason, the emphasis of the sampling plan for physical experiments and computer experiments is not the same. Due to the existence of random errors in the physical experiments, a number of sampling points are usually taken from the boundary of the design space as this enables the user to capture the global trend more precisely in the presence of noise [22]. The identical sampling and evaluation process is often repeated to minimise the impact of random error. Sampling plans for physical experiments include: factorial design, central composite design and box-behnken [23].

Drawbacks of these sampling plans are that they do not include a number of important interior design features and that the design points are deterministic. Sampling plans for computer experiments tend to place sampling points evenly in the interior of the design space and the stochastic process is often observed within the sampling plan. The well-known methods are: orthogonal array algorithm, pseudo-Monte Carlo and Latin hypercube. Extensions and enhancements of the latter two sampling methods can be found within the literature [24]. These sampling schemes avoid providing misleading information in a harmonic response surface; the extensions of the Monte Carlo and Latin hypercube sampling schemes also enable the user to specify the number of samples when large numbers of sampling points are not possible in high dimensional problems.

The Monte Carlo scheme can be understood as a general random sampling approach and it is a commonly used tool in numerical integration, statistical sampling, queueing theory and global optimisation.

`

The quasi-Monte Carlo method was developed in the 1950s and has been largely improved by the accuracy demand in numerical integration, where instead of random samples, deterministic samples are used; this led to a guaranteed error bound in the integration and hence, an expected level of accuracy can be obtained. In addition to the aforementioned advantage, it always achieves higher accuracy and efficiency in the application of numerical integration. For a comprehensive study of the Monte Carlo and Quasi-Monte Carlo methods, readers are referred to [25].

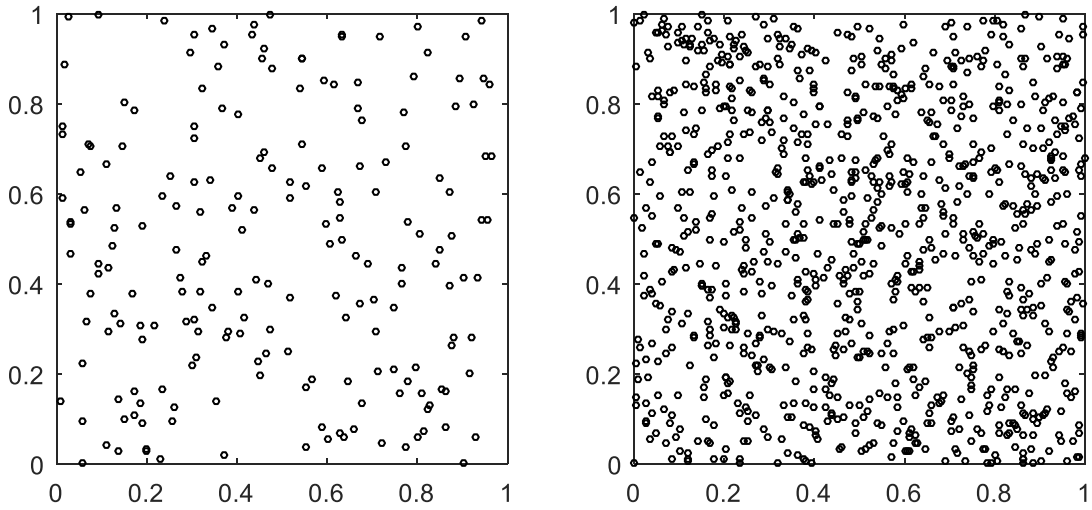


Figure 2.5. (a) Monte Carlo method 200 points, (b) Monte Carlo method 1000 points

As stated in the beginning of this section, the goal of initial sampling is to gain maximum information from the limited number of sampling points, hence minimising the discrepancy in the initial samples would be the optimal choice in most scenarios. In the following section, we briefly compare the quasi-Monte Carlo method and the Monte Carlo method. In MATLAB, the Halton sequence and the Sobol sequence are the two main methods for generating the quasi-random sequence. For random sequence generation and methodology, readers are referred to the MATLAB Statistics and Machine Learning Toolbox™ user manual [26].

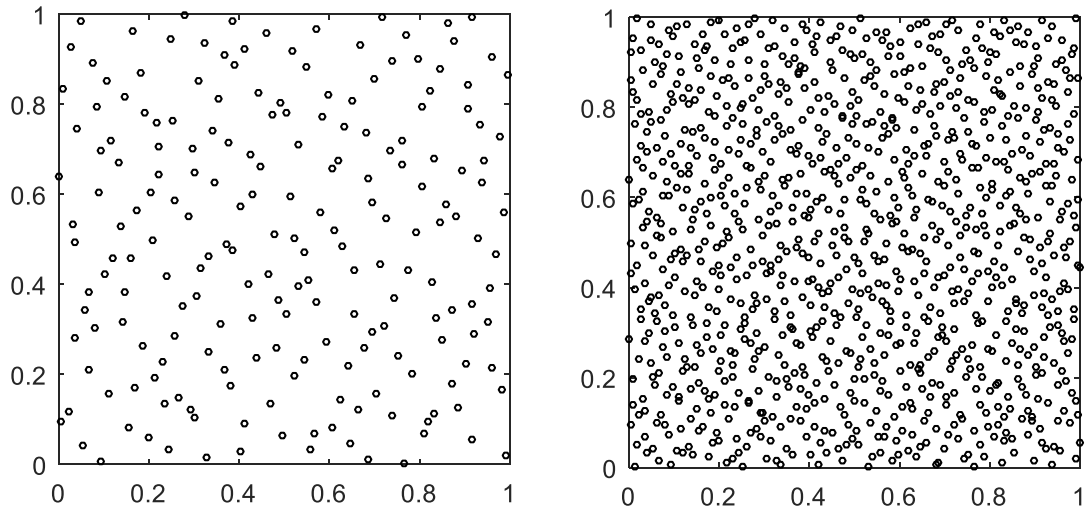


Figure 2.6. (a) Quasi-Monte Carlo method 200 points, (b) Quasi-Monte Carlo method 1000 points.

As can be seen from Figures 2.5 and 2.6, the quasi-Monte Carlo method provides more uniform sampling over the design space and hence effectiveness in the context of the initial design sampling. The Monte Carlo method is also used in a number of publications to compute the hypervolume introduced in Chapter 7 for multi-objective algorithms. The quasi-Monte Carlo method can also increase computation accuracy and efficiency of the hypervolume.

2.4.2 Infill sampling

After the initial sampling process, the initial model is built. Infill points are additional sampling points added during the model updating process. While the model construction processes are essentially the same in different applications, the infill criteria may vary depending on the field of applications. In the optimisation design, the infill rule needs to consider both exploration of the global space and the exploitation of the local area, while in a number of other fields of modelling (such as geometrical analysis) the general purpose of the infill criteria would be to explore the problem space on a global scale as much as possible.

Since Jones (1998) introduced the expected improvement (EI) approach [27], it has drawn much attention and quickly become one of the most popular infill criteria in kriging surrogate model-based optimisation.

At any predicted point, kriging provides both the prediction value $\hat{y}$ and the estimated mean square error $\hat{s}^2$ and denotes the minimum observation as y_{min}. The probability of improvement at the predicted point over the known minimum point is calculated by the Gaussian probability density function:

`

$$P(y \leq y_{min}) = \int_{-\infty}^{y_{min}} Y(x)\, dy = \Phi\left(\frac{y_{min} - \hat{y}}{\hat{s}}\right) \tag{2.18}$$

where $Y(x)$ is the Gaussian function at the predicted point, has a mean $\hat{y}$ and mean square error $\hat{s}^2$ and $\Phi(\cdot)$ is the cumulative distribution function.

The amount of improvement is then:

$$E(I) = \int_{-\infty}^{y_{min}} I \cdot \phi(F)\, dF \tag{2.19}$$

$$\phi(F) = exp\left[-\frac{\left(F - \hat{y}(x)\right)^2 / 2\hat{s}^2(x)}{\sqrt{2\pi} \cdot \hat{s}(x)}\right] \tag{2.20}$$

where $I = \max(y_{min} - y, 0)$, F is the Gaussian variable $N[\hat{y}(x), \hat{s}^2(x)]$ and $\phi(\cdot)$ is the probability density function.

Thus, EI is:

$$E[I(x)] = \begin{cases} \left(y_{min} - \hat{y}(x)\right)\Phi\left(\frac{y_{min} - \hat{y}(x)}{\hat{s}(x)}\right) + \hat{s}\phi\left(\frac{y_{min} - \hat{y}(x)}{\hat{s}(x)}\right), & s > 0 \\ 0, & s = 0 \end{cases} \tag{2.21}$$

where $\phi(\cdot)$ is the probability density function.

The term $\left(y_{min} - \hat{y}(x)\right)$ represents the amount of improvement expected at certain point and the term $\hat{s}$ represents the uncertainty at that point. Therefore, the EI infill criterion simultaneously addresses both exploitation and exploration.

A one-dimensional example of a kriging-based optimisation solver that uses EI infill criterion is illustrated in following figures. The dotted line presents the one-dimensional test function; the solid blue line represents the kriging model; the orange dots are the design points; and the solid black line at the bottom of the figure is the EI infill criterion. The next infill sampling point is taken at the location of maximum EI. Figures 2.7 to 2.11 illustrate the kriging model at the 5th, 10th, 15th and 20th iteration.

`

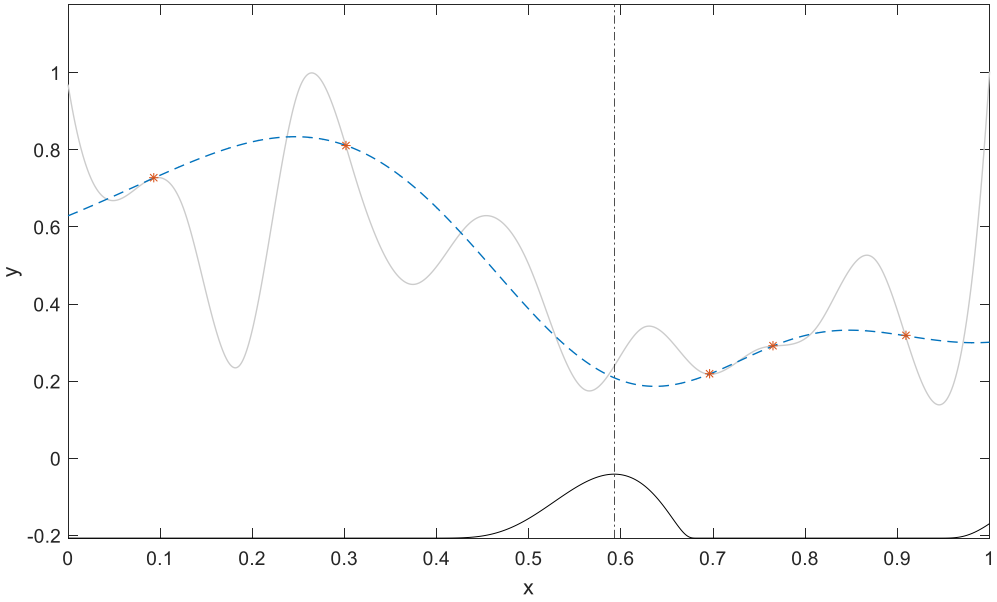


Figure 2.7. Kriging model with EI infill criterion at the 5th iteration.

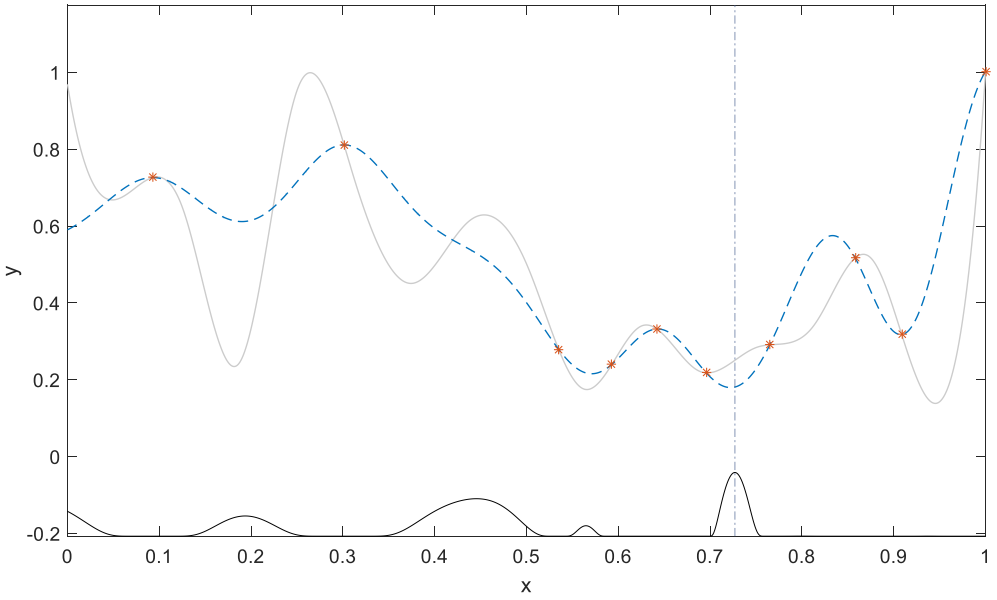


Figure 2.8. Kriging model with EI infill criterion at the 10th iteration.

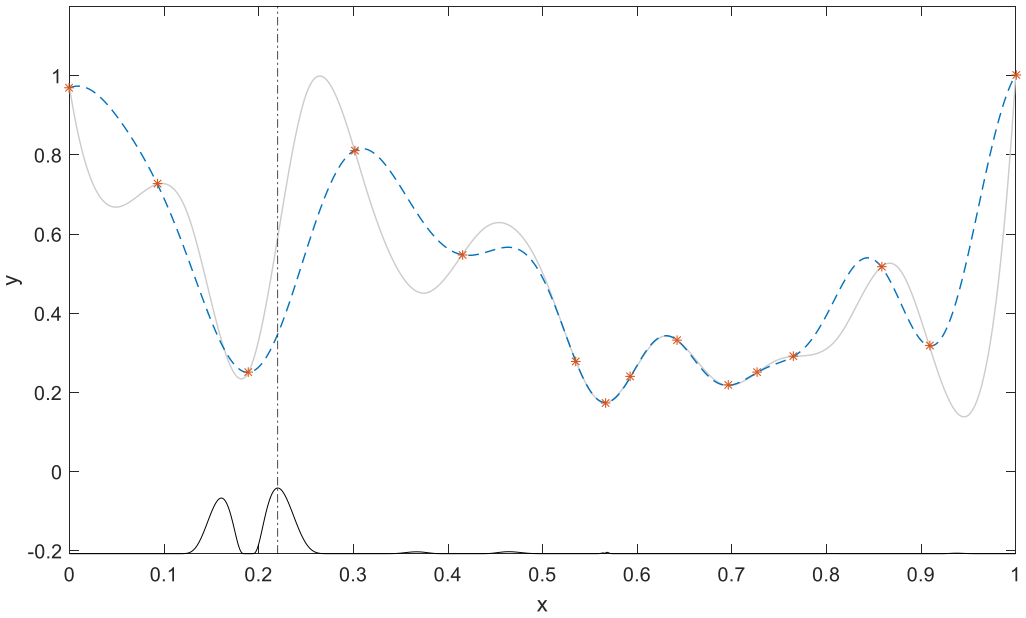


Figure 2.9. Kriging model with EI infill criterion at the 15th iteration.

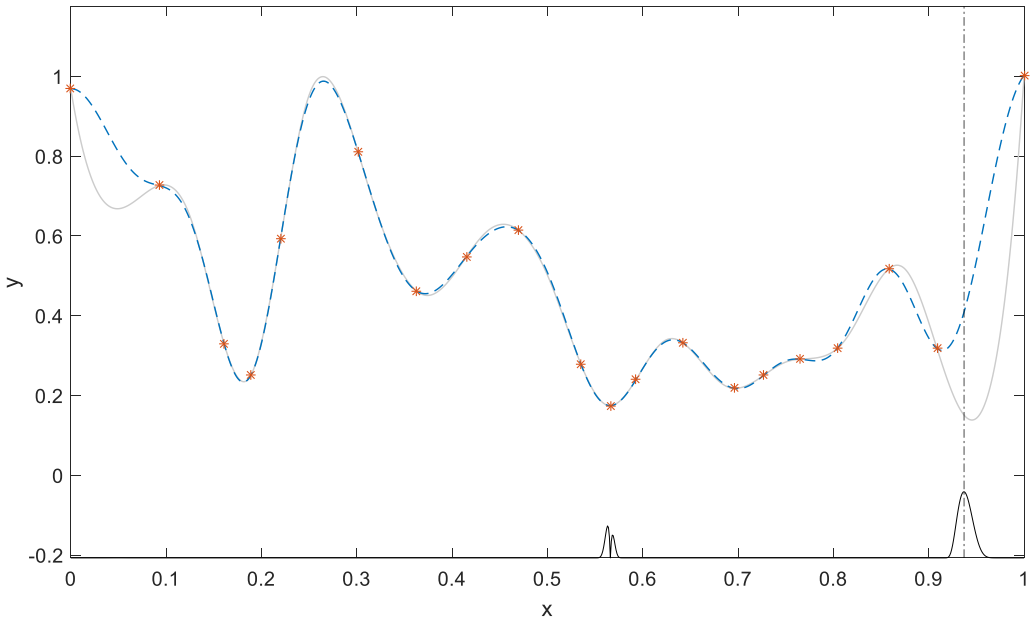


Figure 2.10. Kriging model with EI infill criterion at the 20th iteration.

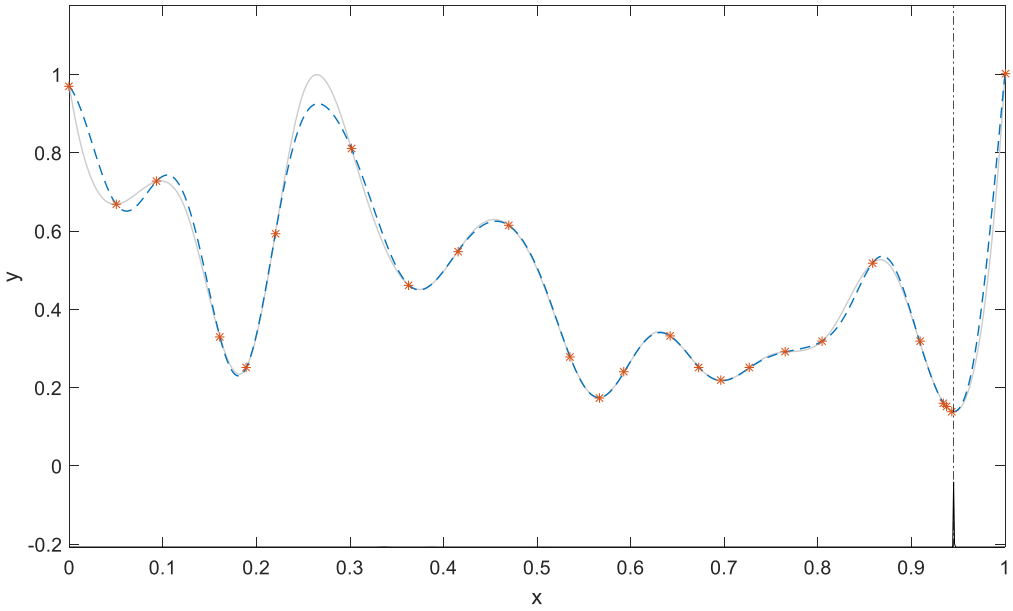


Figure 2.11. Kriging model with EI infill criterion at the 25th iteration.

A number of infill criteria that exist within the literature are more or less similar to the expected improvement and maximising the expected improvement is considered to be superior to other infill criteria in kriging modelling. However, details of these methods are not studied in this chapter, they can be found in [24], [28] and [23]. These methods include: locating the threshold-bounded extremes, locating the regional extremes, minimising surprises, maximum variance, minimising the lower confidence bounding (LCB), maximising the probability of improvement and maximising the MSE.

The following chapters address surrogate-based optimisation, including infill sampling efficiency, robust optimisation, handling large datasets and multi-objective optimisation. Existing results are presented in each chapter and new solutions are proposed. The proposed methods primarily focus on improving the efficiency of the optimisation process.

`

Chapter 3. Efficient sampling plan

3.1 Introduction

In situations in which the underlying problem overwhelms the computational cost incurred from building a surrogate model, infill sampling becomes the most important step that directly affects the efficiency of the overall optimisation process. For this reason, it has been a major research topic within the literature.

In the previous chapter, we have covered the most widely accepted infill sampling criteria in surrogate-based optimisation design, namely expected improvement (EI). In this chapter, we will discuss its major drawbacks and then introduce a modified version which aims at addressing these issues and improving the sampling efficiency.

3.2 EI infill criterion

One of the drawbacks in the standard EI sampling approach is the slow convergence speed when the design space contains multiple local minimum points. The original EI approach tends to over-exploit an existing attraction region until the local minimum is found, this is also discussed in [29].

A test function was built to illustrate how standard EI infill criterion work in one-dimensional optimisation problems. The objective function contains three attraction regions and one global minimum. The attraction region around the global optimum is narrower compared to attraction regions around the other two local optimal points, making it more difficult to find. Coordinates for the sampling points on the x-axis were rounded to 3 digits, hence there are 1,001 viable design sites within the search region. The objective is to find the global minimum at $x = 0.108$. The stopping criteria are as follows:

1. Duplicated design sites (distance between any two infill points is less than 10^{-3})*;
2. Maximum of 30 iterations;
3. The global optimum is found (in practice, the global optimum is often unknown for an optimisation problem, but we could assume it is known in this example without conflicting with the purpose of this test).

Note: 10^{-3} is a user specified parameter, it is the search interval of the optimisation solver. For a normalised search space, a 10^{-3} search interval means that there are $\frac{1}{10^{-3}} + 1$ (i.e. 1001) number of infill sampling criterion evaluations at each iteration, which include kriging prediction, calculation of MSE and calculation of EI; the location with the maximum EI value is selected as the next infill sampling location. A smaller search*

`

interval means better accuracy but more computing cost, a location change smaller than 10^{-3} means duplicated design sites.

The optimisation progress (at the 7th, 11th and 16th iteration) of the Kriging model updated by the standard EI infill criterion is depicted in the figures below. The objective function is represented by a brown dashed line, the Kriging model is noted as the blue line and EI is the orange line at the bottom; furthermore, black arrows and the number n indicate the infill sampling points taken at the n^{th} iteration.

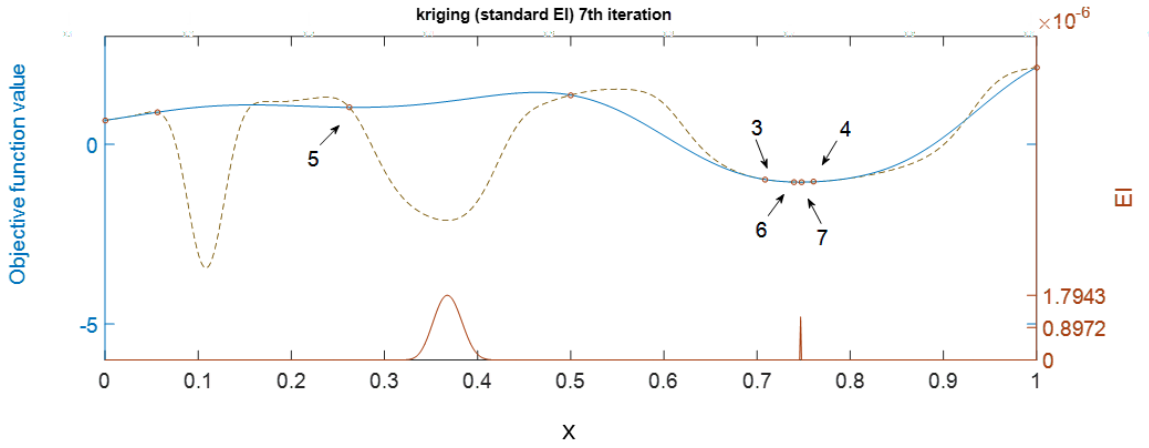


Figure 3.1. Kriging (standard EI) at the 7th iteration.

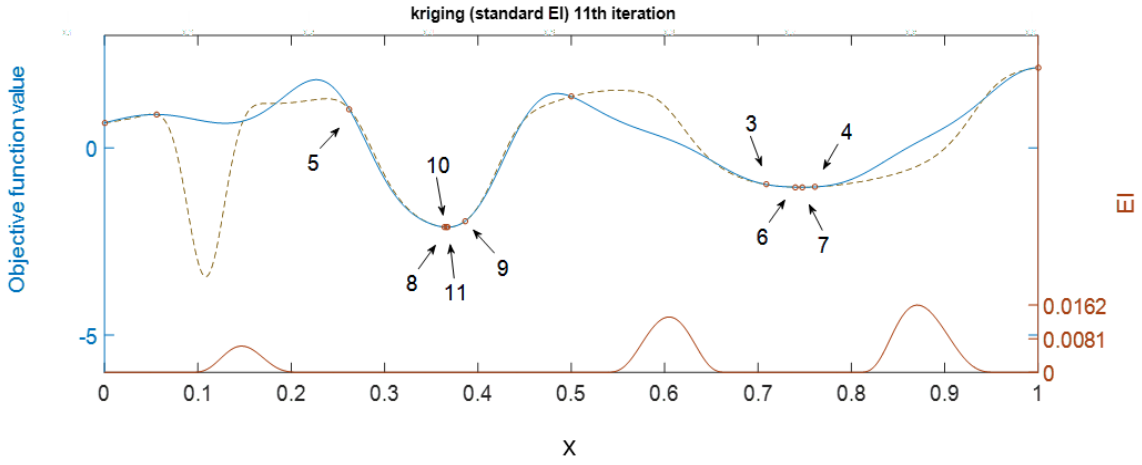


Figure 3.2. Kriging (standard EI) at the 11th iteration.

Figure 3.1 displays the standard EI approach at its seventh iteration. Four infill points taken at the 3rd, 4th, 6thand 7th iteration are within the region between 0.7 and 0.8 on the x-axis. After the seventh iteration (Figure 3.2), another four infill points were added to the region between 0.35 and 0.4 on the x-axis at the 8th, 9th, 10th and 11th iteration, respectively.

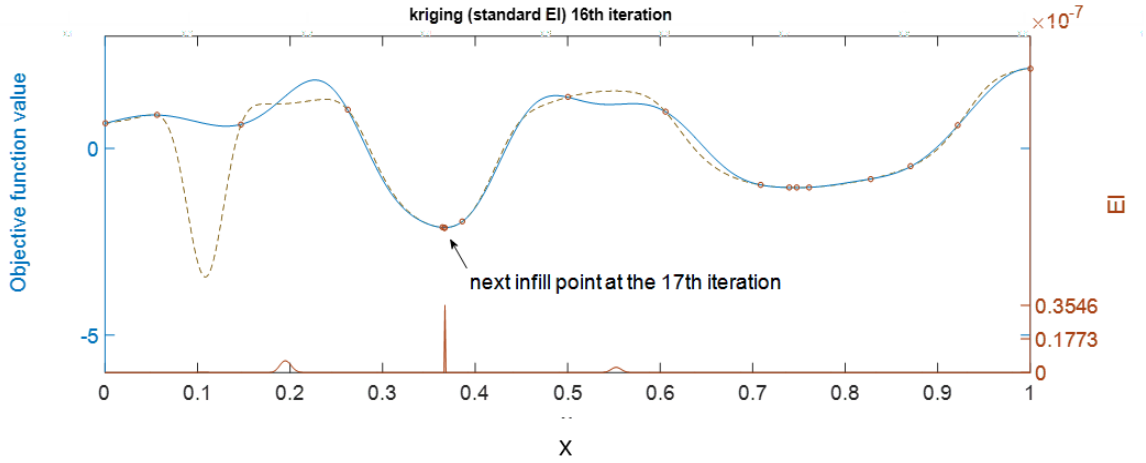


Figure 3.3. Kriging (standard EI) at the 16th iteration.

In Figure 3.3 above, the standard EI approach would suggest to continue to exploit the second attraction region and take the next sampling point at $x = 0.3664$ at the 17th iteration. After rounding x to nearest 3 digits, the next infill point is at $x = 0.366$. However, there is an existing design site (11th iteration) located at $x = 0.366$. The first stopping rule is reached and the program terminates at the 17th iteration.

In this test, the standard EI approach can effectively locate the local minima within an attraction region, but many evaluations were conducted to exploit the adjacent area of the existing points before exploring other regions. This characteristic of the standard EI approach could be counterproductive in the initial stages of the optimisation process. Indeed, emphasising exploration of the entire design space in the beginning of the process is generally preferred. It is better to spread the sampling points more evenly in the early stages of optimisation to avoid being trapped in a local minimum at the beginning and consequently, improve the overall sampling efficiency.

3.3 Efficient sampling scheme

3.3.1 Exploitation and exploration

We illustrated the shortcomings of the standard EI approach in the previous section; to overcome its characteristic of over-exploiting a known region in the early stages of the optimisation process, the mean square error (estimated error) between known sampling points is taken into consideration.

Consider the following sampling criterion:

$$\text{Sampling criterion } = \max\{EI\} \times \text{MSE} \times weight + \max\{\text{MSE}\} \times EI \qquad (3.1)$$

where $\max\{EI\}$ and $\max\{\text{MSE}\}$ are scaling terms applied to account for different values of components and thus, rescale the results. The estimated error at any unknown

point is larger when it is farther away from existing design points. Therefore, the MSE term will overwhelm the EI term at the initial stage of the optimisation process, since there are fewer known sampling points in the model, thus consequently emphasising the exploration of the entire design space.

The MSE term can be considered the exploration part of the infill sampling criterion and the EI term is considered the exploitation part. Although exploration of the design space is preferred in the early stages, exploitation of an attraction region cannot be overlooked, since finding the global optimal point is the ultimate purpose of optimisation design and applying the exploitation part well helps the program efficiently converge on the optimum.

Consider the following method to calculate the weight term v:

$$v = b^{k-m} \tag{3.2}$$

where $b \in]0, 1[$, b controls the decreasing rate of weight term v, k is the k^{th} iteration, and m controls the initial value of v; both parameters b and m are user-specified parameters. The user can specify the initial weight v using the following equation:

$$m = 1 - \frac{\log(b)}{\log(v)} \tag{3.3}$$

If the initial sampling points have covered the design space reasonably well, it is often sufficient to omit m and start with the weight v being equal to b.

Although the parameter b can be set to a value between 0 and 1, exclusive, the range [0.9 1[is more reasonable in practice. The weight v given in (3.2) decays exponentially; once the user-defined parameter b has been specified, the weights on the exploration term at all the iterations are predetermined. Therefore, b should be defined based on the maximum number of iterations.

For an unknown 1-D problem, where the kriging-based optimisation solver is likely to locate the optimum within 100 iterations, the value of b can be set to 0.95, i.e. after 20 iterations and 50 iterations the weight on the exploration part is around 36% and 7%, respectively, compared to 100% weight on the exploitation part. For a 2-D problem, with 500 maximum number of iterations, the parameter b can be set to 0.995, i.e. after 200 iterations and 400 iterations the weight on the exploration part is around 37% and 13%, respectively. Figure 3.4 presents the weight value applied to the exploration term at each iteration.

`

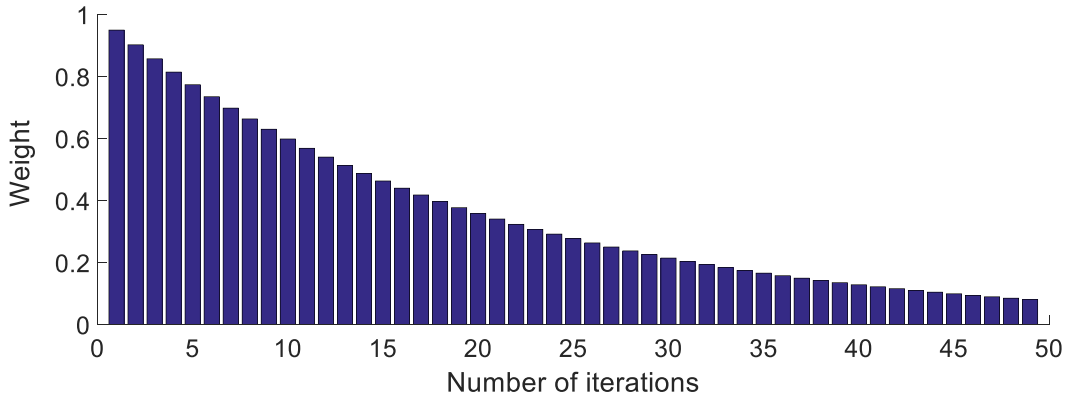


Figure 3.4. Weight at each iteration ($b = 0.95$).

As the optimisation progresses, more sampling points are added to the model; the estimated error gradually loses weight and the focus is shifted to exploitation. The idea of weight shares some similarity to the temperature used in simulated annealing [30]; the drawback of this approach is that the weight on the exploration and exploitation parts is predetermined for each iteration step, which means that the programmer needs to be knowledgeable about the complexity of the problem to efficiently obtain the result.

3.3.2 Dynamic balancing based on model quality

When the complexity of a design problem is unknown, using a fixed weighting parameter could potentially lead to unwanted results, such as over-exploitation in the early stages or over-exploration in later stages of the optimisation process; both scenarios affect the modelling efficiency, however in the worst case, the program would fail to find the optimal point.

Although this is a common problem in surrogate based optimisation, few solutions link this problem to the accuracy of the model itself. In this section, an efficient method is introduced to assess the surrogate model quality, which enables the program to automatically balance the weight on exploration and exploitation parts, i.e. creating a feedback loop from the optimisation result.

The predictor deviation d in iteration $iter$ is defined as:

$$d_{iter} = f(\boldsymbol{x}_{iter}) - Pred_{iter-1}(\boldsymbol{x}_{iter}) \tag{3.4}$$

Where $\boldsymbol{x}_{iter}$ is the location of the infill point in the $iter^{th}$ iteration, $f(\boldsymbol{x})$ is the evaluated objective function at location $\boldsymbol{x}$ and $Pred_{iter-1}(\boldsymbol{x})$ is the predicted objective function value at location $\boldsymbol{x}$ in iteration $iter - 1$.

The deviation d_{iter} is calculated and recorded whenever a new infill point is defined. Finally, the historical root-mean-square deviation (RMSD) is:

$$RMSD = \sqrt{\frac{\sum_{iter=1}^{m} {d_{iter}}^2}{m}} \tag{3.5}$$

where m is the most recent iteration.

To obtain a generalised weight term, an exponentially weighted RMSD was applied in this case to put more weight on recent results; the aim was to emphasise the recent prediction error to reflect the optimisation progress. The exponentially weighted RMSD is calculated using the formula:

$$RMSD_{weighted} = \sqrt{\frac{\sum_{iter=1}^{m}(1-\alpha)^{m-iter} \times {d_{iter}}^2}{\sum_{iter=1}^{m}(1-\alpha)^{m-iter}}} \tag{3.6}$$

where α is the decay parameter and $0 < \alpha < 1$. A larger α will put less weight on past prediction errors and vice versa. When $\alpha = 0$, $RMSD_{weighted}$ is identical to $RMSD$.

We obtained a generalised weight term that represents the current optimisation progress in terms of model quality by taking the ratio of the exponentially weighted standard deviation $RMSD_{weighted}$ and regular standard deviation $RMSD$ of historical prediction errors:

$$v = weighted\_RMSD / RMSD \tag{3.7}$$

The parameter v can be regarded as a measurement of model quality at any stage; the value of $weight$ usually lies within the range of $[0, 1+\alpha]$; and α controls the gradient of the exponential weight function – unless α is insignificant and the current prediction error is substantially larger than historical prediction errors. As the prediction errors decrease, α will gradually move toward a smaller value close to zero.

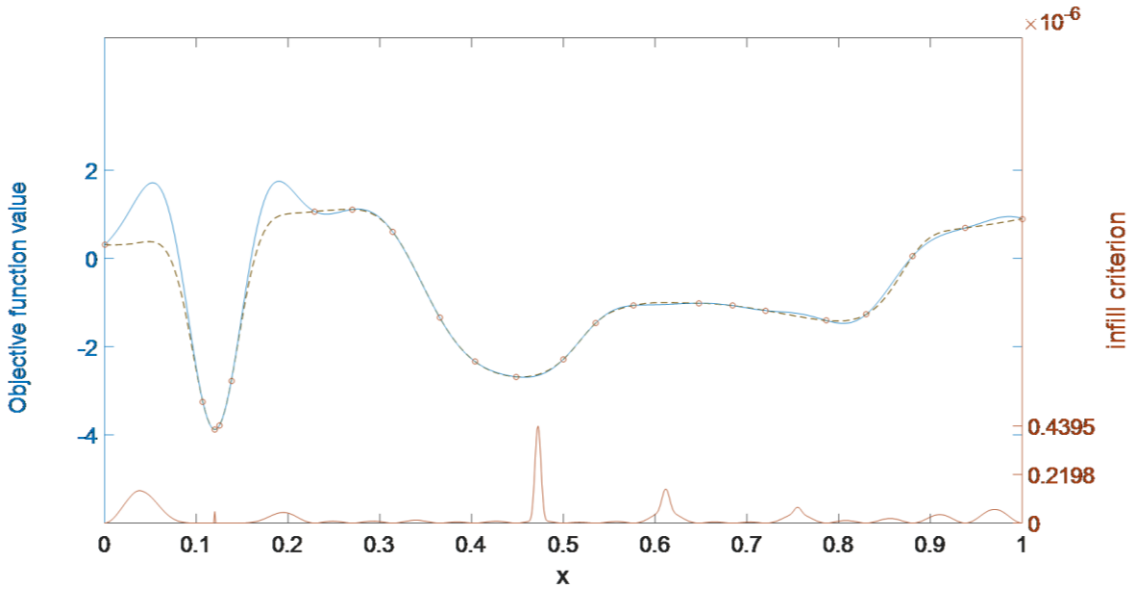
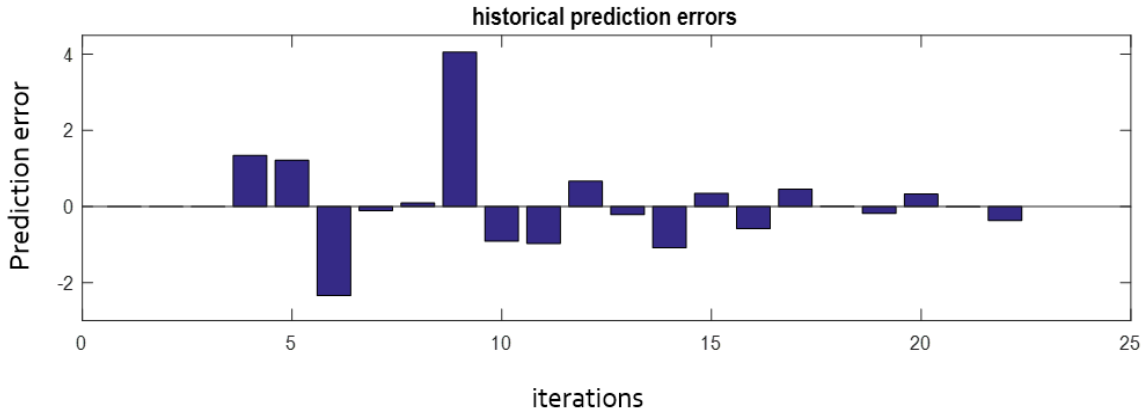
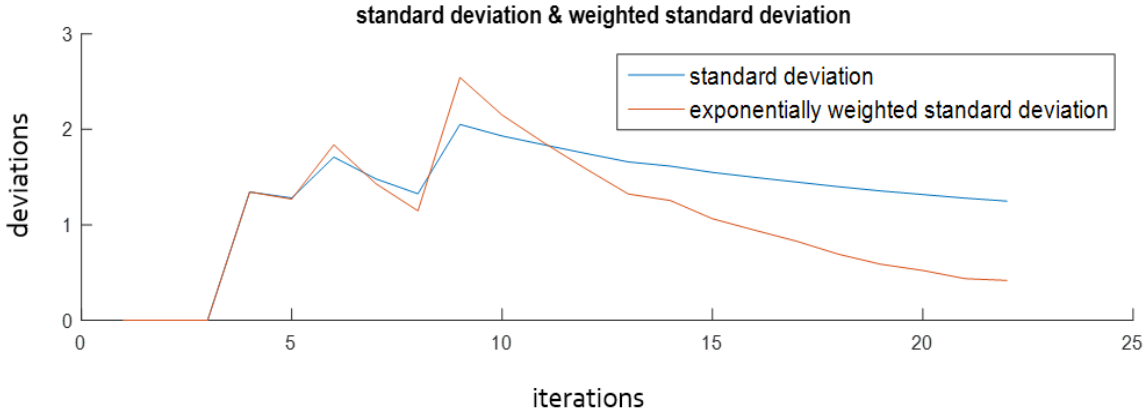


Figure 3.5. Kriging model at the end of the optimisation process.

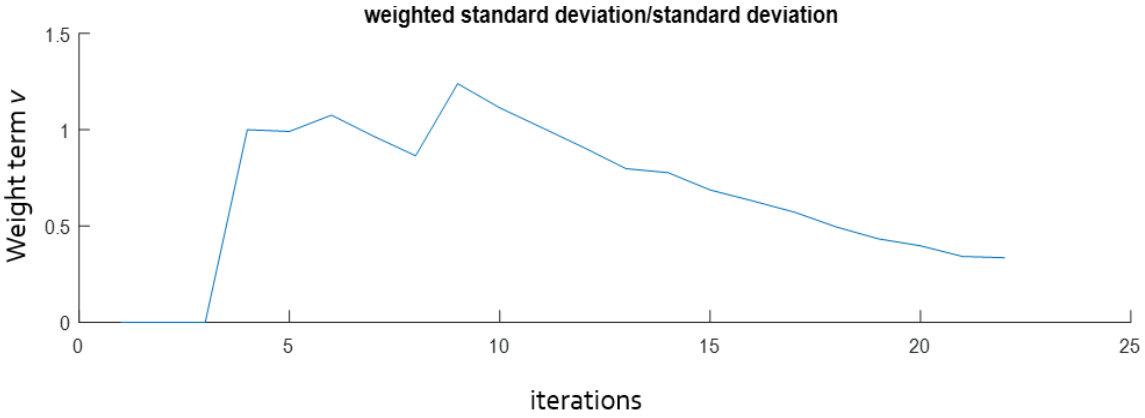
Figure 3.5 displays the shape of the model at the end of optimisation (the program continued to explore the design region after the optimum was found) and Figure 3.6(a) presents the historical prediction error at each iteration. Furthermore, Figure 3.6(b) plots the standard deviation and exponentially weighted standard deviation at each iteration and Figure 3.6(c) is the weight values at the corresponding iterations.



(a)



(b)



(c)

Figure 3.6. (a) Historical prediction error, (b) standard deviation and exponentially weighted standard deviation, (c) weight at corresponding iterations.

As can be seen in Figure 3.6 (a), the large positive historical prediction error at the 9th iteration indicates the objective function value of the new infill sampling points is much smaller than the predicted one in the previous iteration. Larger deviations of historical prediction errors are more often seen during the exploration stage, the prediction error is expected to decrease as more infill sampling points have been added and the kriging model becomes more accurate. As shown in Figure 3.6 (c), as the prediction error decreases (more accurate model), the parameter v, and hence the weight on the exploration term in (3.1), also decrease.

3.4 Test and comparisons

We compared the standard EI sampling method and the proposed sampling scheme in this section. Figures 3.7 and 3.8 illustrate the final surrogate model built using these two infill criteria. Because both EI and the new sampling criterion are deterministic, running multiple optimisation tests on an identical test function will generate identical results; for this reason, the test function used in this chapter includes a noise term. Standard EI infill criterion and the new sampling criterion were tested on an identical test function and then the test function was rebuilt using a random noise term and another set of tests was executed. The test function is as follows:

$$f(x) = a \cdot cos\big(w(x-p)\big)e^{-(w(x-p))^m} + b \cdot sin\left(e^{-(v(x-q))^k}\right) \qquad (3.8)$$
$$+c \cdot cos\big(u(x-r)\big)e^{-(u(x-r))^l} + f_{wave}$$

where $a = -3.5$, $w = 1$, $p = 0.1$, $m = 2$, $b = -1.8$, $v = 0.2$, $q_1 = 0.75$, $k = 6$, $c = -2$, $u = 0.3$, $q_2 = 0.45$, $l = 4$ and f_{wave} is an interpolation function of a set of randomly generated points.

Figures 3.7 and 3.8 illustrate one of the test results. It should be noted that the EI infill criterion does not always converge on a local minimal point; EI typically fails to find the global optimum.

`

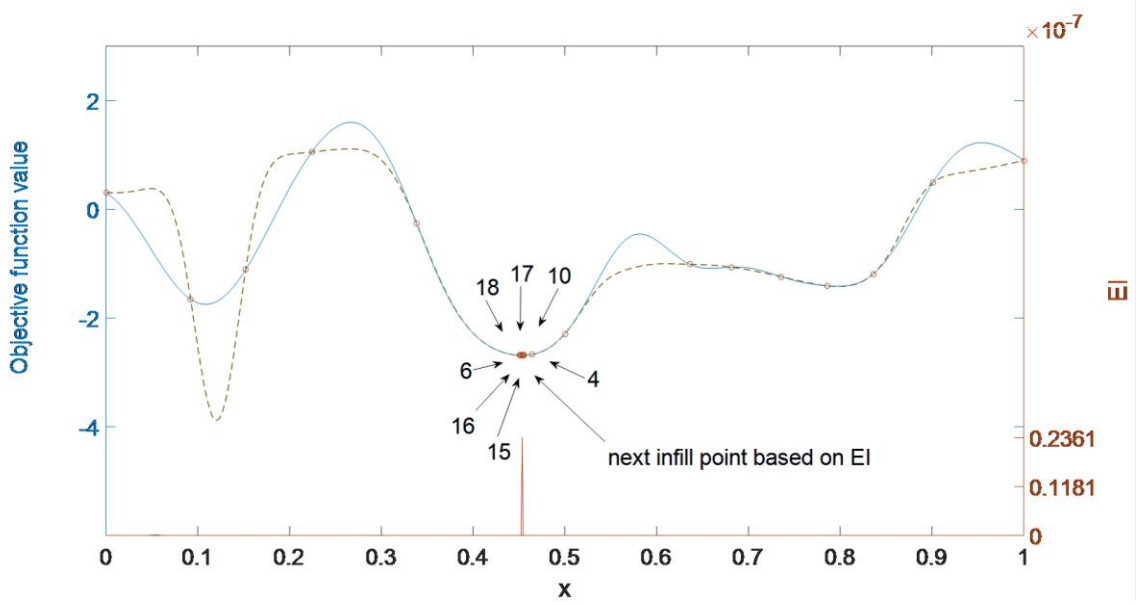


Figure 3.7. Final kriging model (standard EI).

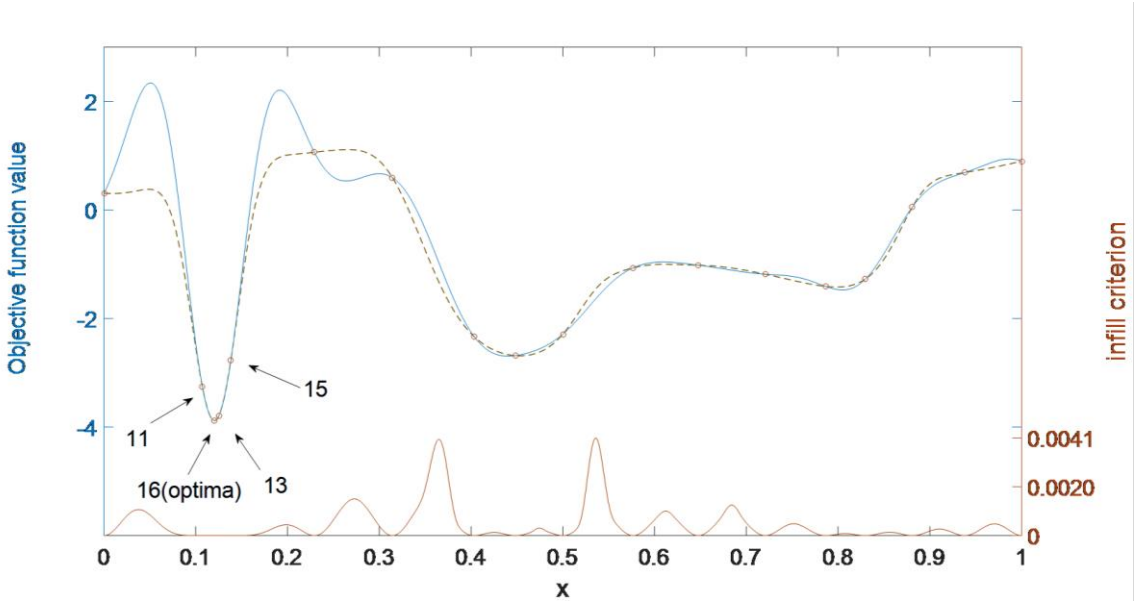


Figure 3.8. Final kriging model (Efficient sampling criterion).

Figures 3.9 and 3.10 illustrate the convergence histories of the standard EI infill approach and the proposed infill approach. The proposed infill approach had a higher success rate of 96% compared to the standard EI approach, which was 74%. Figure 3.9 indicates that the standard EI approach converges faster compared to the new infill criterion, but none of the failed EI processes run more than 25 iterations; this further indicates that duplicate design points were taken during the optimisation process, which is the consequence of over-exploitation. The new infill sampling criterion, on the other hand, converges more slowly but is more reliable and provides a much higher chance of finding the global optimum.

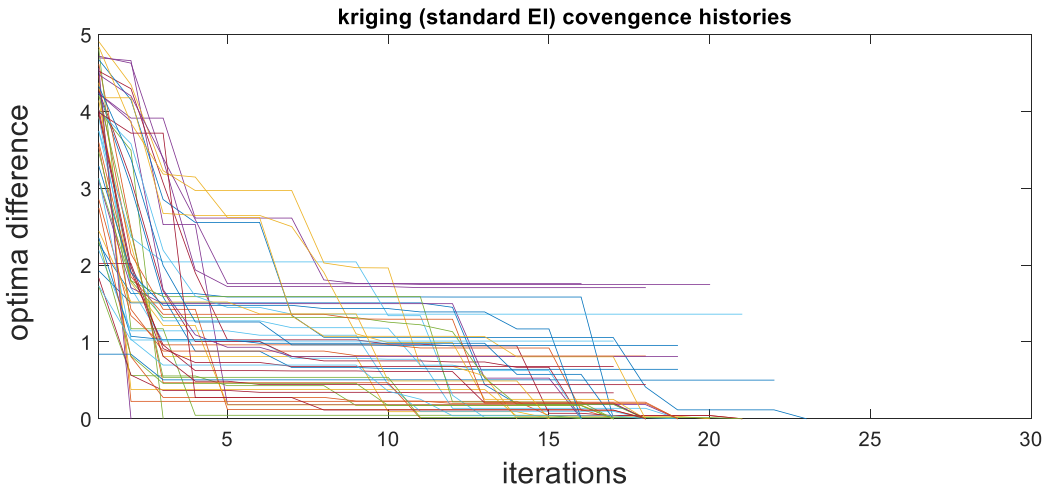


Figure 3.9. Convergence history of a standard EI approach (success rate = 37/50).

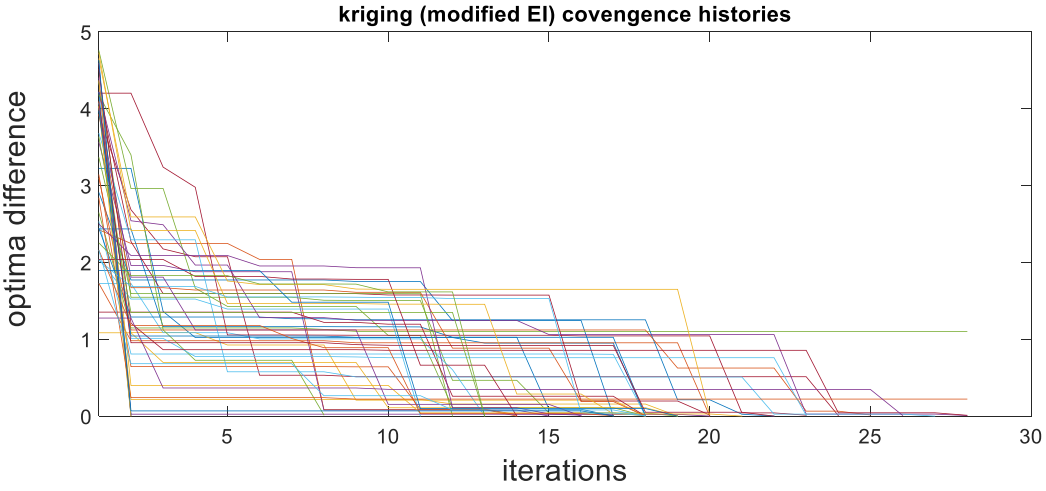


Figure 3.10. Convergence history of a modified EI infill criterion (success rate = 48/50).

3.5 Efficient evaluation of infill criteria

3.5.1 Exhaustive search

In the context of global optimisation (based on surrogate modelling), the location of the next sampling point is based on the infill point sampling scheme, which contains a set of rules or formulae. For example, the standard EI approach uses the predicted function value, the estimated error and the current optimal value as the input variables and the output is the expected improvement for that particular point; the point with the largest expected improvement is the next infill point. In order to reliably locate the point with maximum EI, all points within the design space need to be considered. Thus, the predicted objective function and the corresponding MSE need to be calculated at all points. This may seem straightforward for a one-dimensional problem, but it could

`

create another case of 'combinational explosion' in more practical, multi-dimensional problems if the same sampling interval is maintained for EI. With regard to four dimensions, 1/100 of a step in each direction would require 10^8 calculations to find the maximum EI, which would even create a computational dilemma for faster surrogate models. Conversely, a larger step of 1/10 would be easier to handle, but unlikely to capture the actual optimum.

Figure 3.11 illustrates a one-dimensional example problem, showing the original objective function (dotted brown line) and the kriging model (bold blue line), while the bold cinnabar line at the bottom (with crosses on top) is the true EI curve. Moreover, 101 EI sampling locations were evenly distributed within the design region with a fixed interval of 0.01. The small black crosses on top of the EI curve mark the sampled EI at these predefined locations. As presented in the figure, the sampled EI is a reasonable approximation of the true EI curve, but not overly accurate at the "critical" location around the maximum EI.

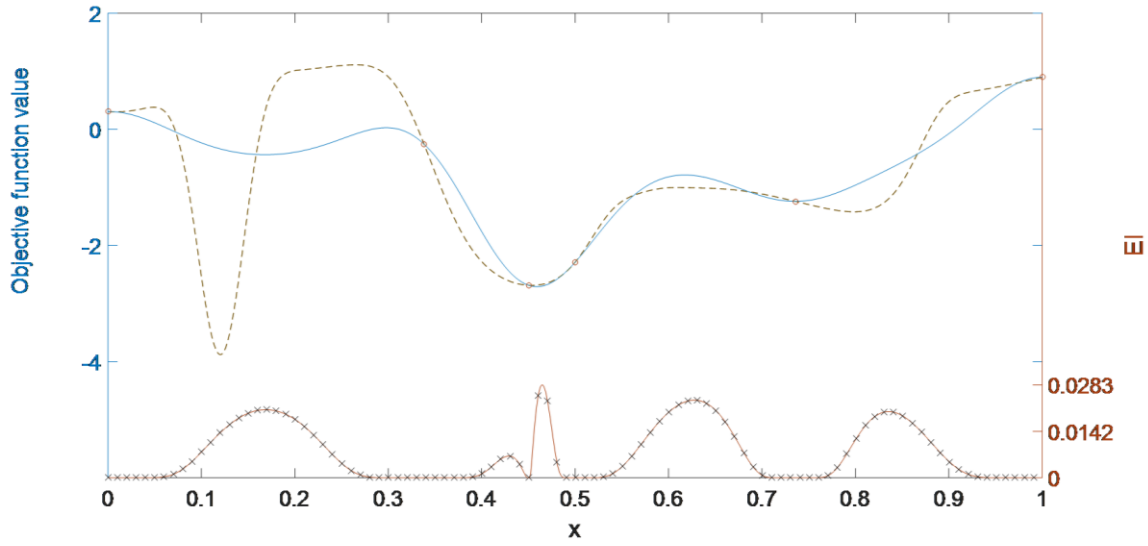


Figure 3.11. EI at predefined sampling locations for a 1D test function

An alternative to exhaustively searching for the maximum sampling criterion value using a predefined step size could be to treat the task as a small optimisation problem in its own right. A direct search global optimisation method (such as genetic algorithm, simulated annealing, particle swarm or gradient-based local optimisation methods) could be employed for such a purpose, which would effectively eliminate the aforementioned step-size problem. The drawback of using any heuristic optimisation algorithm to replace the exhaustive search is that, depending on the algorithm and its settings, there is always the possibility that the algorithm will not locate the optima of the infill criterion function. The deterministic local search methods are more effective and therefore, preferred in this scenario.

3.5.2 Gradient-based search

Evidently, the calculated EI function is always zero at known design points and its optimal value always lies between two existing design points. Thus, it is possible to increase the efficiency of its optimisation by utilising the local gradient-based methods. In this way, roughly half the objective function prediction and MSE calculation can be reduced compared with the exhaustive search method.

In Figure 3.12, the search begins from a known design location and gradually moves to the next design point; each process is terminated when the local maximum EI is found and the gradient becomes negative.

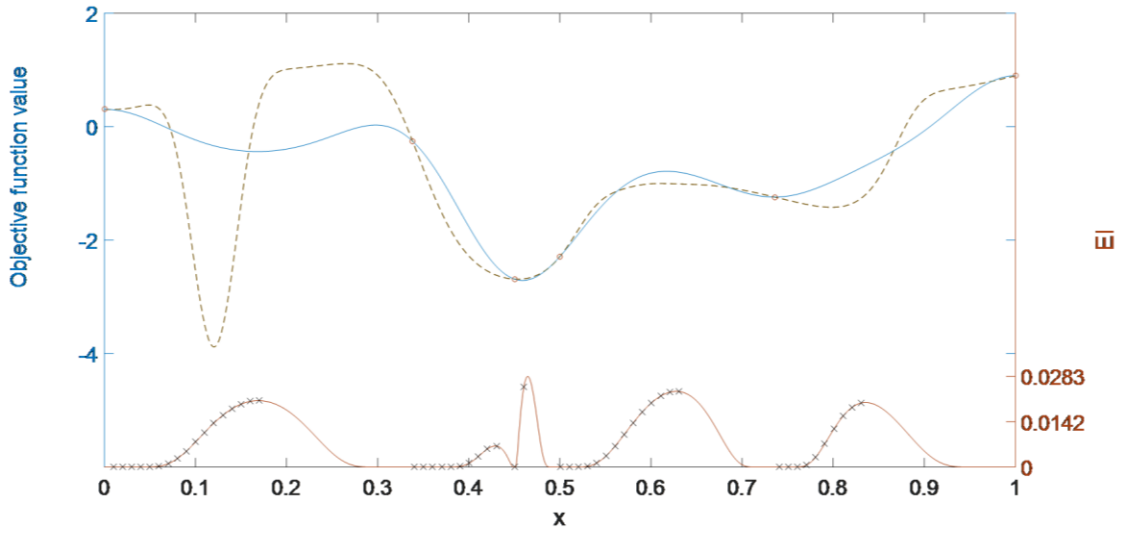


Figure 3.12. Gradient search starts from known sampling points.

Furthermore, a more efficient optimisation method would be to initiate the gradient search in the middle of two existing design points, as this is often the place where the maximum EI is found. This could speed up the process since it would only be necessary to calculate a few predictions of the objective function and the associated MSEs. As illustrated in Figure 3.13 below, the corresponding mid-point of two adjacent design points on the EI curve is marked by an "**x**". The search began at this location and moved toward both the left and right until the gradient became negative (or positive for the left side). The total number of EI calculations decreased by 75%.

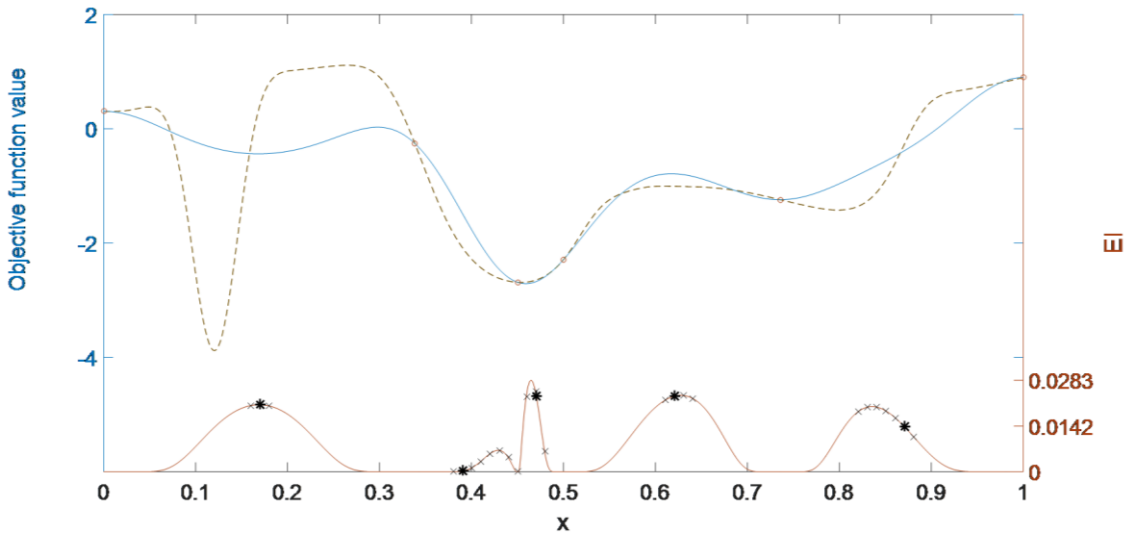


Figure 3.13. Gradient search starts from the mid-point of two sampling points.

As can be seen from Figures 3.11, 3.12 and 3.13, even 100 sampling points on the EI curve in a one-dimensional problem cannot locate the maximum EI with any reasonable accuracy. By improving the efficiency of the sampling process for the EI function, sampling points can be concentrated on more important areas and consequently, can locate the maximum EI more accurately, as illustrated in Figure 3.14 below; while the predefined density of the sampling points for the EI is doubled (200 points), only 50% of the EI evaluation is required.

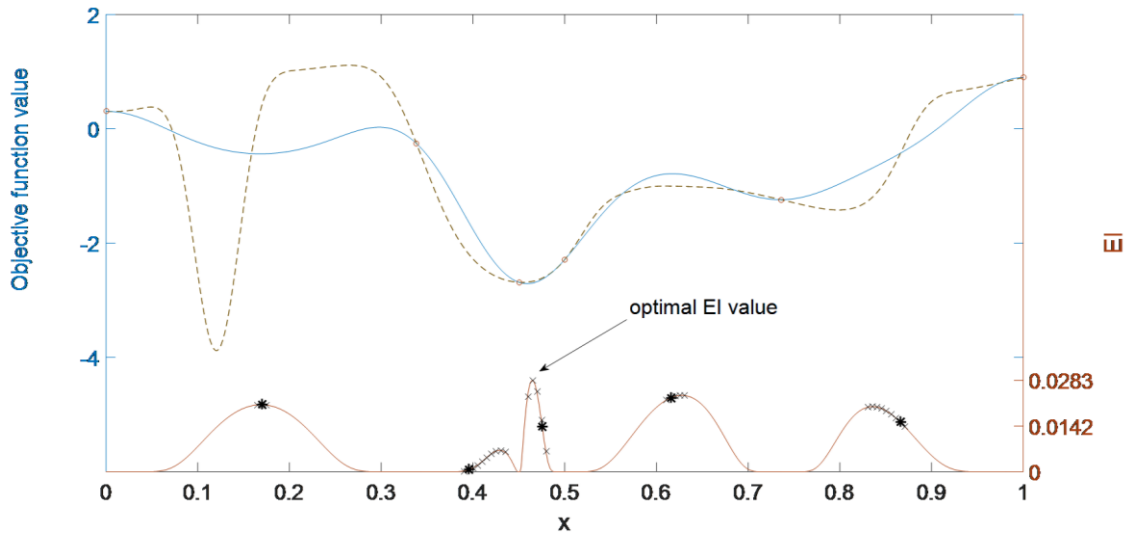


Figure 3.14. Gradient search starts from the mid-point of two sampling points (sampling density doubled).

Applying the same methodology to our proposed infill sampling approach, we obtained similar results. The 4th and 7th iterations are presented in Figures 3.15 and 3.16.

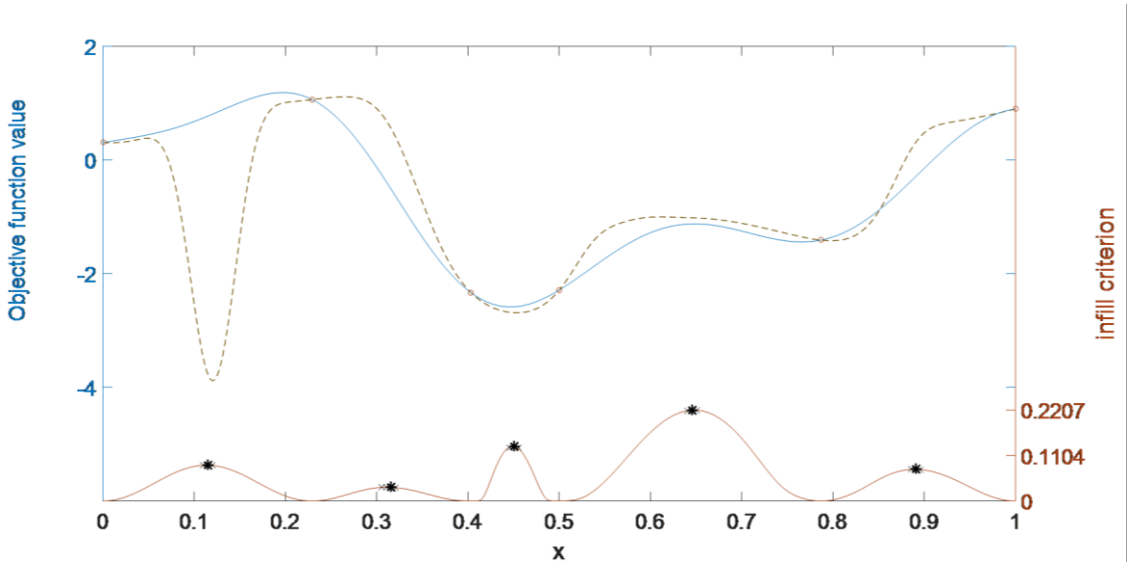


Figure 3.15. Gradient search starts from the mid-point of two sampling points (modified EI criterion at 4th iteration).

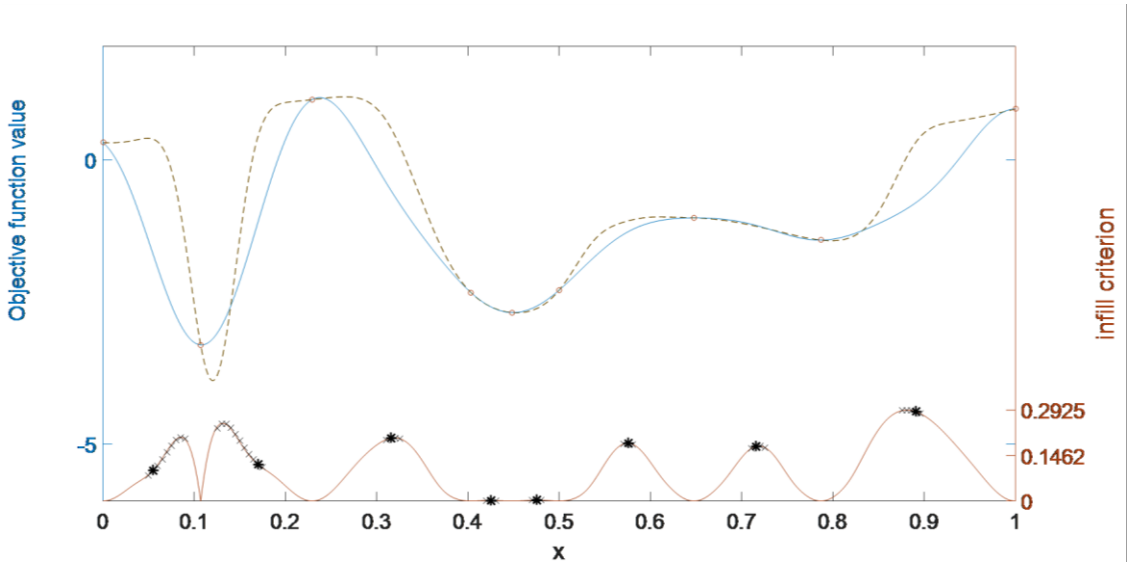


Figure 3.16. Gradient search starts from the mid-point of two sampling points (modified EI criterion at 7th iteration).

It is worth mentioning that in exceedingly rare scenarios – when there is a small number of known design sites in the design region (2-3 points in a 1D space) – the EI may contain two local maximum values between two adjacent design sites. Because this scenario only occurs in the beginning stage of the modelling process, when the sampling points are sparse, it could be countered by taking more initial sampling points or by using exhaustive searches in the beginning.

For a multidimensional design space, in which the local maximum EI is located between multiple existing design sites, a multi-start strategy can be applied to replace a single starting point (as in the 1D case) by simply generating starting points in the middle of each pair of two existing design sites. This is a combination problem and the number of

`

starting points *p* for *n* known points is *p = nC*2. The formula for combination problems is as follows:

$$nCr = \frac{n!}{(n-r)!} \times \frac{1}{r!}$$

where the symbol "!" stands for the factorial function. Hence, for 100 existing design points, with starting points located between each pair of existing design sites, the number of starting points p is:

$$p = 100C2 = \frac{100!}{(100-2)!} \times \frac{1}{2!} = \frac{100 \times 99}{2} = 4950$$

Hence – in this case – 4,950 starting points for the gradient descent calculations are generated.

3.6 Conclusion

A method for quantifying the model quality was presented in this chapter and a new infill sampling criterion was introduced. The new criterion utilises the information of model quality to dynamically self-balance the weight on exploration and exploitation and to perform better in the test problems, both in terms of efficiency and convergence rate.

A basic gradient search method for infill sampling criterion evaluation was discussed; the new approach improved the accuracy of the optimisation process of infill sampling criterion, while significantly increasing the efficiency.

`

`

Chapter 4. Robust optimisation

4.1 Introduction

Problem formulation:

$$\text{Minimise } f(\boldsymbol{x}) \tag{4.1}$$

$$\text{Subject to } x \in D$$

$$g_i(\boldsymbol{x}, \boldsymbol{u}_i) \leq 0,$$

$$\boldsymbol{u}_i \in U_i, \qquad i = 1, 2 \ldots, n.$$

where $f(x)$ is the objective function to be optimised, $g(x)$ is the constraints and $\boldsymbol{u}_i$ is the uncertainty set.

Since the objective function is a description of the relationship between the input parameters and the output results in real-world problems, uncertainties are presented in the majority of situations. These uncertainties can be the result of manufacturing accuracy, material quality or environmental effects; to deal with the uncertainty, for example in the process of manufacturing a product, the manufacturer may wish to establish certain standards to control the quality of the product, often known as the design tolerance. A product's quality is acceptable within this tolerance range, while others may be recycled or abandoned. To achieve the optimum output, with consideration for tolerance, optimisation with tolerance is proposed; this class of optimisation problems is sometimes called tolerance design optimisation or simply, robust optimisation.

The history of robust optimisation can be traced back to 1989 when Taguchi first introduced the concept of the quality of a design [31]. He introduced the signal-to-noise ratio (SNR):

$$SNR = -10log_{10}(MSD) \tag{4.2}$$

where MSD is the mean squared deviation in objective function value y, and

$$MSD = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{4.3}$$

where n is the number of observations, y_i is the corresponding objective function value and $\hat{y}$ is the target point at which robustness is evaluated.

The target with the minimal objective function variance is obtained by maximising SNR. The problem with Taguchi's method is that it does not scale well with higher dimension

`

problems; for an N dimensional/variables problem, the objective function evaluation is in order of 2^n.

While the general mathematical theories of robust optimisation algorithms have been studied, the algorithm is generally directly implemented in practical problems. One example of recent research results is [32]. Recent published papers and journals have indicated that the development of robust algorithms tends to focus more on specific real-world problems. Depending on the type of uncertainty, they can be further classified into different classes of robust optimisation problems.

4.2 Robustness measures

Possible source of uncertainties [33]:

- Environmental/external sources:
  Changes in the external environment are not described in the objective function.

- Imperfection of the mathematical model:
  The objective function often is only an approximation of the design and may not be perfectly mapped to the practical problem.

- Design tolerance/output uncertainty:
  Machines in the real-world situation are manufactured up to a certain degree of precision. Even if the optimum solution is known, the production line may not be capable of producing the exact design.

Five types of robustness measures have been classified in [33] based on different types of uncertainties.

- Deterministic robustness measure
- Expected robustness measure
- Probabilistic threshold robustness measure
- Statistical feasibility robustness measure
- Possibilistic robustness measure

Deterministic robustness measure: The method implemented for this type of measure is the worst-case approach; the worst uncertainties are taken into consideration, forming an individual function in addition to the objective function that needs to be minimised, called the robust counterpart function. This type of robust design was considered in this project since the topic is subject to the robust design under deterministic uncertainties.

`

Expected robustness measure: The uncertainty obeys a distribution function (which is usually obtained based on statistics or experience). The robustness of the result is based on the expected value of the robust counterpart function. This type of problems are often solved as a multi-objective optimisation problem with the variance of the design being the additional objective.

Probabilistic threshold robustness measure: This type of robustness measure is defined by the probability of the objective function that is smaller than a predefined threshold at a given location via repeated sampling.

Statistical feasibility robustness measure: statistical feasibility robustness measure shares some similarities with the probabilistic threshold robustness measure; however, it handles uncertainties which are related to the constraints, and its robustness measure is defined by the probability of fulfilment of constraints.

Possibilistic robustness measure: Unlike the other 4 types of robustness measures which are based on the complete information of the problem, uncertainties come from potential error sources are classified as possibilistic uncertainties; the information of these uncertainties are based on subjective estimations.

More information about robustness measures can be found in [33], where the author has discussed each type of robustness measures in detail together with various types of uncertainties. In addition, some discussion on methods for handling problems with different type of robustness measures have also been given.

Robust optimisation problems are distinguished by the robustness measure of the design; optimisation approaches to the problem depend on how uncertainty is treated. The first two types of robustness measures are more commonly found within the literature. In this chapter, we consider the first type of robustness measure, namely the deterministic robustness measure, to be the conservative approach to robust optimisation design. This type of problem for robust optimisation is also referred to as tolerance design.

Unlike standard global optimisation problems, complex engineering robust optimisation designs are often problem-specific. Depending on the particular problem, the uncertainty measure is defined specifically and the solution varies accordingly. For expected robustness measures, the design problem can be treated as a multi-objective optimisation problem [32] [33] [34], where the uncertainty (variance) is another objective to be optimised along with the objective function (mean). The robust solution is then chosen from the Pareto front by the decision maker. When the uncertainty distribution variable is unknown, a more conservative approach is to solve the problem in the worst-case scenario.

The work of Song Xiao on Kriging has developed several methods to address the robustness of the design based on the surrogate model, including the six sigma quality approach and enhancements made to improve the performance of the worst-case

`

optimisation method and average performance assessment [35] [36]. The six sigma approach has been found to outperform other methods in assessing the robustness measure of optimal solutions [35].

The output *f* of a black-box function, when the input variable *x* contains deterministic types of uncertainties, can be expressed by a simplified equation (ignoring possible other sources of uncertainties and assuming the uncertainty ε is independent of the input variable *x*) [33]:

$$f = f(x + \varepsilon) \tag{4.4}$$

where $\varepsilon[-\varepsilon, \varepsilon]$, the distribution of uncertainty ε, is unknown, but the magnitude is bounded by a given range ε.

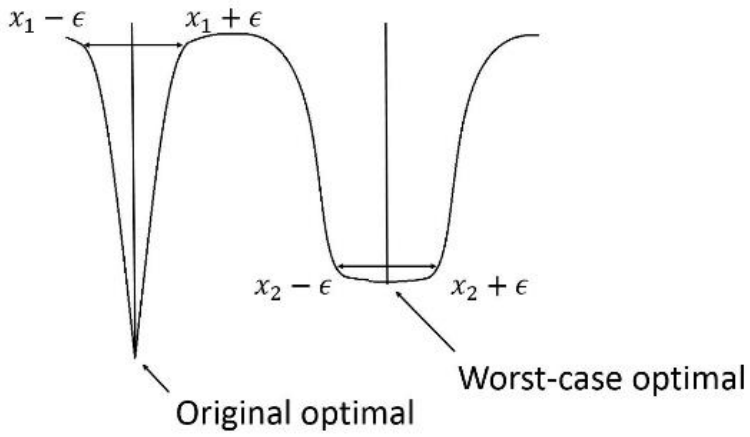


Figure 4.1. Worst-case problem 1D examples.

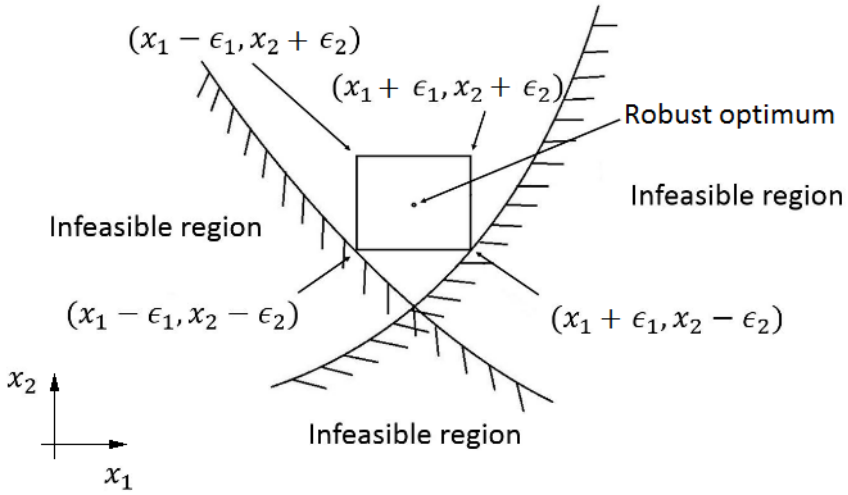


Figure 4.2. Worst-case problem 2D examples with constraints.

Two simple examples are demonstrated in Figures 4.1 and 4.2. The 1D illustration reveals that, depending on the size of the uncertainty ε, the preferred worst-case robust optimum may differ from the original (theoretical) optimal point. This is due to the fact that when the parameter varies within the specified limits (e.g. imposed by

`

manufacturing tolerances), the performance worsens significantly, whereas the robust optimum ensures good performance throughout. The 2D example illustrates that the worst-case optimum needs to consider design constraints, as the optimum solution with uncertainties must not violate the infeasible region.

4.3 A brief review of existing approaches

For a deterministic type of uncertainty, the basic approach is to transform the robust optimisation problem into a standard optimisation problem by optimising the worst-case of the original objective function (multiple objective function evaluations are needed at each design stage). The number of objective function calls may be significantly increased and many unimportant and possibly nearly duplicated design points will be allocated during this process, thus making the optimisation extremely inefficient. This large number of function calls will be of particular concern to designers, especially when the objective function is expensive to evaluate, which is often the case in electromechanical or electromagnetic design where the main tool for field modelling involves numerical computation (such as finite elements).

Recently, a number of more efficient kriging-based approaches for solving worst-case optimisation problems have been proposed within the literature. The authors of [37] use the mean and variants to assess the robustness, while their proposed strategy utilises the gradient information computed from the kriging model. In [38], the EI infill sampling approach is combined with a relaxation procedure based on a kriging model. In [39], the EI infill sampling approach is applied to the worst-case response surface and calculated based on the kriging model.

4.4 Worst-case problems

Worst-case problems are also known as min-max problems; these two names are used interchangeably in the literature. The name describes the two main components within the optimisation process, that is solving a min-max problem is to find the location of the minimal worst-case objective function, where the worst-case objective function at a given location is equal to the maximum objective function value within the design tolerance of that location.

A 1D-worst-case problem is presented in Figure 4.3; the tolerance in this case is $x \pm 0.04$. The grey line shows the original objective function, and the brown dashed line shows its worst-case objective function.

`

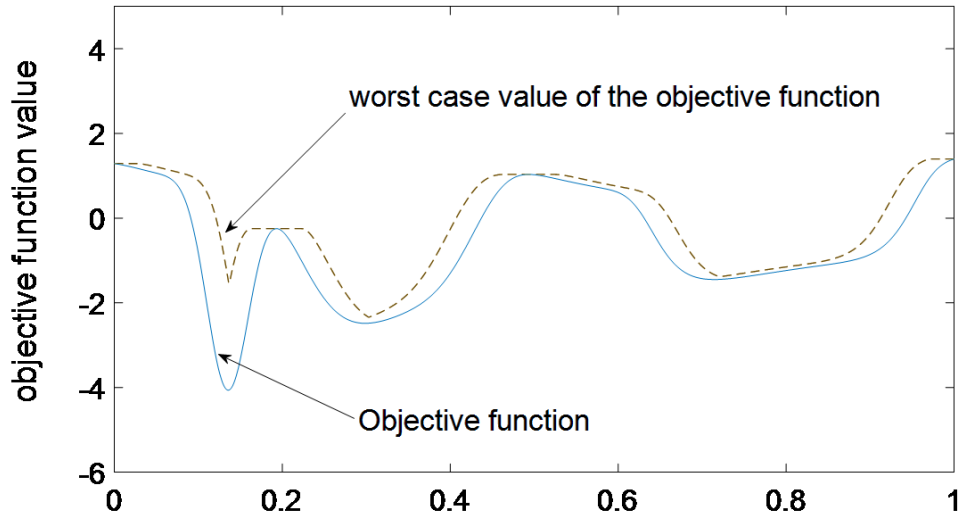
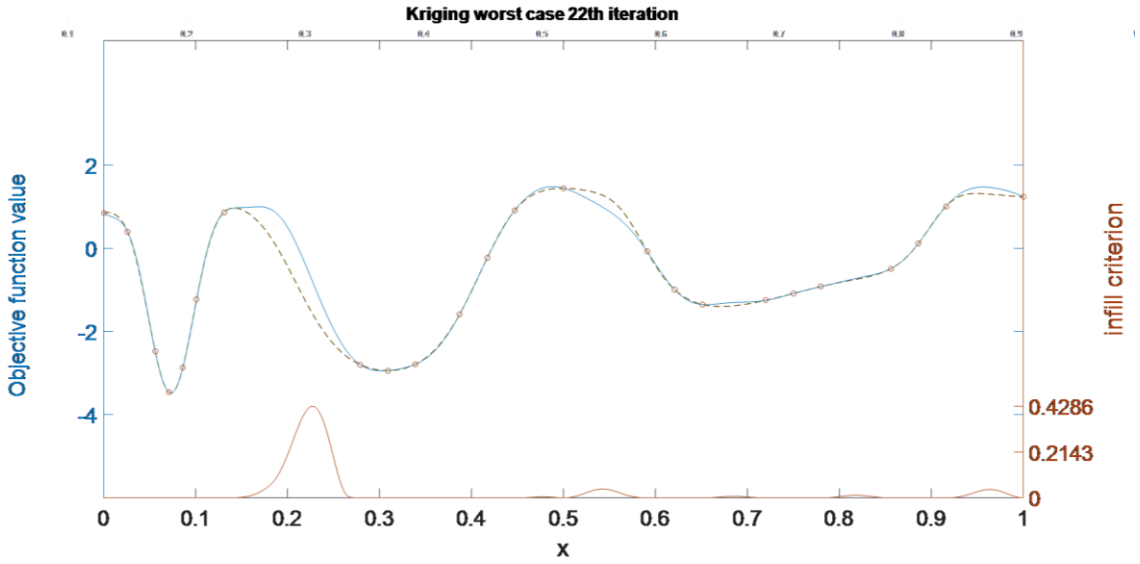


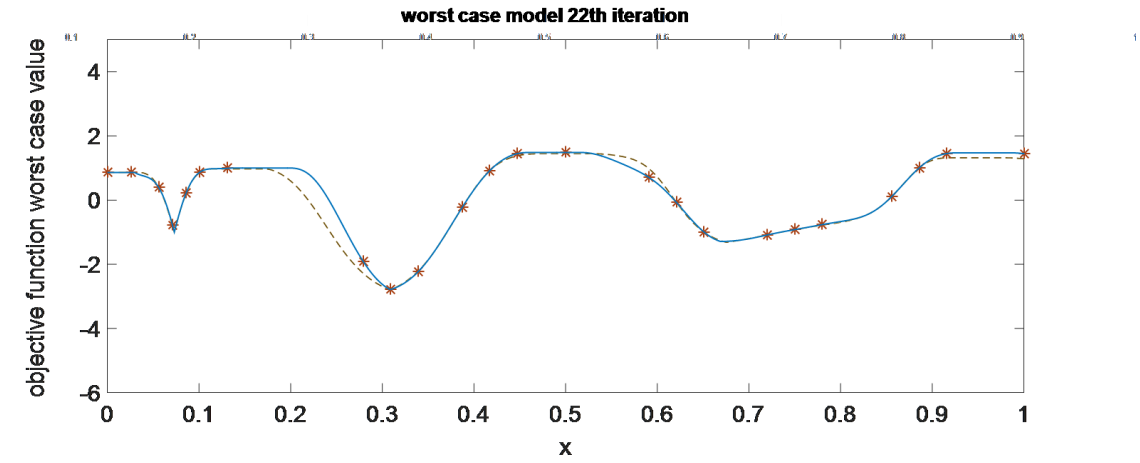
Figure 4.3. Original objective function and its worst-case objective function.

One of the difficulties of robust optimisation using a surrogate modelling approach is that the surrogate model is built based on the known sampling location of the objective function, without counting the robustness measure and after accounting for the worst-case scenario. The geography of the response surface can be quite different from the original objective function and this response surface is not known until the worst-case measure is applied to existing design sites. Therefore, the infill criteria based on the surrogate model might be misleading if the tolerance is comparatively large, resulting in inefficient sampling point allocation.

A majority of the approaches used within the literature determine the robustness measure of the problem using the robust counterpart, where the uncertainty is added to the objective function; the robustness of this design site is evaluated after the point is taken. In the worst-case scenario, two additional sampling points are taken to compute the robustness counterpart. This approach is extremely inefficient, especially for computationally expensive design problems. An example of this approach is illustrated in Figure 4.4. In this chapter, we propose a more active approach that takes the robustness measure into consideration during the infill sampling process.



(a)



(b)

Figure 4.4. (a) Kriging model based a passive approach, (b) worst-case estimation.

4.5 A two stage approach

This section presents a two-stage approach for solving computationally expensive, worst-case optimisation problems. We focus on maximising the usage of available information, while delaying the calculation of the worst-case value at sampling points to achieve a more efficient sampling scheme for the worst-case type of robust design optimisation.

The worst-case optimisation problem is often referred to as the minimax problem and is characterised by an extra “layer” of optimisation. Therefore, the infill sampling criteria for global optimisation are often found inappropriate within the context of the worst-case optimisation problems. The worst-case value of the objective function at any given

point does not depend on information given by that point alone (including the kriging prediction, mean squared error MSE, gradient, etc.), since information from its neighbouring points also needs to be taken into account.

4.5.1 First stage

The algorithm consists of two stages: in the first stage, the kriging model is updated by sequentially adding infill points at each iteration based on the worst-case expected improvement (WCEI) – this expected improvement measure is recalculated from standard EI, by taking the minimal EI value within the worst-case region of that design point (design site):

$$WCEI(x) = \max\{\min[EI(x+\varepsilon)], 0\} \qquad (4.5)$$

$$x + \varepsilon \in X$$

where X is a set of points located within the worst-case region of the unknown point x. A 1D example is illustrated in Figure 4.5, where the boundary ε of the worst-case design is ±0.3.

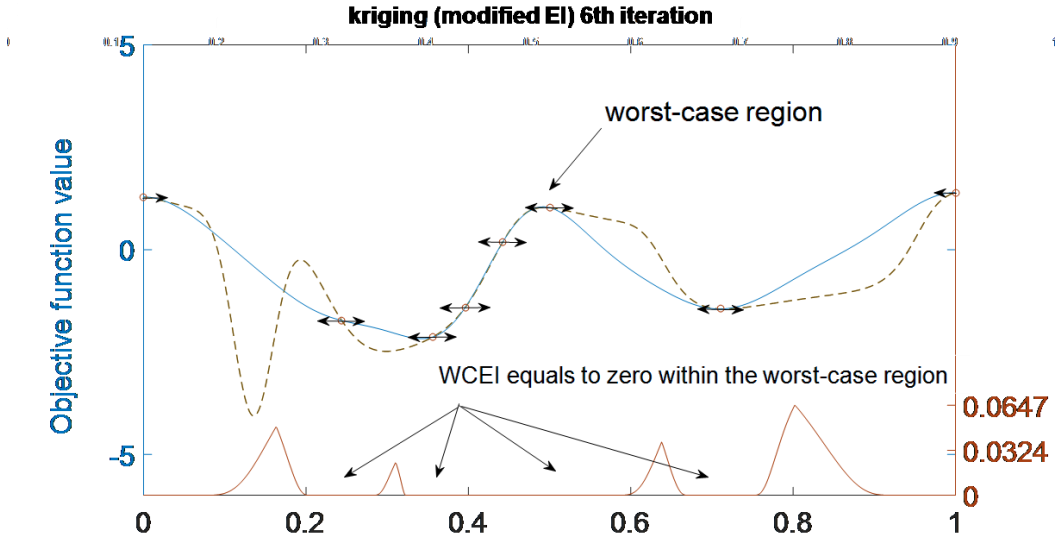


Figure 4.5. The worst-case regions of existing design sites in a 1D example.

The extra layer of the minimax problem is embedded within the WCEI; the new infill sampling point will be located where the minimal expected improvement around the target point is the largest. The WCEI is equal to zero at the locations within the worst-case region of existing design sites; consequently, these areas are banned from being future infill locations at the model updating stage. During the process of model updating, the worst-case estimation of the objective function is computed simultaneously based on the kriging model that was constructed using the existing design sites at that iteration.

4.5.2 Second stage

The second stage is triggered when the maximum WCEI within the design space becomes less than a predefined value or when stage one has exceeded its allowance, if such a limit has been imposed. An exploitation process occurs in stage two; the worst-case region around the worst-case optimum is exploited and validated using a modified EI approach, where instead of calculating the improvement, an expected "deterioration" is estimated to indicate where the maximal worsening is located within the worst-case region of the worst-case optimum

$$E[D(x)] = \begin{cases} (\hat{y}(x) - y_{wc})\Phi\big(u(x)\big) + \hat{s}\phi(u)\,, & s > 0 \\ 0, & s = 0 \end{cases} \tag{4.6}$$

$$u = \frac{\hat{y}(x) - y_{min}}{\hat{s}(x)} \tag{4.7}$$

This process is repeated until the value of the expected deterioration is zero or smaller than a predefined value; at this stage, the location of the worst-case estimated optimum is added as the next infill point and the associated objective function is evaluated. When the range of the underlying objective function surface is large, both the location and value of the actual worst-case optimum can differ from the estimated one. Therefore, the above validation process provides a more accurate prediction within the area of interest and thus, helps the program efficiently and accurately locate the best worst-case optimum.

4.6 Examples

4.6.1 Test function

The worst-case optimisation routine following the two-stage approach is first illustrated using a 1D test example. Figure 4.6 presents the original test function and its associated worst-case distribution; for the purpose of illustration, the design tolerance is assumed to be ±0.035. It can be observed that both the landscape and, in particular, the position of the optimum differ noticeably between the original function and the worst-case version.

`

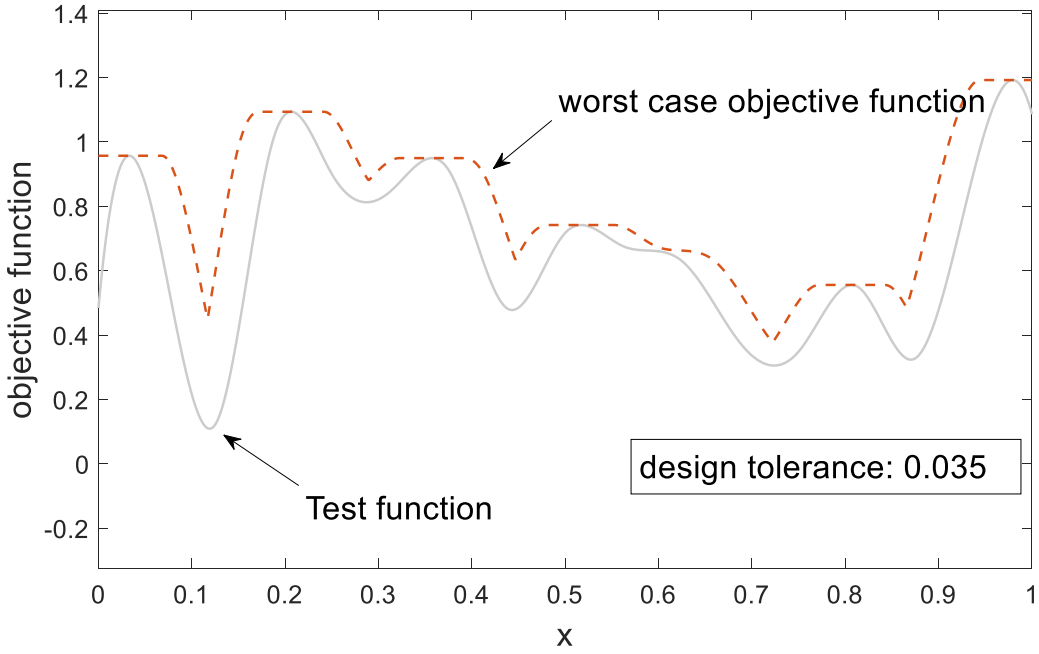


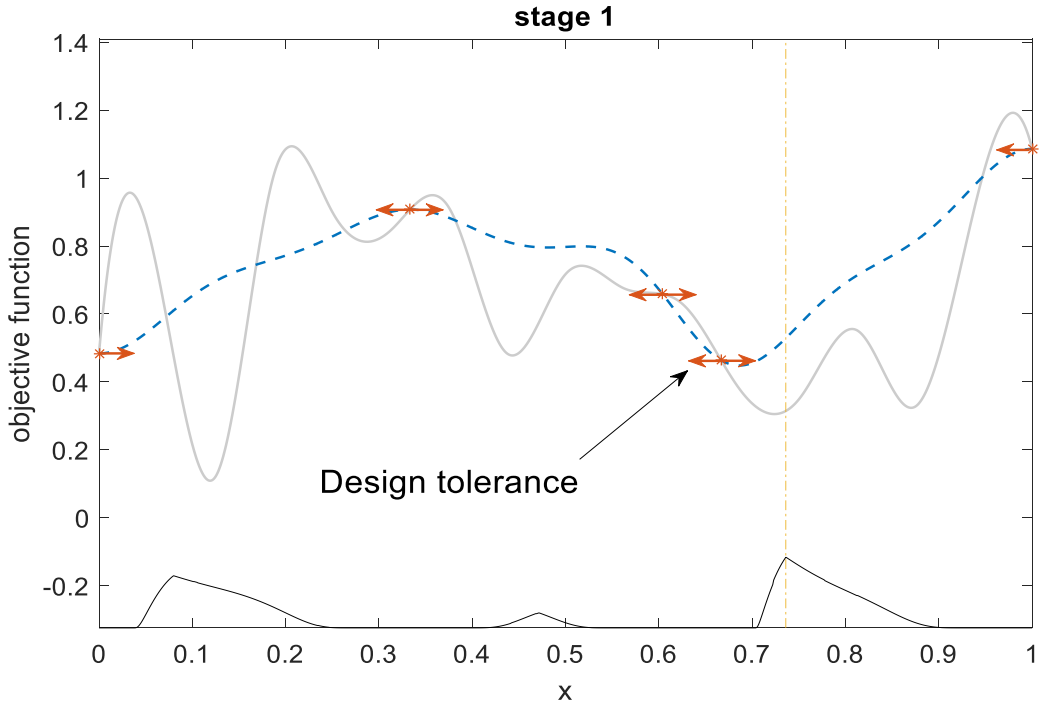
Figure 4.6. The original test function and its worst-case value.

4.6.2 Stage-one

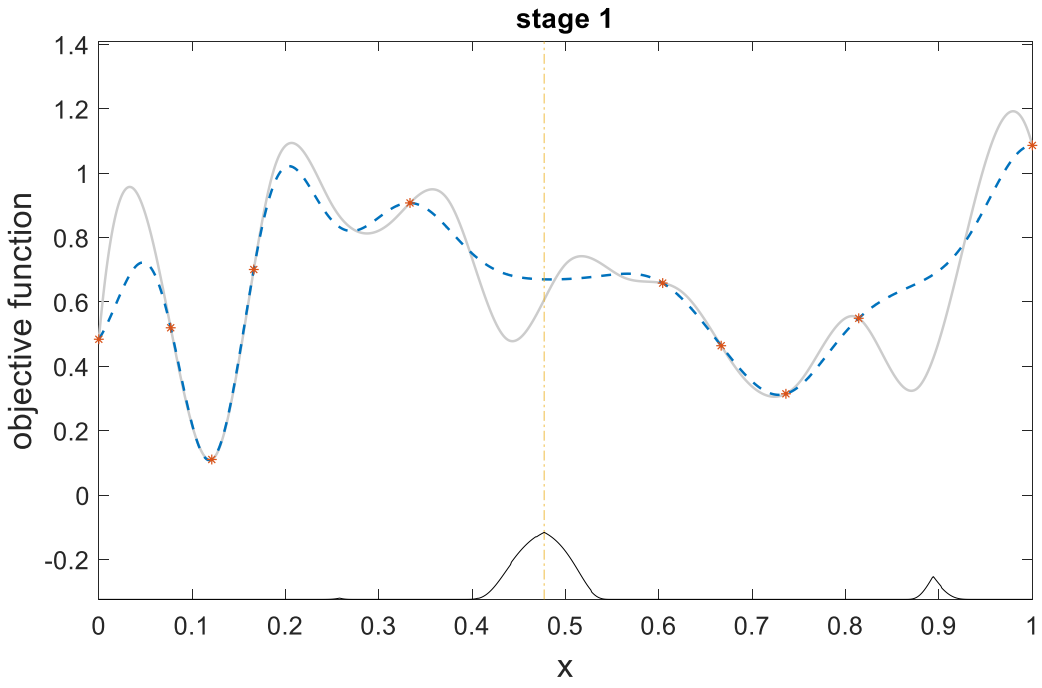
In Figures 4.7 (a) to (f), the grey solid line depicts the objective function, while the blue dashed line represents the kriging prediction. The black solid line at the bottom of the figures shows the scaled values of infill criteria.

Figure 4.7 (a) illustrates the design tolerance and the initial kriging model with 5 sampling points, during stage-one, where new infill points are added based on the WCEI criterion. Figure 4.7 (b) shows the kriging model at the 10^{th} iteration; it provides a better approximation since more sampling points are added. As shown in Figure 4.7 (c), the maximum WCEI within the design space is less than the predefined value of 10^{-3}*, the optimisation solver then enters stage-two.

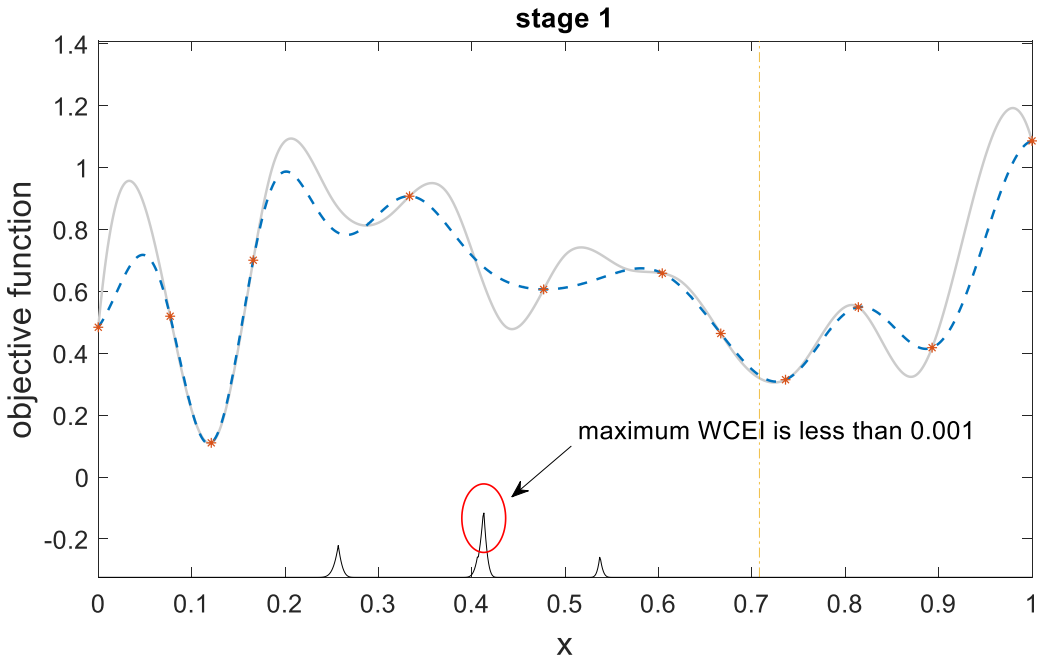
Note: This value can be specified by the designer based on the information available from the design problem; a smaller value will result in more infill points been added in stage one, consequently more exploration and exploitation before stage two; 10^{-3} is a reasonable value for the normalised test function in our case.*


stage 1
objective function
x
Design tolerance
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(a)


stage 1
objective function
x
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(b)



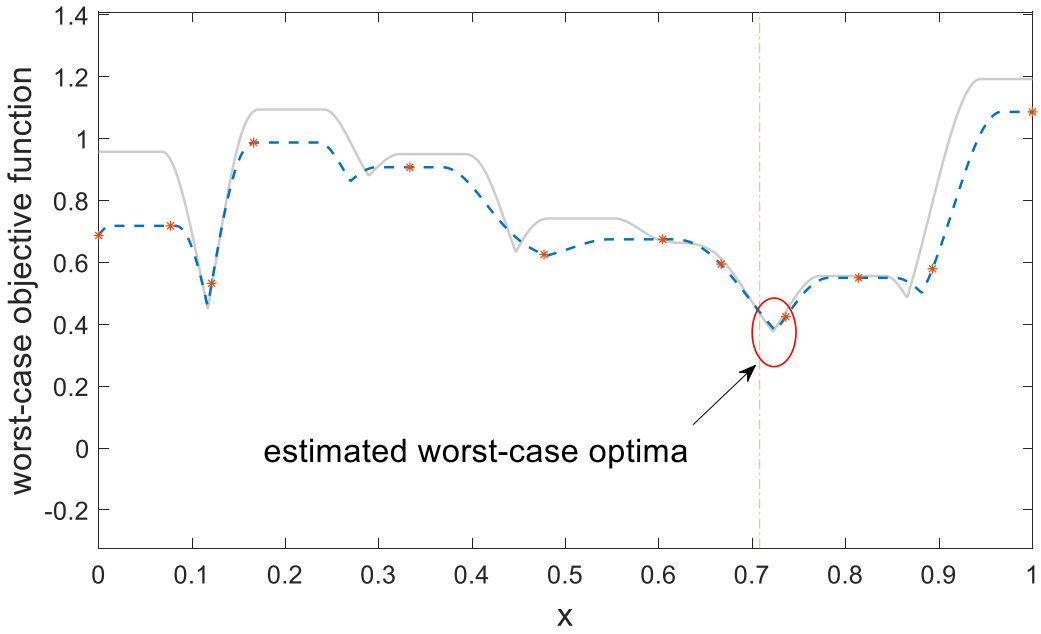
(c)

Figure 4.7. (a) Design tolerance and initial kriging model, (b) kriging model with 10 sampling points, (c) kriging model with 12 sampling points.

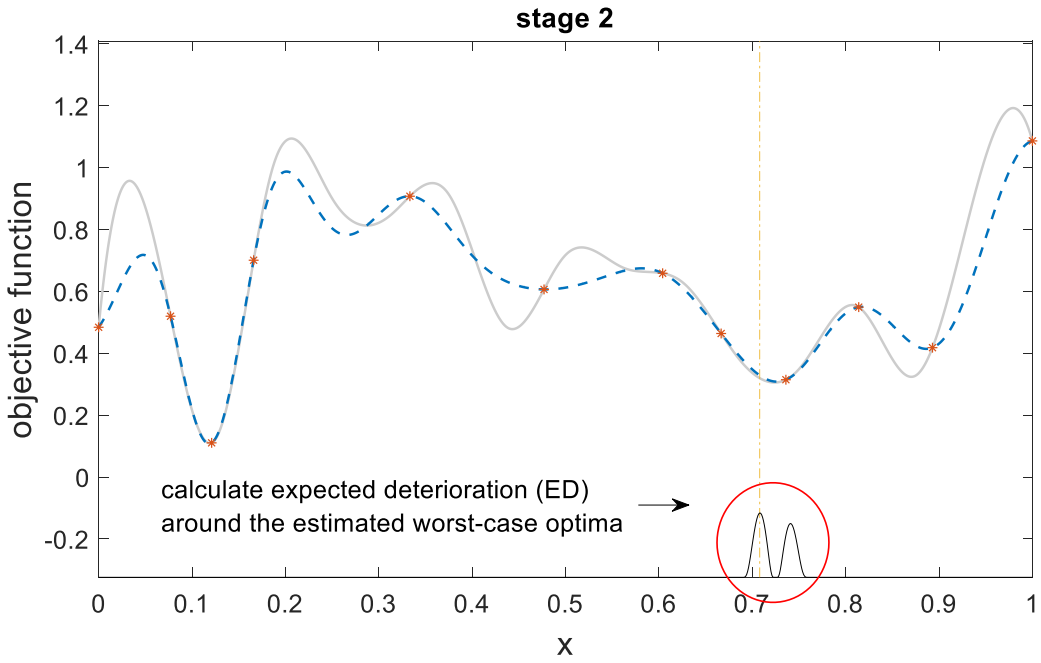
4.6.3 Stage-two

The optimisation solver enters stage two and the region around the estimated worst-case optimum at $x = 0.74$ (see Figure 4.8(a)) is exploited. The ED infill criterion in Figure 4.8(b) depicts the value of expected deterioration within this region.

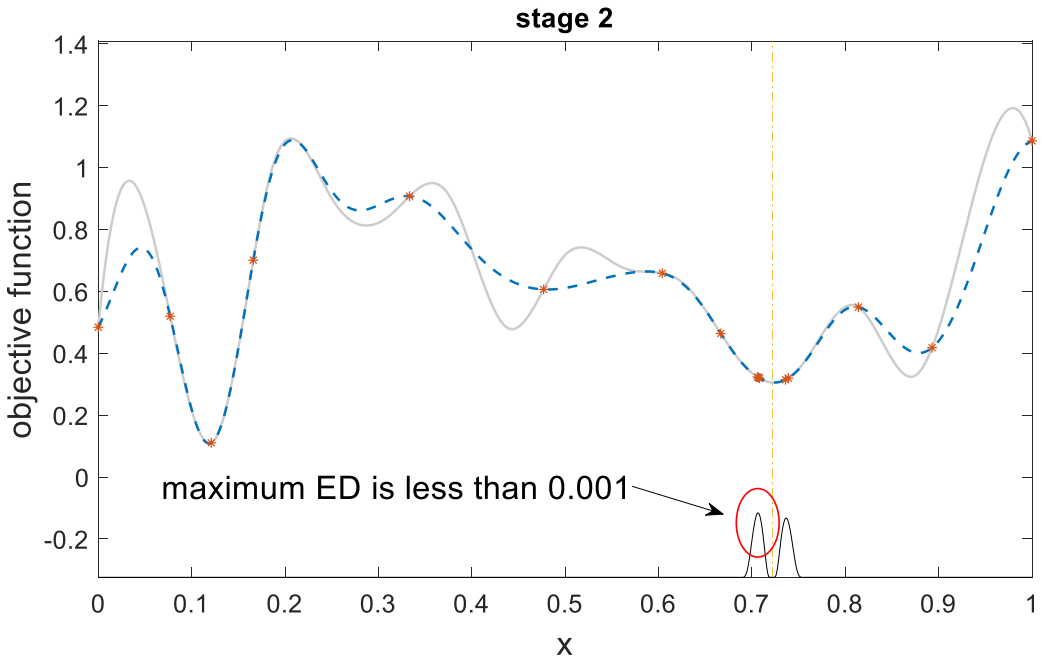
The kriging model is updated during the validation process, as in Figure 4.8 (c); the value of the expected deterioration is smaller than the predefined value 10^{-3} at the 15^{th} iteration; the final worst-case optimum is then located at $x = 0.72$ in Figure 4.8 (d). The last step may be repeated multiple times if the updated kriging model shows a different estimated robust optimum from previous iterations. Figure 4.8 (e) presents the final shape of the estimated worst-case objective function and the underlying worst-case objective function.


worst-case objective function
x
estimated worst-case optima

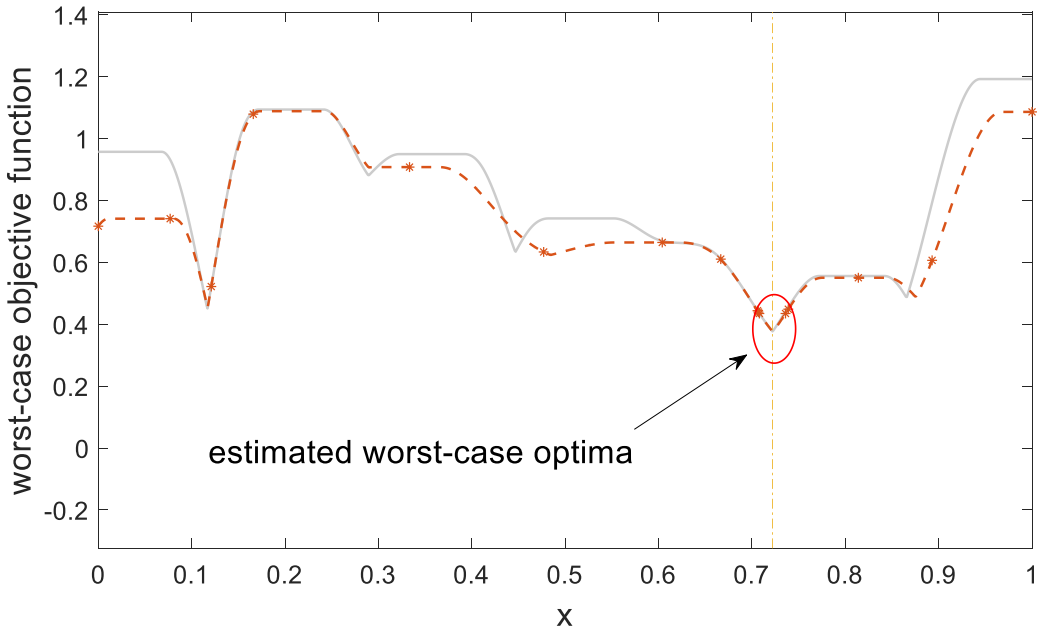
(a)


stage 2
objective function
x
calculate expected deterioration (ED)
around the estimated worst-case optima

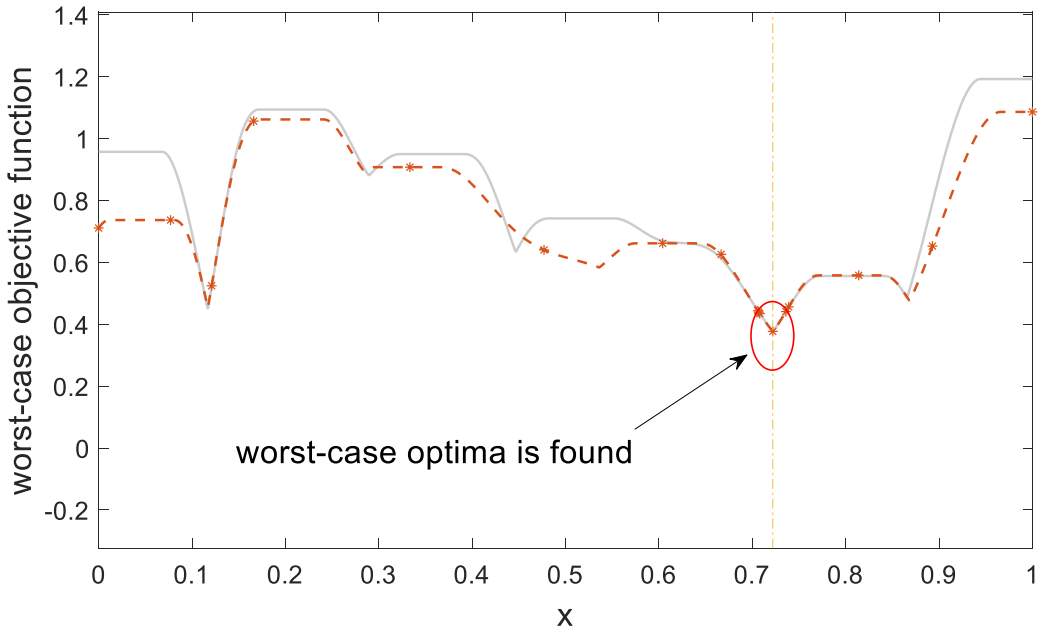
(b)


stage 2
objective function
maximum ED is less than 0.001
x

(c)


worst-case objective function
estimated worst-case optima
x

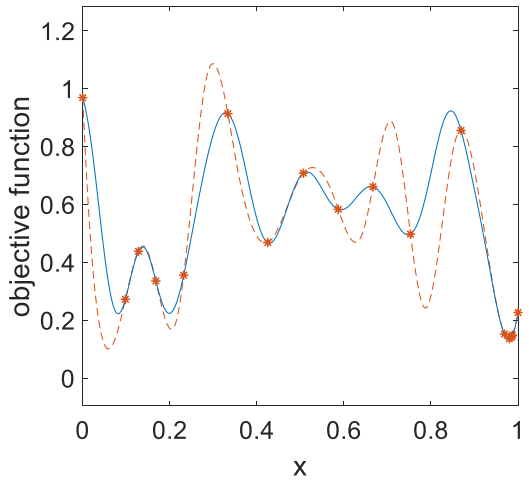
(d)



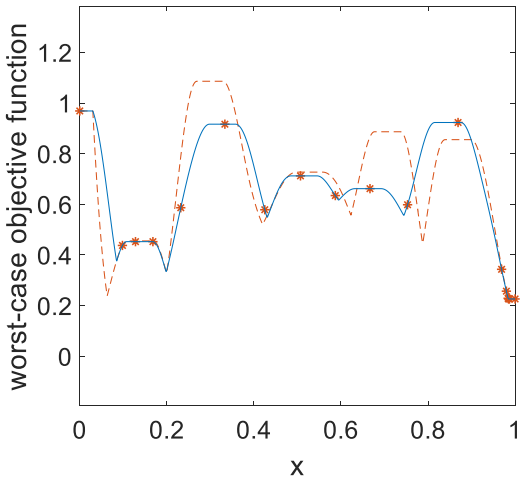
(e)

Figure 4.8. (a) Estimated worst-case model at the 12th iteration, (b) kriging model and ED infill sampling criterion at the 12th iteration, (c) maximum ED is less than 10^{-3} at the 15th iteration, (d) estimated worst-case model at the 15th iteration, (e) final estimated worst-case function and the located worst-case optimum.

As previously mentioned, the worst-case response surface of the original objective function may change dramatically from the original one. A number of additional results are provided by Figure 4.9.



(a)



(b)

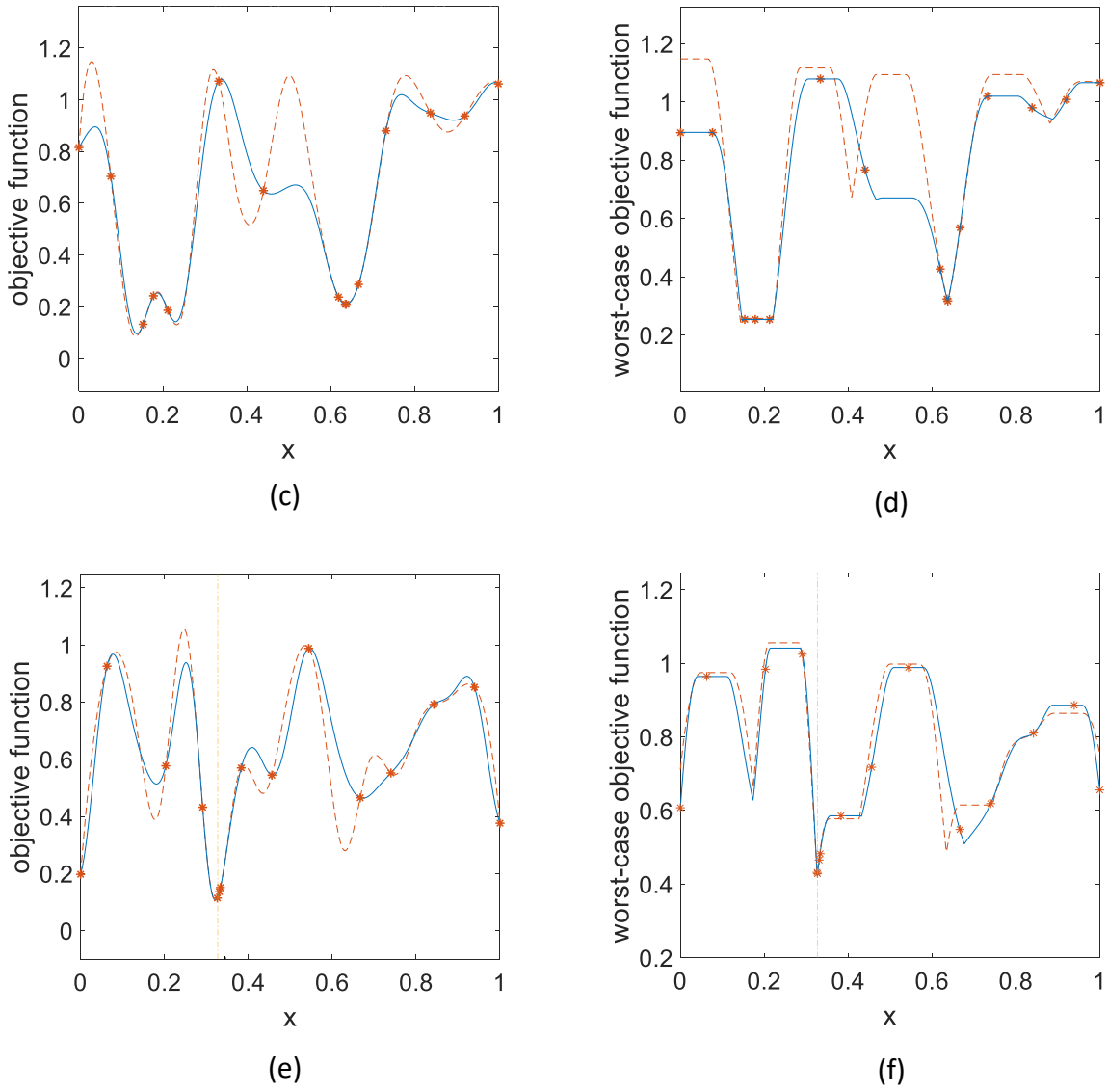


Figure 4.9. (a), (c), (e) kriging models and the original test function, (b),(d),(f) predicted robust models and robust version of the original function.

4.7 Solving practical problems

4.7.1 T.E.A.M. 22

T.E.A.M. workshop problems [40] consist of a set of practical electromagnetic optimisation design problems for benchmarking the performance of algorithms. A list of benchmark problems can be found on the International Compumag Society website. Each function evaluation requires the full finite element solution of a non-linear problem, which is computationally inefficient if directly used in combination with any optimisation method. Therefore, using a surrogate model based on optimisation techniques is preferred.

`

We tested the proposed approach on two practical benchmark problems. For both problems, the uncertainty boundary for each design parameter (upper and lower limit) was defined as 1% of their given design range.

The superconducting magnetic energy storage device in T.E.A.M. problem 22 [41] contains two superconducting coils; the design objective was to achieve a minimal stray field when the stored energy was equal to 180 MJ. The configuration of the inner coil is delineated in the 3 Parameter ("discrete") case, and therefore three parameters were optimised: namely, the radius R_2, height h_2 and thickness d_2 of the outer coil, as indicated in Figure 4.10.

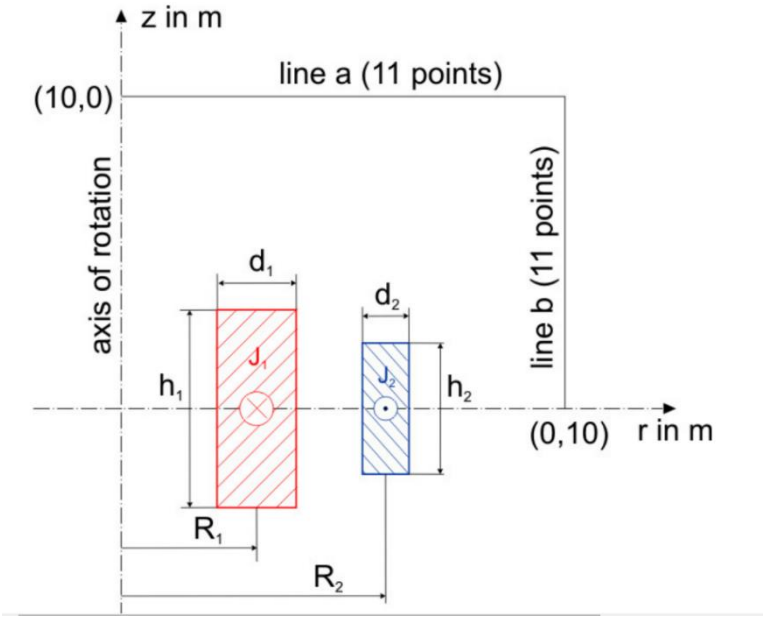


Figure 4.10. The superconducting magnetic energy storage device [40], [41].

The objective function is given as:

$$OF = \frac{B_{stray}^2}{B_{norm}^2} + \frac{|E - E_{ref}|}{E_{ref}} \tag{4.8}$$

where $E_{ref} = 180\ MJ$, $\boldsymbol{B_{norm}} = 3\ \mu T$ and B_{stray}^2 is defined as:

$$B_{stray}^2 = \frac{\sum_{i=1}^{n=22} |B_{stray,i}|}{22} \tag{4.9}$$

The magnetic field needs to meet certain physical conditions in order to guarantee superconductivity, as shown in Figure 4.11. This is known as the "quench condition". Quench condition is modelled by following inequality constraint:

$$|\boldsymbol{J}| + 6.4|\boldsymbol{B}| - 54.0 \leq 0 \tag{4.10}$$

where $\boldsymbol{J}$ is the current density and $\boldsymbol{B}$ is the maximum magnetic flux density, measured in A/mm^2 and T, respectively.

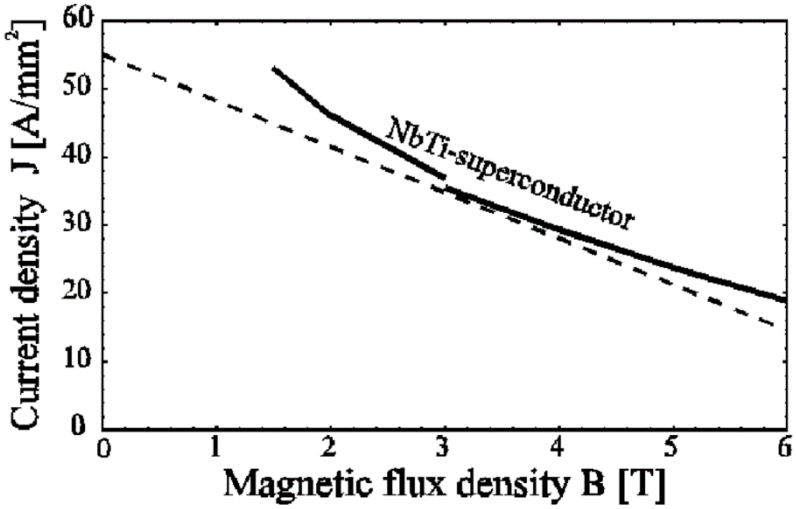


Figure 4.11. Critical curve of an industrial superconductor [40], [41].

Table 4.1 Comparison of performance of various optimisation methods (T.E.A.M. 22)

Algorithm	Optimum	Optimum (MagNet)	R_2	d_2	$h_2/2$	No of function calls
GA	0.134	0.1270	3.040	0.386	0.240	2400
SA	0.098	0.0916	3.078	0.390	0.237	5025
HuTS	0.089	0.1034	3.080	0.380	0.246	3821
NTS	0.089	0.1278	3.080	0.370	0.254	1800
PBIL	0.101	0.110	3.110	0.421	0.241	3278
Kriging EI	0.0875	0.0875	3.090	0.394	0.236	211
Kriging AWEI	0.0875	0.0875	3.090	0.400	0.232	323
Kriging WCEI (worst case)	0.1459	0.1459	3.021	0.391	0.250	277

Genetic algorithm GA [42]; Simulated Annealing SA [43]; Tabu Search HuTS [44]; Universal Tabu search [45]; New Tabu Search NTS [46]; Kriging EI [35]; Kriging AWEI [35]

Table 4.1 summarises the findings by citing the results from other publications and including the robust "worst-case" design. To compare and analyse the optimal value, an additional 'optimal value' has been obtained by implementing an identical finite-element model (FEM) setup (using MagNet software) for all the optimisation methods in the table. The outputs from the MagNet FEM are similar to the original values taken from the literature. The results show that kriging-assisted optimisation algorithms performed consistently well by achieving a marginally better solution with reduced effort (the number of necessary function calls reduced by almost an order of magnitude).

Not surprisingly, the worst-case optimal value is slightly larger than the non-robust optimum obtained by other methods in Table 4.1; this indicates that the response surface around the optimum is not flat, but nevertheless relatively shallow. The location

`

of the worst-optimum also shows a slight difference from other optima, indicating that the response surface around the optimum is not symmetrical about R_2, d_2 or h_2; however, a smaller value of R_2 indicates the response surface being shallower towards smaller R_2 for the area within the specified 1% uncertainty of parameters, while the opposite conclusion can be drawn for parameter h_2.

Table 4.2 Comparison of time cost between various optimisation methods

Algorithms	Time (s) per 10^3 iterations (algorithm)*	Estimated time (ms) per iteration (algorithm)	Time (s) per FEM call	Number of iterations	Estimated overall time (h)
GA	3.302	3.302	~22	2400	14.7
SA	2.197	2.197	~22	5025	30.7
HuTS	5.347	5.347	~22	3821	23.4
UTS	-	-	~22	1800	~11.0
NTS	-	-	~22	3278	~20.0
Kriging EI	52.04	52.04	~22	211	1.3
Kriging AWEI	59.12	59.12	~22	323	2.0
Kriging WCEI	726.39	726.39	~22	453	2.9

**Note: Experiments were carried out on a 3.4GHz PC. The MATLAB codes for Universal Tabu search UTS and New Tabu Search NTS were not available; the times for running non-surrogate model-based optimisation algorithms are very small (around 1000 times smaller) compared to the time for each FEM call, hence they can be neglected without affecting the comparison.*

Table 4.2 displays the estimated overall time cost for each algorithm of Table 4.1. The overall time cost is estimated based on the time cost per FEM call, number of FEM calls, and the time consumed by the optimisation algorithm. Each algorithm has been set up to solve a dummy test function (computation cost of the objective function can be neglected), the algorithm carried out 1,000 iterations for the objective problem and the time cost for each iteration was estimated.

It can be observed that the overall time cost of kriging-based approaches is much smaller than other direct search optimisation algorithms. This is mainly due to the high number of function calls used by these algorithms and the high computation cost of each objective function call. Since there is an extra layer of optimisation process in the proposed kriging-based approach for the worst-case optimisation, the time cost per iteration is much higher than for other regular kriging approaches. Nevertheless, when the objective function is time consuming to evaluate, kriging-based approaches have shown a great advantage in terms of time efficiency over other direct search optimisation methods.

`

4.7.2 T.E.A.M. 25

The device presented in Figure 4.12 is a die press with an electromagnet used to create a strong magnetic field to orient magnetic powder in a component to produce an anisotropic permanent magnet. Figure 4.12 illustrates a close-up picture of the device; the model in Figure 4.13 was built with a 2D finite element model using MagNet software. The objective of this problem is to optimise the shape of the two die moulds so that the objective function is minimised.

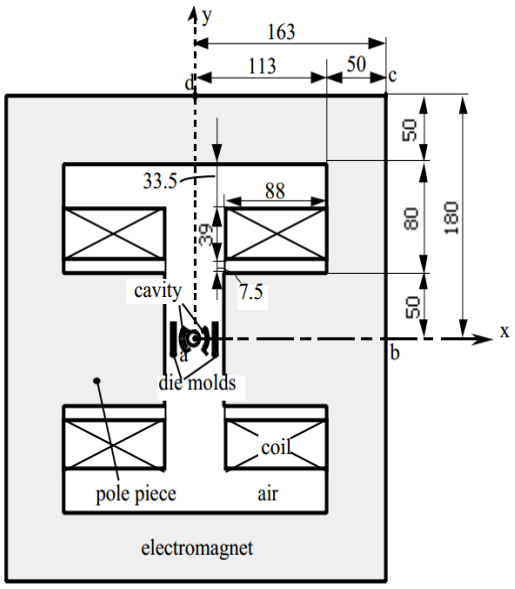


Figure 4.12. Device in T.E.A.M. problem 25. [37], [43].

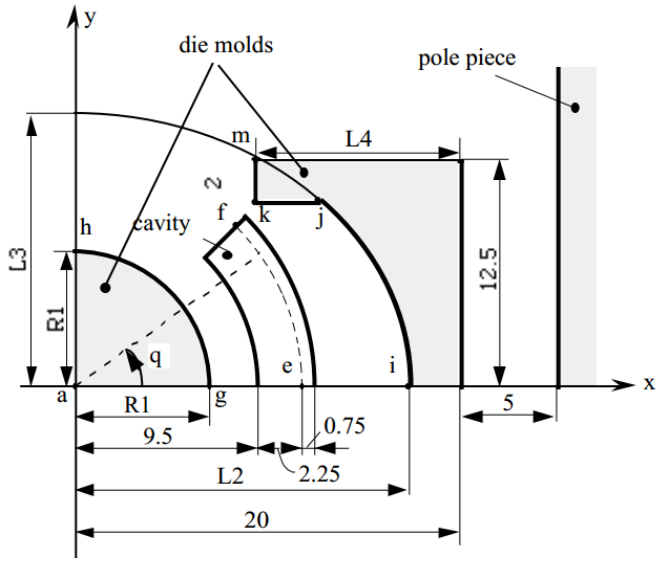


Figure 4.13. Detailed 2D view of the device.

Problem description:

In Figure 4.13, the shape g-h of the inner die mould is assumed to be a circle, while the inside shape i-j-m of the outer die mould is regarded as an ellipse and line j-k is parallel to the x-axis.

The ampere-turns of each coil is 4253AT.

The die press and electromagnet are made of steel and the B-H curve of the steel is given as:

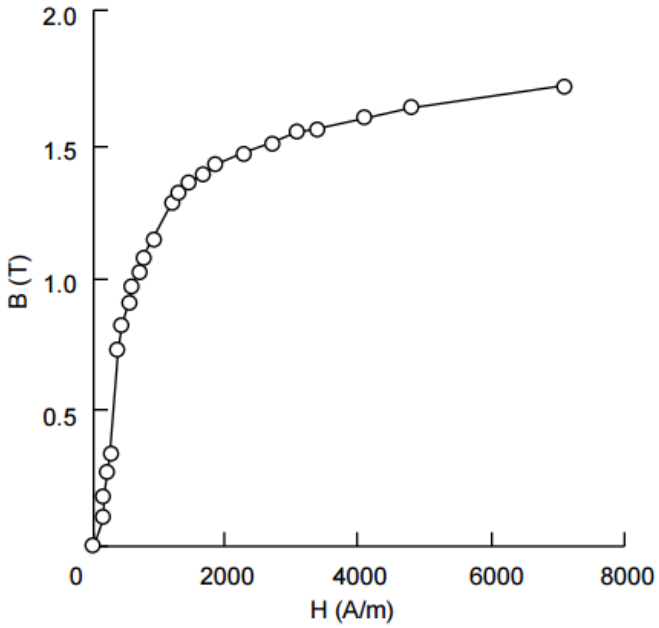


Figure 4.14. B/H curve of the steel.

Table 4.3 Typical B/H curve values

B(T)	H(A/m)	B(T)	H(A/m)
0	0	1.27	1164
0.11	140	1.32	1299
0.18	178	1.36	1462
0.28	215	1.39	1640
0.35	253	1.42	1851
0.74	391	1.47	2262
0.82	452	1.51	2685
0.91	529	1.54	3038
0.98	596	1.56	3395
1.02	677	1.6	4094
1.08	774	1.64	4756
1.15	902	1.72	7079

`

The magnetic flux density is given by:

$$\begin{cases} B_x = 0.35\cos\theta\ (T) \\ B_y = 0.35\sin\theta\ (T) \end{cases} \tag{4.11}$$

where B_x and B_y are the x and y components of flux density at a point along the e-f line at an angle of θ.

R1, L2, L3 and L4 are the design parameters to be optimised so that the objective function *W* is minimised. The objective function W is:

$$W = \sum_{i=1}^{n}\left\{\left(B_{xip} - B_{xio}\right)^2 + \left(B_{yip} - B_{yio}\right)^2\right\} \tag{4.12}$$

where B_x and B_y are the *x* and *y* components of magnetic flux density at points along the curve e-f, while subscripts *p* and *o* denote the calculated and desired values, respectively. The constraints are listed in Table 4.4 and the results in Table 4.5.

Table 4.4 Constraints of the parameters

Variable	Lower-boundary (mm)	Upper-boundary (mm)
R1	5	9.4
L2	12.6	18
L3	14	45
L4	4	19

Table 4.5 Comparison of performance of various optimisation methods (T.E.A.M. 25)

Algorithms	Optimum (10^{-4})	MagNet Optimum (10^{-4})	R1 (mm)	L2 (mm)	L3 (mm)	L4 (mm)	No of function calls
GA	2.686	1.3391	7.2996	14.174	14.001	14.326	3421
SA	1.622	2.0696	7.2252	14.322	14.110	14.306	2145
HuTS	0.500	1.3502	7.3780	14.613	14.371	14.204	1580
UTS	1.050	1.4647	7.5487	14.908	14.506	14.416	931
NTS	0.648	1.6907	7.4337	14.732	14.428	14.237	575
Kriging EI	0.452	0.4527	7.2	14.1	14	14.5	265
Kriging AWEI	0.412	0.4125	7.2	14	14	14.5	214
Dual kriging	0.323	0.3231	7.1	13.9	14.014	14.273	234
Kriging WCEI (worst case)	5.4442	5.4442	7.104	13.891	14.035	14.270	453

Genetic algorithm GA [42]; Simulated Annealing SA [43]; Tabu Search HuTS [44]; Universal Tabu search [45]; New Tabu Search NTS [46]; Kriging EI [35]; Kriging AWEI [35]; Dual kriging [47].

`

Similarly to the previous T.E.A.M. problem 22, an additional 'optimal value' has been obtained by implementing an identical finite-element model (FEM) setup (using MagNet software) for all the optimisation methods in the table. The main observations resulting from the T.E.A.M. 25 study is broadly in line with what was previously demonstrated, although – on this occasion – the worst-case optimum obtained by kriging WCEI in Table 4.5 is a magnitude larger compared to the non-robust optimum obtained by other kriging methods; this indicates that unlike in the previous case, the response surface around the optimum is much rougher and is likely to have steep hills around the optimum. The fact that the location of the worst-case optimum did not shift in a particular direction indicates that the response surface within the specified 1% uncertainty of parameters around the optimum location is equally rough and/or has steep hills. The higher number of FEM calls (452 FEM calls compared to around 230 FEM calls) shows that many more infill sampling points had to be added to the model before the final solution was obtained; part of the reason may be due to the rough response surface around the optimum location, hence a larger number of FEM calls during the validation process. However, it should be noted that different initial sampling locations alone can cause a different total number of required FEM calls at the end; moreover, the worst-case optimisation solver may use a different number of function calls to obtain the final solution or converge to a different local/global optimum, even when an identical FEM is implemented.

It is also interesting to note that for both T.E.A.M. problems, the originally published results (when the problems were first suggested) appear to be reasonably robust, more so than the subsequently offered solutions. Nevertheless, the most important conclusion from this study is that the kriging-assisted optimisation is reliable and offers superbly efficient computation, both for the "traditional" (global) optimisation and the robust formulation. Finally, the worst-case (minimax) approach appears to be a helpful methodology for robust optimisation.

Table 4.6. Comparison of time cost between various optimisation methods

Algorithms	Time (s) per 10^3 iterations (algorithm)*	Estimated time (ms) per iteration (algorithm)	Time (s) per FEM call	Number of iterations	Estimated overall time (h)
GA	3.302	3.302	~10	3421	9.5
SA	2.197	2.197	~10	2145	6
HuTS	5.347	5.347	~10	1580	4.4
UTS	-	-	~10	931	2.6
NTS	-	-	~10	575	1.6
Kriging EI	52.04	52.04	~10	265	0.7
Kriging AWEI	59.12	59.12	~10	214	0.6
Dual kriging	80.20	80.90	~10	234	0.7
Kriging WCEI	726.39	726.39	~10	453	1.3

**Note: As before, experiments were carried out on a 3.4GHz PC. The MATLAB codes for Universal Tabu search UTS and New Tabu Search NTS were not available; the times for running non-surrogate model-based optimisation algorithms are very small (around 1000 times smaller) compared to the time for each FEM call, hence they can be neglected without affecting the comparison.*

In a similar comparison as before, Table 4.6 displays the estimated overall time cost of each algorithm of Table 4.5. The overall time cost was estimated based on the time cost per FEM call, the number of FEM calls, and the time consumed by the optimisation algorithm. Each algorithm has been set up to solve a dummy test function (where the computation cost of the objective function can be neglected); the algorithm carried out 1,000 iterations for the objective problem and the time cost for each iteration was estimated.

Comparing with the previous T.E.A.M. problem 22, the time cost per FEM call is approximately halved for the T.E.A.M. 25 problem. Although the overall time savings are somewhat smaller than for the previous T.E.A.M. problem, the kriging-based methods still show a great advantage in time consuming design optimisation problems.

4.8 Conclusions

A two-stage approach to worst-case optimisation problems was proposed and details of the algorithm discussed. The suggested method does not compute the worst-case value nor the corresponding robustness measure for any design site during the model updating stage. This is to avoid the objective function evaluation at a location that would contribute less to the overall model landscape, which would have occurred if the worst-case value had been evaluated for the newly added infill point. Instead, the explicit search for the robust optimum occurs in the second stage after the model updating process has completed, including the addition of a validation process to exploit the region around the estimated worst-case optimum. A more efficient infill criterion selection algorithm was introduced. The proposed optimisation method was validated using simple test functions and two multi-dimensional practical electromagnetic design problems, T.E.A.M. 22 and T.E.A.M. 25. The test results indicate that, with the aid of kriging surrogate modelling techniques, the proposed methodology significantly reduces the number of FEM function calls compared to other methods and thus, is computationally significantly efficient for both global and robust optimisation.

`

Chapter 5. Kriging for larger datasets

5.1 Introduction

Surrogate modelling techniques are helpful tools in design optimisation, especially when the underlying problem is computationally expensive. This situation frequently arises in the design of electromagnetic devices, where time-consuming finite element simulations may be necessary to ensure accurate performance prediction [7]. Kriging-based methodologies have been shown to be particularly useful and accurate when estimating the underlying problem while reducing the number of required objective function calls. Unfortunately, the complexity of the algorithm increases, since solving the kriging model involves the inversion of a correlation matrix, which results in $O(n^3)$ computation cost and $O(n^2)$ storage cost[48]. Hence, depending on the computer's hardware, there is a limitation on the maximum number of design points that a conventional kriging model may be built upon [49][50]. The exponential growth of the correlation matrix size led to the well-known bottle-neck of the kriging method. Thus, efficient application of kriging is often limited to smaller scale design problems.

Much effort has been devoted to addressing this bottlenecking phenomenon that occurs when kriging methods are applied to large datasets. A majority of methods approximate the original matrix to handle large datasets, at the expense of accuracy. Approximation shares the same ideals as surrogate modelling itself; a kriging model is built to approximate a more computational expensive problem at the expense of the original problem's accuracy. Some examples include: zooming-in modelling [7], moving-window kriging [51], covariance tapering [52] and fixed rank kriging [49]. Although these four methods provide valuable works on reducing the size of covariance matrixes, there are a number of limitations. For example, moving window Kriging[51] is not suitable for infill sampling-based optimisation and a popular EI criterion requires an MSE at unknown locations to be calculated to generate the next infill point. The moving window approach would either cause non-continuous MSE values at the boundaries of windows or generate different MSE values for the same unknown location due to different kriging models. The covariance matrix tapering approach relies on the sparseness of covariance matrixes. In MATLAB, the storage requirement for a sparse matrix with 50% zero elements and a full matrix is approximately the same; in other words, the percentage of zero elements in a sparse matrix needs to be greater than 50% for it to have an advantage in storage space over a conventional full matrix. This significantly limits the effectiveness of the covariance matrix tapering approach. Fixed rank kriging is designed to reduce the computational burden of massive datasets instead of addressing the memory problem.

`

In this chapter, we present two independent methods that address the memory problem in covariance matrices. The first approach is dual kriging, which builds two kriging models on two sets of data and separates the data using a critical points selection mechanism. The second approach involves model centring and points aggregation. Both methods enable the memory usage of the covariance matrix to be fixed at a constant level at the expense of information.

5.2 Dual kriging

The dual Kriging approach proposed in this chapter is based on the idea that as optimisation progresses, additional design points are added to the surrogate model; once the surrogate model (of the entire design space or an area within the design space) is reasonably accurate, only some of the sampling points are needed, especially those close to the areas considered potentially attractive. Thus, as the total number of sampling points increases and the computer's memory limit is neared, we may instead remove some of the less attractive points from the current kriging model to keep the total number of points constant; the removed points may be used to create a secondary kriging model.

Consider the kriging model in Figure 5.1 below, where the grey line shows the objective function and the blue dashed line shows the kriging model with 20 sampling points. Based on the ideas discussed in the previous paragraph, some points can potentially be removed, as illustrated in the Figure.

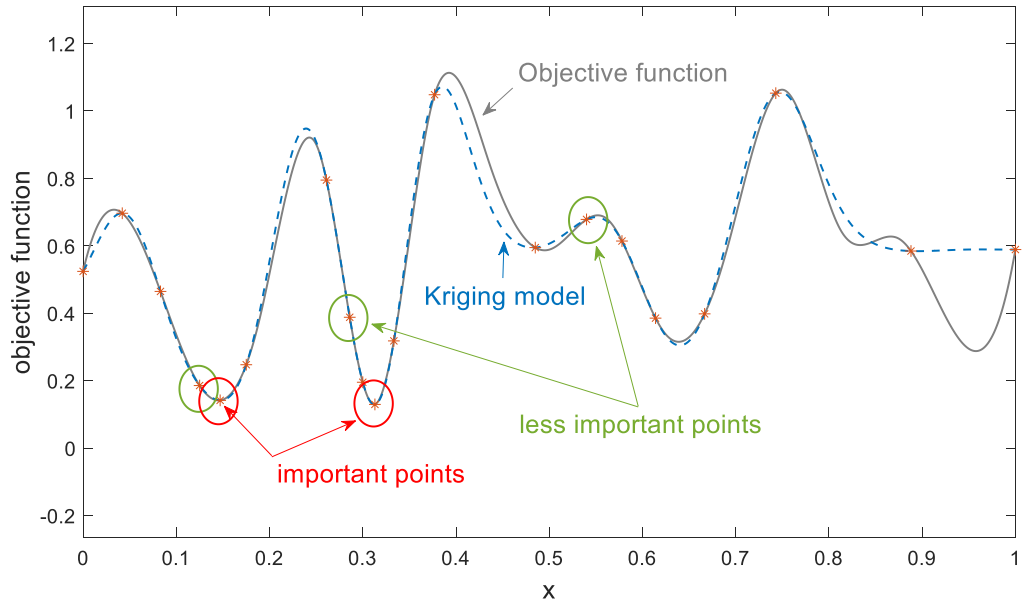


Figure 5.1. Ordinary kriging model with 20 sampling points.

5.2.1 Points removal

Design points that are located in a flat region away from the current global optimal or have a similar objective function value while geographically close to other known design points are considered to be less important. For each known design point, its nearest $n + 1$ number of neighbour points is found, where n equals the dimension of the problem, then weight the function value at each neighbour point, where the weight is inversely proportional to the Euclidean distance of the point and that neighbour. The weights are scaled so that they sum to 1. For the i^{th} existing point in the design space, the weight of its k^{th} nearest neighbour is given by:

$$weight_k(i) = \frac{1}{d_k} \cdot \left(\sum_{j=1}^{n+1} \frac{1}{d_j} \right)^{-1} \tag{5.1}$$

where d_k is the distance between the i^{th} design point and its k^{th} nearest neighbour. The weighted value $Y_{weighted}$ of all nearest neighbours at the location of the target point is therefore:

$$Y_{weighted}(i) = \sum_{j=1}^{n+1} \left(weight_j(i) \cdot N(j) \right) \tag{5.2}$$

where N is the function values of the neighbours of the i^{th} existing point and the removal criterion C is given by:

$$C(i) = \left| Y_{weighted}(i) - Y(i) \right| \cdot \sum_{j=1}^{n+1} d_j \tag{5.3}$$

where $Y(i)$ is the function value at the i^{th} existing point.

5.2.2 Points allocation

Figure 5.2 is the plot of the memory savings curve against the preserved sampling points in the main Kriging, both in terms of percentage. There exists a trade-off between the gain of memory savings for each design point removed and the total percentage of memory savings. The gain of memory savings is largest at the left end of the curve (where its differentiation is largest), while the total amount of memory savings is largest at the right end of the curve 9where its differentiation is smallest). Thus, as more points are removed from the main kriging model, the memory savings decreases.

`

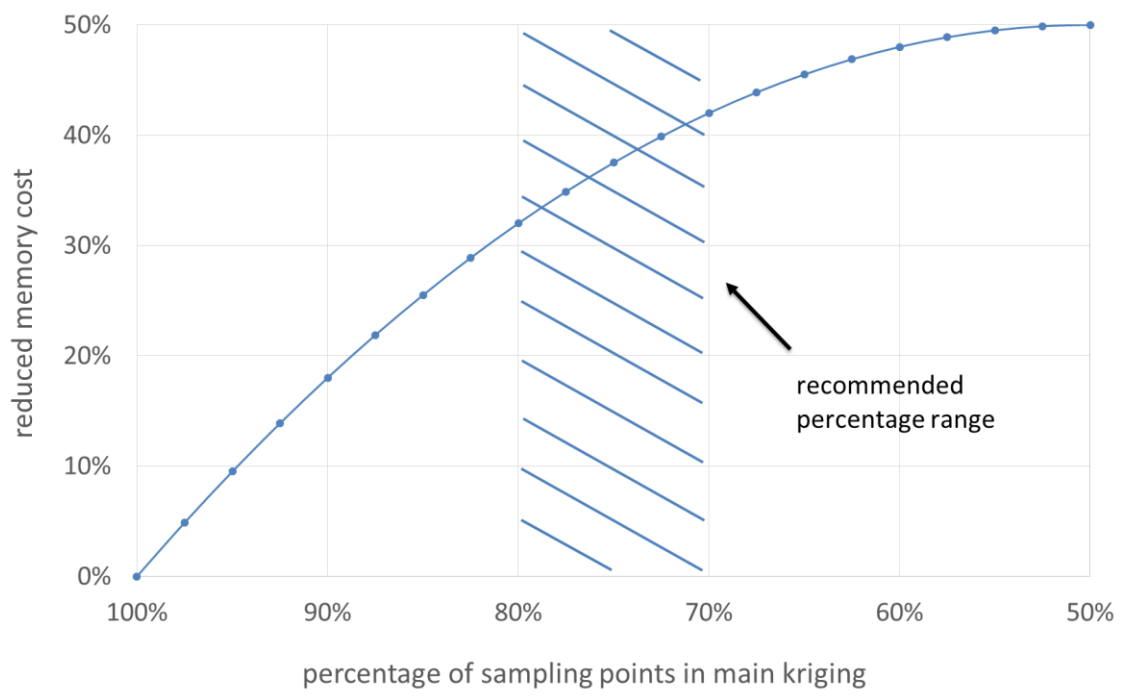


Figure 5.2. Efficiency of sampling points allocation.

The overall reduced memory cost Y can be written as:

$$Y = 1 - \frac{m^2 + n^2}{N^2} \tag{5.4}$$

The percentage of sampling points in the main kriging is:

$$X = \frac{m}{N} \tag{5.5}$$

where m and n are the number of sampling points in the main and secondary kriging; N is the total number of sampling points and $N = m + n$.

Therefore, the relationship between the overall memory savings and the percentage of sampling points in Figure 5.1 can be calculated as:

$$Y = 2X(1 - X) \tag{5.6}$$

Differentiating Y with respect to X, we obtain the point's removal efficiency:

$$\frac{dY}{dX} = 2(1 - 2X) \tag{5.7}$$

For $0 \leq X \leq 0.5$, the gain of memory savings for each design point removed is at its maximum when $X = 0$ and minimum when $X = 0.5$.

To make the overall memory savings meaningful while retaining a relatively high gain of memory savings for each design point removed, we can choose to operate between 20% and 30% of the reduced number of points (shaded area). For example, at 20% (main

`

Kriging model preserving 80% of points) and 30% (main Kriging model preserving 70% of points), the memory saving is 32% and 42%, respectively.

While applying dual kriging could potentially save computer memory demands by as much as 50% compared to a single kriging approach for the same number of sampling points, the major challenge of the dual kriging approach lies in the selection of the new sampling points. If only the main kriging model were to be considered, there is a risk of a new location to be at or close to a point which has just been removed. Thus, to inhibit this scenario, the location of the removed points must be recorded and taken into consideration during the infill point selection process. There are various possible ways to achieve this goal, such as using MSEs from both kriging models so that a modified MSE is used to replace the MSE calculated from a single model. The modified MSE is the product of the MSEs from the main and secondary kriging models. By using estimated error from both kriging models, information from previously sampled points is used efficiently.

5.2.3 One-dimensional example

The proposed dual kriging approach is illustrated by a 1D example in this section, the test function is given as follows:

$$f(x) = a \cdot cos\big(w(x-p)\big)e^{-\left(w(x-p)\right)^m} + b \cdot sin\left(e^{-\left(v(x-q)\right)^k}\right) \qquad (5.8)$$

$$+c \cdot cos\big(u(x-r)\big)e^{-\left(u(x-r)\right)^l} + f_{wave}$$

where $a = -3.5$, $w = 1$, $p = 0.1$, $m = 2$, $b = -1.8$, $v = 0.2$, $q_1 = 0.75$, $k = 6$, $c = -2$, $u = 0.3$, $q_2 = 0.45$, $l = 4$, and f_{wave} is an interpolation function of a set of randomly generated points.

The maximum number of sampling points is limited to 20. The second kriging starts when the number of existing points reaches 10 points, which equals half of the maximum number of sampling points allowed.

The process is illustrated in Figures 5.3-5.9. The solid grey line is the underlying test function and the orange dashed line and points are the main kriging model and associated design points, respectively; the green dashed line and points are the secondary kriging model and associated design points, respectively; the solid black line is the infill criterion based on modified MSEs from both kriging models and the dashed vertical line represents the next infill location. As the optimisation progresses, the number of design points in the main kriging is maintained at 10, and the least important point is reallocated to the secondary kriging; at the 20th iteration each kriging model consists of 10 points, so that the covariance matrix size is reduced by 50 percent.

`

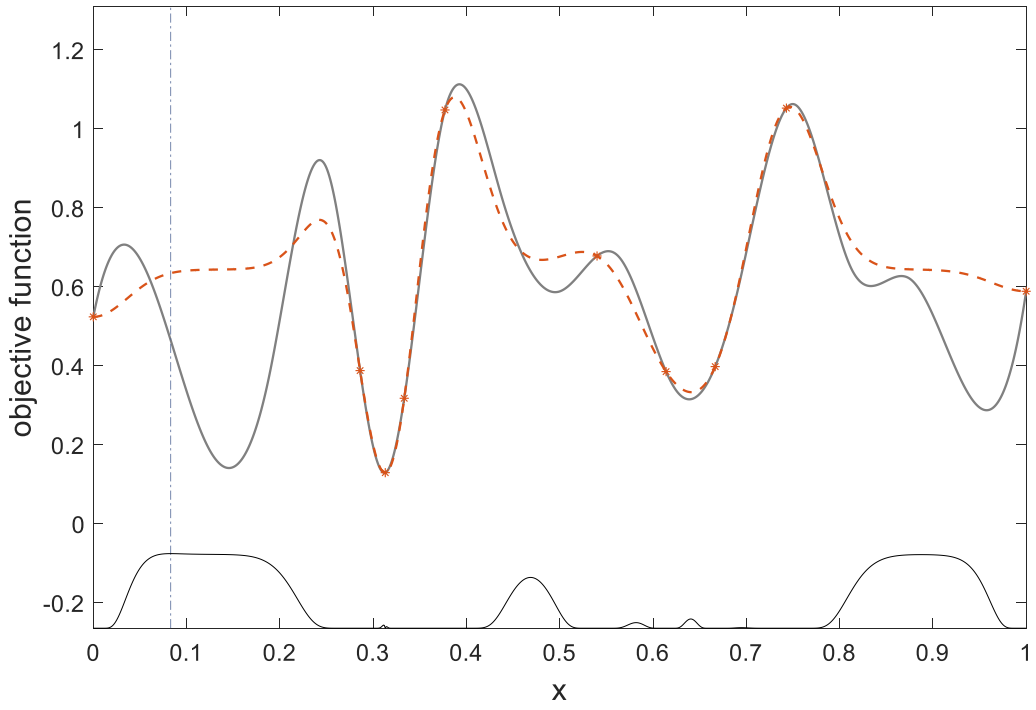


Figure 5.3. Dual kriging at the 10th iteration, no point has been removed yet.

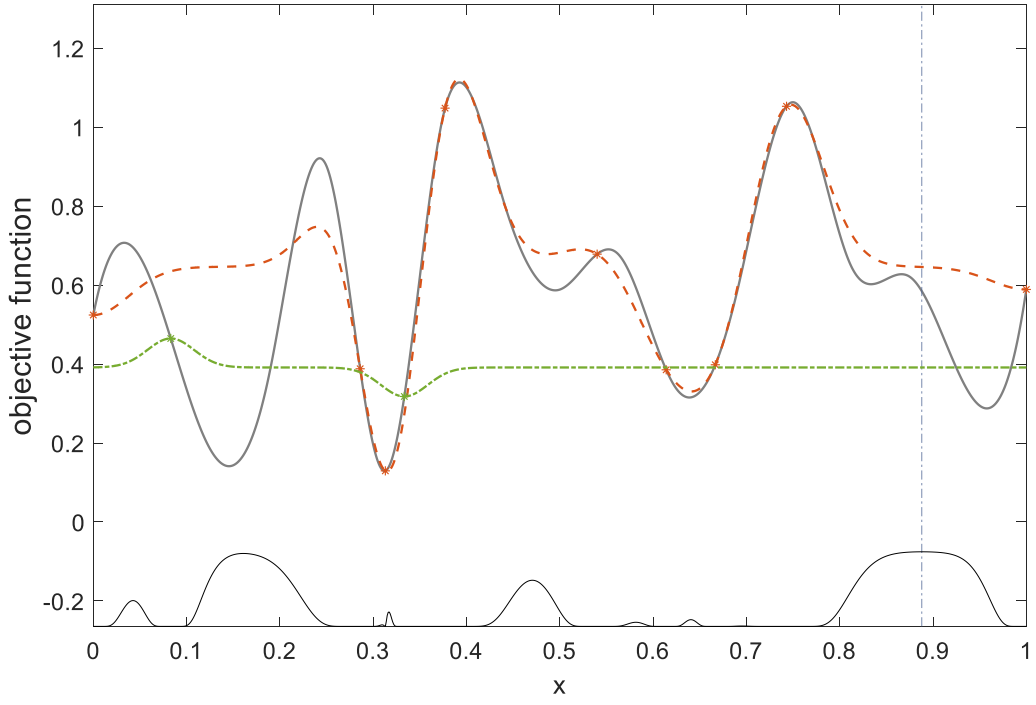


Figure 5.4. Dual kriging after the 12th iteration; two points have been removed from the main kriging.

`

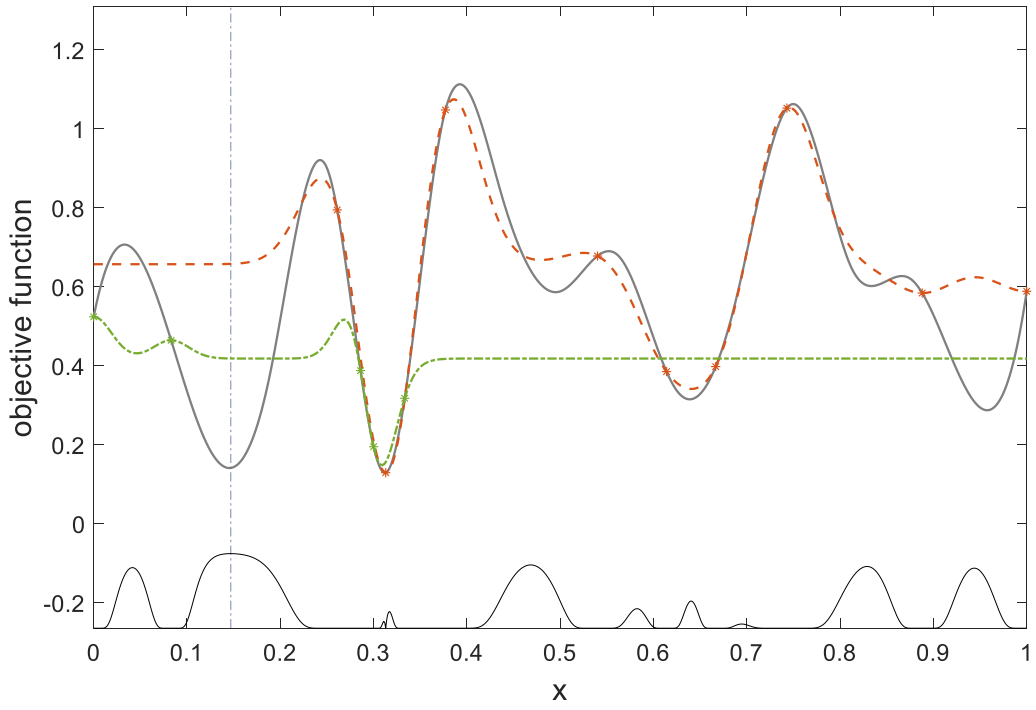


Figure 5.5. Dual kriging after the 14th iteration, four points have been removed from the main kriging.

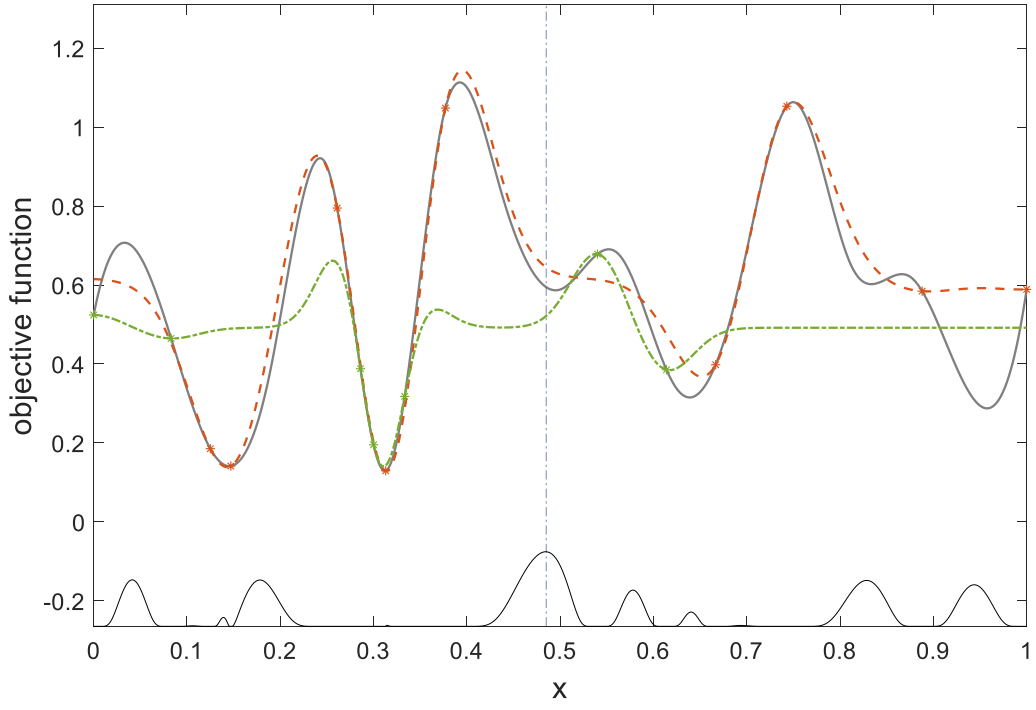


Figure 5.6. Dual kriging after the 16th iteration, six points have been removed from the main kriging.

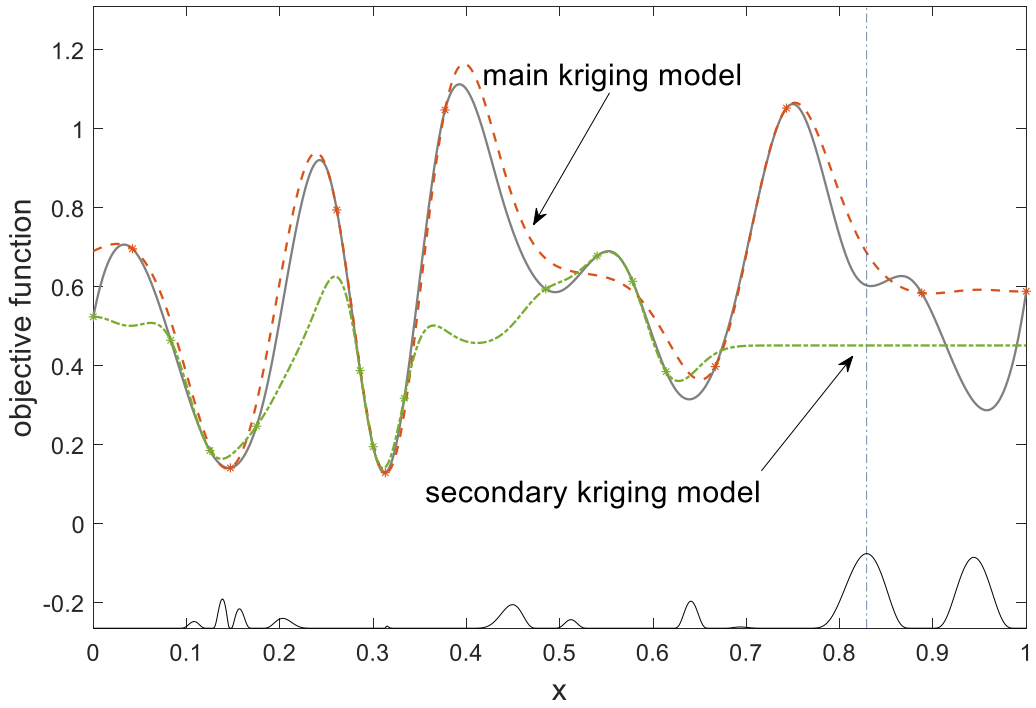


Figure 5.7. Dual kriging after the 20th iteration, ten points have been removed from the main kriging.

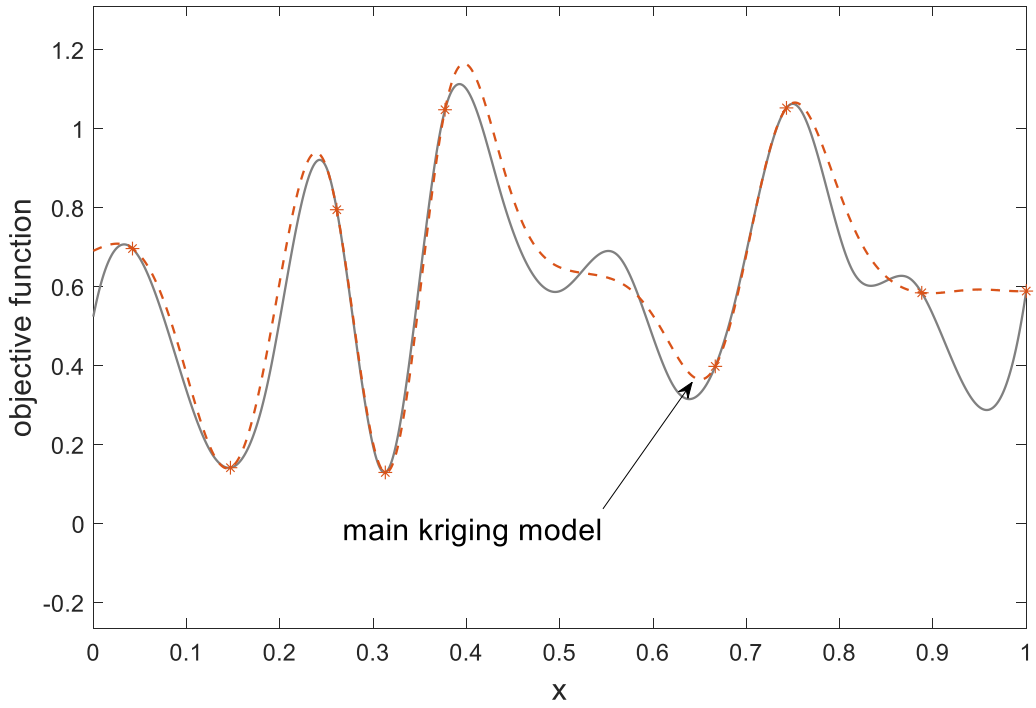


Figure 5.8. Main kriging at the 20th iteration (10 points).

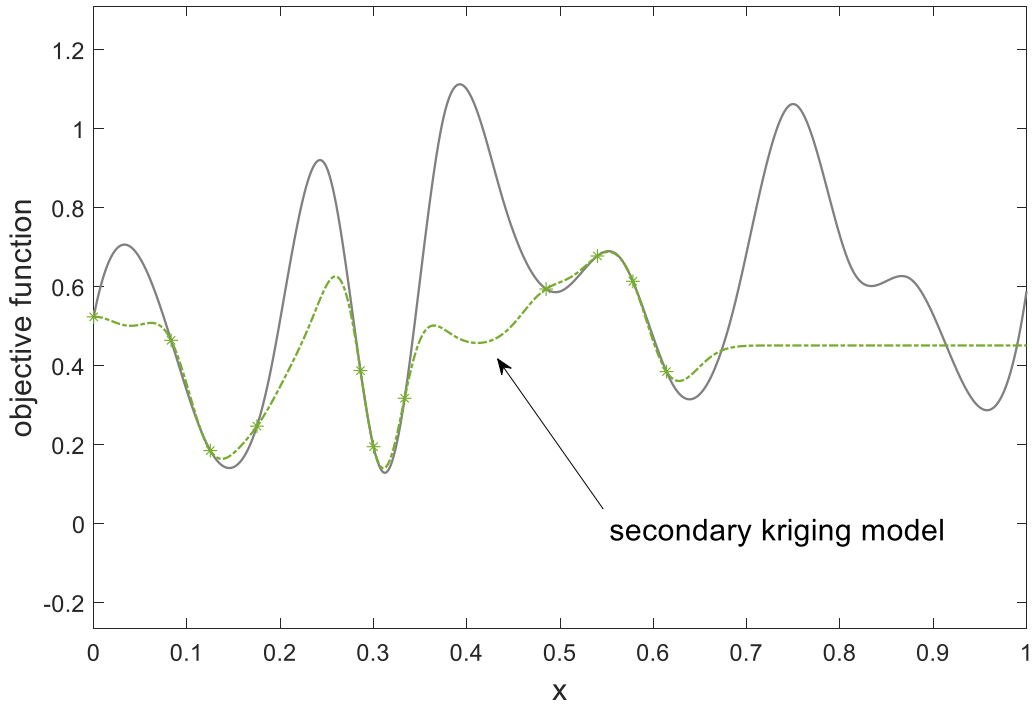


Figure 5.9. Secondary kriging at the 20th iteration (10 points).

5.2.4 Solving a practical problem

To illustrate the proposed optimisation methodology within the context of electromagnetic design, T.E.A.M. problem 25 was studied [11], which is a die press with an electromagnet for the orientation of magnetic powder (this is used to produce an anisotropic permanent magnet). Problem details are provided in section 4.7.2.

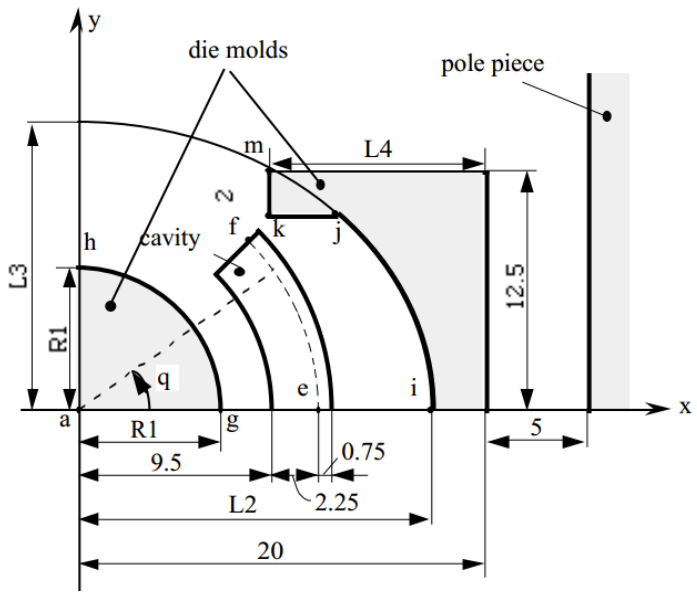


Figure 5.10. Detailed 2D view of the device.

Table 5.1. Comparison of performance of various optimisation methods (T.E.A.M. 25)

Algorithms	Optimum (10^{-4})	MagNet Optimum (10^{-4})	R1 (mm)	L2 (mm)	L3 (mm)	L4 (mm)	No of function calls
GA	2.686	1.3391	7.2996	14.174	14.001	14.326	3421
SA	1.622	2.0696	7.2252	14.322	14.110	14.306	2145
HuTS	0.500	1.3502	7.3780	14.613	14.371	14.204	1580
UTS	1.050	1.4647	7.5487	14.908	14.506	14.416	931
NTS	0.648	1.6907	7.4337	14.732	14.428	14.237	575
Kriging EI	0.452	0.4527	7.2	14.1	14	14.5	265
Kriging AWEI	0.412	0.4125	7.2	14	14	14.5	214
Dual kriging	0.323	0.3231	7.007	13.891	14.035	14.270	242

Genetic algorithm GA [42]; Simulated Annealing SA [30]; Tabu Search HuTS [44]; Universal Tabu search [45]; New Tabu Search NTS [46]; Kriging EI and Kriging AWEI (with specified step size) [35];

As shown in Table 5.1 are additional 'optimal values' obtained by implementing an identical finite-element model (FEM) setup (using MagNet software) for all the optimisation methods in the table.

The optimum location provided by the dual kriging approach is slightly different to those obtained by other optimisation methods in Table 5.1, while the optimal value is slightly improved. Parameters R1 and L2 are smaller than the ones found by other methods, L3 and L4 is comparatively smaller too. This may be explained by looking at the results in Table 4.4 in Section 4.7.2 in Chapter 4; as the response surface around the optimum is comparatively rough, the function values may have differed by more than one

magnitude even with only a slight change in parameters (±1% uncertainty in parameters). Therefore, there exist a possibility that models used by other authors were less accurate (since the results were published much earlier). Another possible explanation is that due to the rough respond surface around the optimum location, the algorithm may have converged to different mini-local optima within that area.

Table 5.2. Comparison of time cost between various optimisation methods

Algorithms	Time (s) per 10^3 iterations (algorithm)*	Estimated time (ms) per iteration (algorithm)	Time (s) per FEM call	Number of iterations	Estimated overall time (h)
GA	3.302	3.302	~10	3421	9.5
SA	2.197	2.197	~10	2145	6
HuTS	5.347	5.347	~10	1580	4.4
UTS	-	-	~10	931	2.6
NTS	-	-	~10	575	1.6
Kriging EI	52.04	52.04	~10	265	0.7
Kriging AWEI	59.12	59.12	~10	214	0.6
Dual kriging	80.20	80.90	~10	234	0.7

**Note: Again, the experiments were carried out on a 3.4GHz PC. The MATLAB codes for Universal Tabu search UTS and New Tabu Search NTS were not available, hence the time cost for running the algorithms were neglected. Nevertheless, the time cost for running non-surrogate model-based optimisation algorithms are comparably small; the time cost for each FEM call is more than 1000 times higher than the time consumed by the algorithm, hence it can be neglected without affecting the comparison.*

Again, in a similar comparison as before, Table 5.2 displays the estimated overall time cost of each algorithm of Table 5.1. The overall time cost was estimated based on the time cost per FEM call, the number of FEM calls, and the time consumed by the optimisation algorithm. Each algorithm has been set up to solve a dummy test function (where the computation cost of the objective function can be neglected); the algorithm carried out 1,000 iterations for the objective problem and the time cost for each iteration was estimated.

The overall time cost of dual kriging and other kriging-based approaches is much smaller than other direct search optimisation algorithms as shown in the table, mainly due to the high number of function calls used by these algorithms and the high computation cost of each objective function call. In general, the kriging performed significantly better than other methods in terms of a better optimum. However, this is primarily due to the reduced computing times (measured in the number of necessary FE calculations). The dual kriging required marginally more iterations, but produced a slightly better result. In this sense, all kriging models are similar and superior to other methods.

`

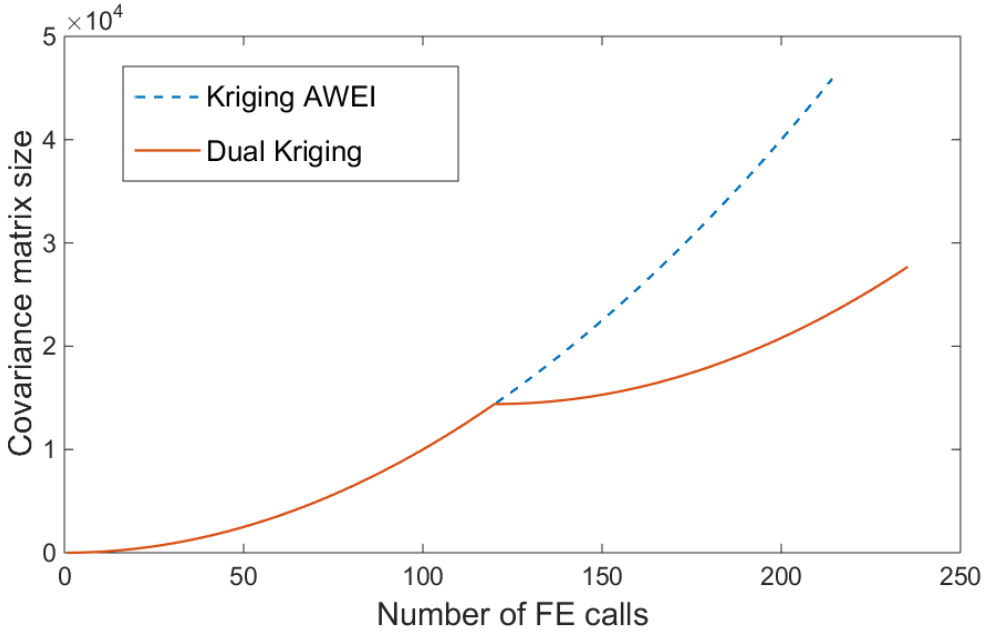


Figure 5.11. Memory requirements for the covariance matrix at each iteration.

The dual kriging algorithm was triggered after 120 FE calls, with one sampling point removed when a new one was added in subsequent iterations. The sizes of the covariance matrix in the kriging model for the "standard" AWEI [35] routine and the new dual algorithm are compared in Figure 5.11, as optimisation progresses. Memory savings on this occasion reached 36%.

5.3 Kriging with points aggregation

In this section, the kriging with points aggregation is proposed. This method involves locating the most interesting search area for the next infill point and then aggregating the points outside this centre area. Finally, a kriging model is built for the infill point search within the identified centre area.

5.3.1 Centre positioning

The objective of this step is to find a region within the search space that is most worth exploiting in the next iteration. Points located inside this region will be preserved, whereas points far from this region may be aggregated based on their distance to the region itself and the next infill sampling point will be placed within this region.

To determine which region is more interesting, the following three factors are considered:

1. Sample rate within that region;
2. Standard deviation of the function value;
3. Mean of the function value.

`

These three measures are defined by C_1, C_2 and C_3, respectively:

$$C_1 = {R_{C_1}}^{-1} \times \sum_{i=1}^{k} \|\boldsymbol{x_i} - \boldsymbol{c}\|^{\frac{1}{2}}, \quad R_{C_1} = k \times \max\left\{\|\boldsymbol{x_i} - \boldsymbol{c}\|^{\frac{1}{2}}\right\} \tag{5.12}$$

$$C_2 = {R_{C_2}}^{-1} \times \sqrt{\frac{\sum_{i=1}^{k}(y_i - \mu)^2}{k}}, \quad R_{C_2} = \frac{range(\boldsymbol{y})}{2} \tag{5.13}$$

$$C_3 = {R_{C_3}}^{-1} \times \left(max(\boldsymbol{Y}) - \frac{\sum_{i=1}^{k} -w_i y_i}{\sum_{i=1}^{k} w_i}\right), \quad R_{C_3} = range(\boldsymbol{Y}), \tag{5.14}$$

$$w = e^{-v^{-5}(\boldsymbol{x_i} - \boldsymbol{c})} \tag{5.15}$$

where $\boldsymbol{x}_i$ is the location of the i^{th} known point, $\boldsymbol{c}$ denotes the centre, k defines the number of closest neighbourhood points around $\boldsymbol{c}$, y_i denotes the objective function values of the i^{th} closest neighborhood points, w_i is the weight term which has an inverse relationship with the distance from point i to the centre c, μ is the mean of y and v is the calculated probability.

C_1 in (1) is the sum of the square roots of the Euclidean distances between the hypothetical centre $\boldsymbol{c}$ and *k* nearest points around it. The value of C_1 is a measure of a sample rate within the region; it determines how close a hypothetical centre $\boldsymbol{c}$ is located in relation to its nearest *k* points, while the square root de-emphasises the importance of remote points.

C_2 is the weighted standard deviation of the objective function values for all the neighbourhood points.

Finally, C_3 is the weighted mean of the objective function values for all the neighbourhood points. Each point is weighted by an exponential function w, whose gradient is controlled by the parameter *v.* The smaller values of *v* apply less weight on remote points.

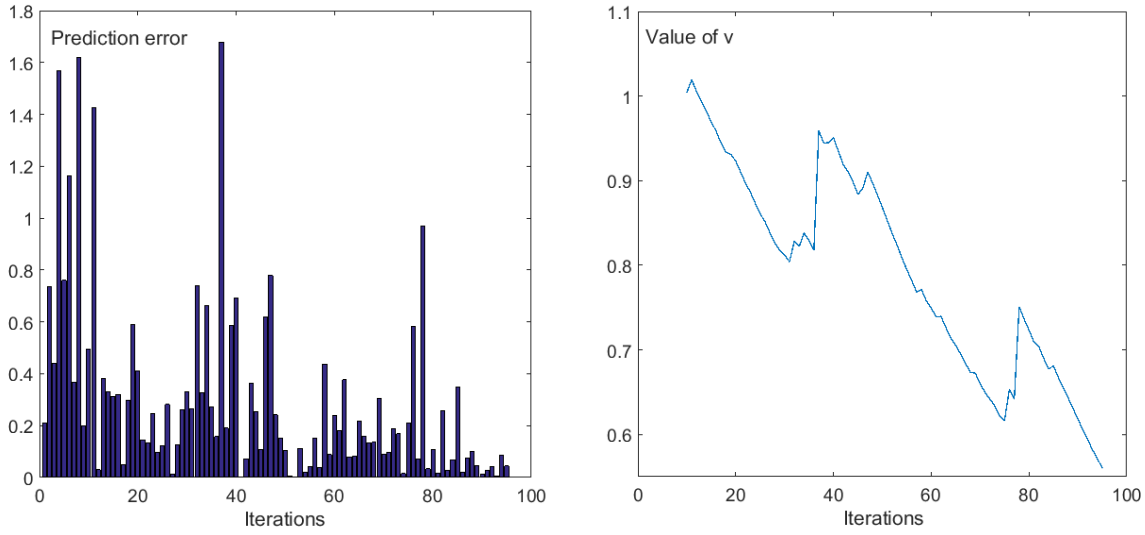
Both C_1 and C_2 terms encourage exploration of the under-sampled and rough areas, respectively, while C_3 focuses on the exploitation of the current optimum region. The final criterion for the model centre is a combination of the exploration term and the exploitation term:

$$Criterion_c(x) = \begin{cases} C_1 + C_2, & random(0,1) < v \\ C_3, & otherwise \end{cases} \tag{5.16}$$

The probability of exploration and exploitation is controlled by a parameter v, whose value is related to the root-mean-square deviation (RMSD) of the kriging model. Instead of a deterministic mixture of exploration and exploitation terms, a stochastic approach was applied to search for the objective at different stages while eliminating the risk of the deterministic criterion function being trapped in a local optimum.

`

The parameter v is the ratio of the exponentially weighted RMSD and regular RMSD of historical prediction errors, a detailed study of which is presented in Chapter 3. The proposed approach automatically balances the exploration and the exploitation based on the kriging model quality in section 3.3.2 and the parameter v is a suitable measure to switch the criterion between exploration and exploitation based on the model quality. In Figure 5.12, when more infill sampling points are added to the model, v decreases as the RMSD decreases. The rate of decrement can be controlled by α, the details of which are also given in section 3.3.2.



(a) (b)

Figure 5.12. (a) The history of the prediction error, (b) the history of the value of v, as iterations progress.

For an arbitrary location x inside the design space, a corresponding $Criterion_c$ can be calculated. Subsequently, the location with the maximum $Criterion_c$ is defined as the model centre.

Regardless of which criterion function is used, this optimisation stage involves finding the number of closest neighbourhood points k around $\boldsymbol{c}$ and calculating the weighting term v, which requires extra computational resources. Therefore, finding the hypothetical centre $\boldsymbol{c}$ using exhaustive search is not practical. This task may be seen as a global optimisation problem with the input $\boldsymbol{c}$, the output $Criterion_c$ and the objective function $Criterion_c = (C_1 + C_2)$ or $Criterion_c = C_3$. Because the centre only defines an area for the new infill point, an approximate solution will suffice at this step. Thus, a stochastic sequential global optimisation method simulated annealing is recommended for this task. The algorithm is simple, fast and its parameter is relatively easy to control. It may be argued that since the precision of estimating the intermediate optimum is not that important, a sequential method will have the advantage of fewer function calls and more flexibility over population-based methods.

The response surface of a 2D function is plotted in Figure 5.13, with the red crosses at the bottom marking the location of existing design points. Figures 5.14 (a) and (b) illustrate the criterion function for exploration and exploitation terms at the 90th iteration, respectively. As can be seen from the figures, the C1+C2 term encourages search in less sampled and non-smooth areas around *x*=[0.92,0.79], while the C_3 function suggests exploration of the area around the minimum at *x*=[0.20 0.27]. The test function is given as follows:

$$f(x_j) = c + a\prod_{j=1}^{j=2} cos\left(w(x_j - p_j)\right) e^{-\left(w(x_j-p_j)\right)^m} \qquad (5.17)$$

$$+b\prod_{j=1}^{j=2} sin\left(e^{-\left(v(x_j-q_j)\right)^k}\right) + f_{wave}$$

where $c = 4.5$, $a = 3.5$, $w = 8.4$, $p_1 = 0.2$, $p_2 = 0.3$, $m = 2$, $b = 2.8$, $v = 6.4$, $q_1 = 6.5$, $q_2 = 8$, $k = 6$, and f_{wave} is an interpolation function of a set of randomly generated points.

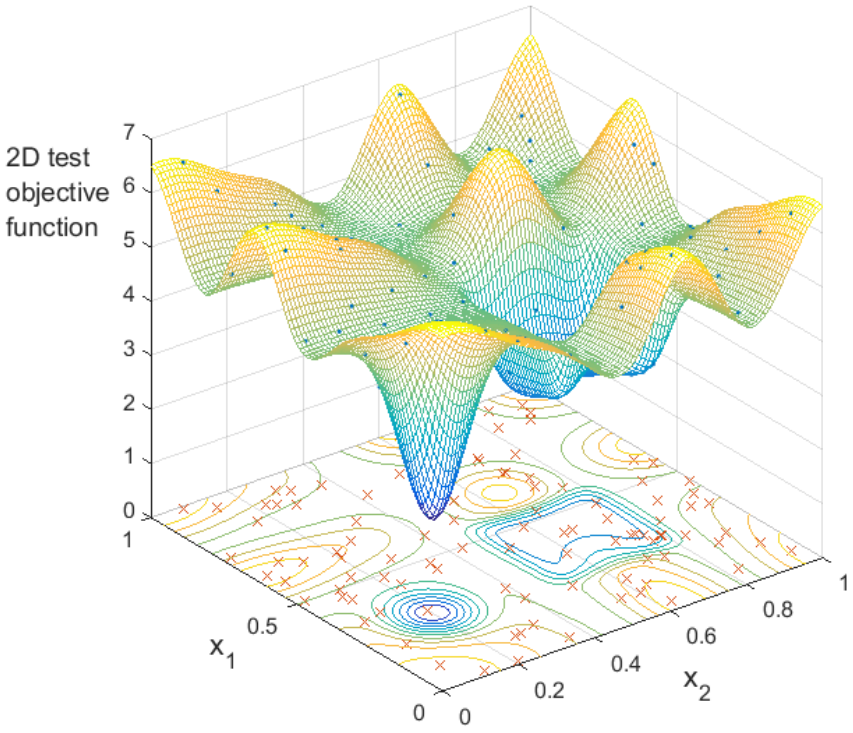
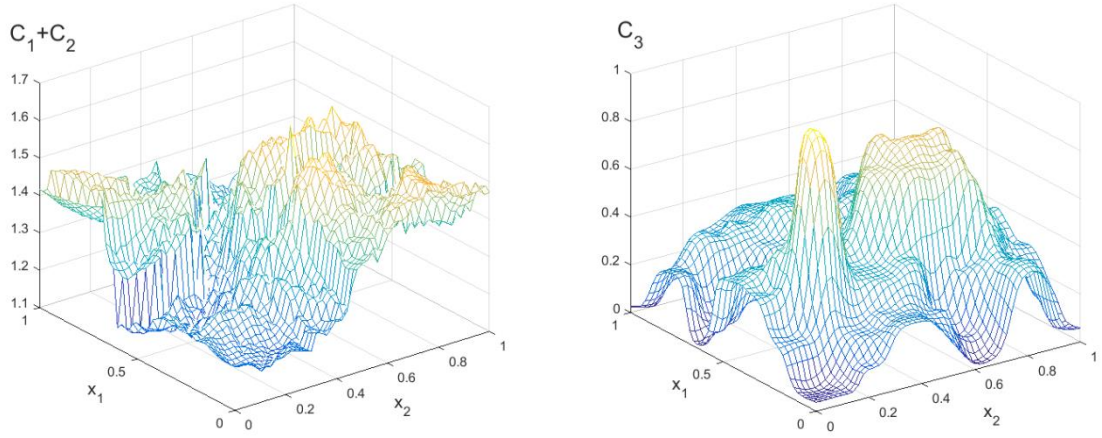


Figure 5.13. A 2D test function and existing design points.



(a) (b)

Figure 5.14. (a) Exploration functions $C_1 + C_2$, (b) exploitation function C_3.

5.3.2 Outside points aggregation

The objective of this step is to aggregate existing design points that are farther from the centre of the model into a smaller number of nodes (the "knots"), so that the total number of nodes and points in the model can be fitted into the memory. The problem of outside points aggregation involves hierarchical cluster analysis (HCA) [53] and a single variable optimisation design. The objective of HCA is to group outsider points into a set of clusters, so that the number of clusters is equal to the number of nodes.

There is a variety of literature that concerns cluster analysis, in particular in the field of data science, and many algorithms have been published. Points aggregation can be treated as a *k*-mean clustering problem, in which there are significantly more clusters to be identified compared to conventional clustering problems. In this thesis, we developed a sequential algorithm for weighted points clustering, the pseudo code of which is detailed below:

$for\ m = 1: sample\ size$
    $for\ n = 1: number\ of\ clusters$
        $\cdot\ calculate\ new\ cluster\ centroid$
        $\cdot\ calculate\ weighted\ Euclidean\ distance\ to\ centroid$
    end
    $\cdot\ find\ the\ cluster(x) with\ minimum\ weighted\ Euclidean\ distance\ e$
    $if\ e < weighted\ dissimilarity$
        $\cdot\ add\ point(m)\ to\ cluster(x)$
    $else$
        $\cdot\ create\ a\ new\ cluster$
    end
end

`

The cluster's centroid $\boldsymbol{o}$ of a set of m points $\boldsymbol{x}$ is given by:

$$\boldsymbol{o}(\boldsymbol{x}) = \frac{1}{p} \times \sum_{i=1}^{p} \boldsymbol{x}_i \tag{5.18}$$

where p is the number of points to be considered.

The Euclidean distance is weighted by the distance between the cluster's centroid $\boldsymbol{o}$ and the model's centre $\boldsymbol{c}$**,** based on a correlation function. The dissimilarity is also weighted, since a larger distance results in a lower correlation and therefore, a loss in information due to points aggregation will have a smaller impact on the prediction result within the centre area. Therefore, the design space is normalised and each cluster's centroid is weighted by the Gaussian function.

The Gaussian correlation function in the kriging model is used to calculate the weight w; the original function is given by:

$$f(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\theta \boldsymbol{d}_{ij}{}^2} \tag{5.19}$$

Because the hyper parameter θ needs to be tuned during the model construction process and is unknown at the stage of outside points aggregation, we specify $\theta = 2$. Doing so provides a smoother decay in correlation and provides generally sound results when the underlying problem is unknown.

The optimisation problem is defined as $OF(d) = (n - q)^2$, where d is the input variable dissimilarity, n is the number of nodes/clusters generated during the clustering process and $\boldsymbol{c}$ is the number of nodes that can be fitted into the memory. The pseudo code provided above exhibits a basic workflow of the clustering process; to speed up the process, clusters with the minimum value of $n - c$ are retained and a new clustering iteration begins with these existing clusters. The clustering process is terminated when the sum of the number of existing clusters and the number of unclassified points is less than the number of nodes previously calculated.

The following example illustrates outside points aggregation applied to a 2D scenario. Figure 5.15 (a) exhibits the clustering without Gaussian weights, while Figure 5.15 (b) illustrates the clustering with the Gaussian weight terms applied. The problem consists of 1,000 observations, assuming that the memory can build a kriging model up to 500 design points. We specify that 40% of the memory is used to store the interior points within the model's central area, while the remaining 60% is used to store nodes related to outside points. The 800 points outside the centre area are aggregated into 300 nodes (a node may consist of a single point or a group of points).

`

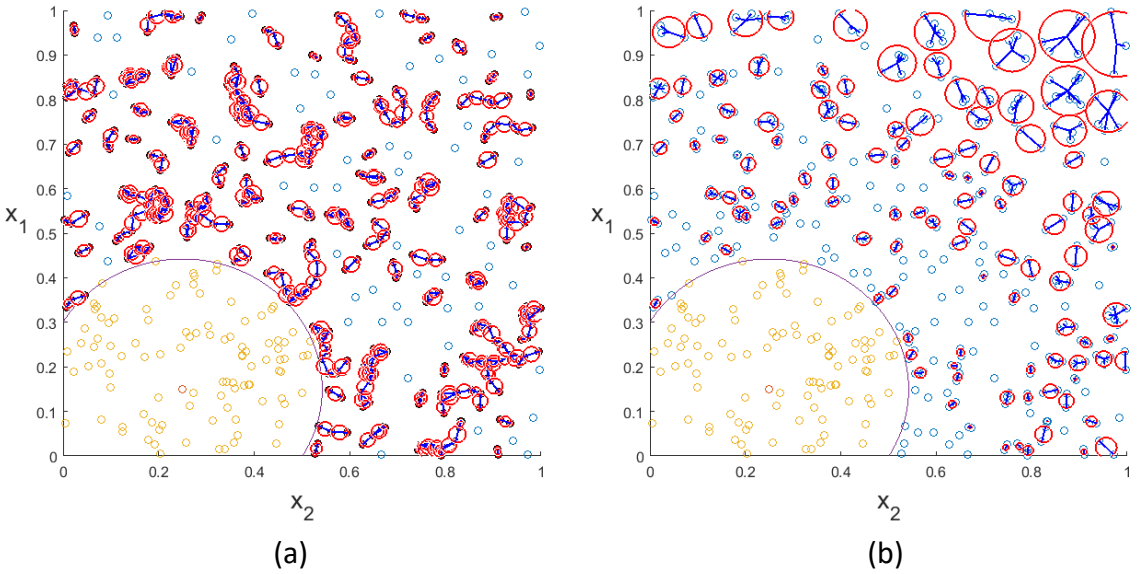


Figure 5.15. (a) Clustering without Gaussian weight functions, (b) Clustering with Gaussian weight functions.

5.3.3 A two-dimensional example

The point aggregation technique is illustrated by 2D examples in Figures 5.16 to 5.22. For demonstration purpose, the maximum number of sampling points was limited to 50, while the number of available nodes for the kriging model was limited to 25, i.e. the covariance matrix would always have 25^2 elements despite the fact that 50 design points were sampled in total. Amongst the 25 nodes allowance, 60% space is allocated for points inside the centre region, i.e. 15 nodes (original points) inside and 10 nodes (aggregated points) outside.

Figure 5.16 shows the objective function; the nodes including the aggregated points and original points are marked with an 'x' symbol, the red lines show he search path of the solver. When the number of sampling points reaches 25 (Figure 5.18) and an additional infill sampling point has been added (Figure 5.19), the points aggregation technique is triggered to keep the total number of nodes equals to 25. The blue circle shows the boundary of the centre region.

The final kriging model after points aggregation has been applied is shown in Figure 5.22, and the response surface is compared to an ordinary kriging model with 50 points in Figure 5.23. As can be seen, the proposed method produces a reasonably accurate response surface for the region around the optimal solution, while achieving 75% reduction on the covariance matrix size (25 nodes compared to 50 nodes, $1 - 25^2/50^2 = 0.75$).

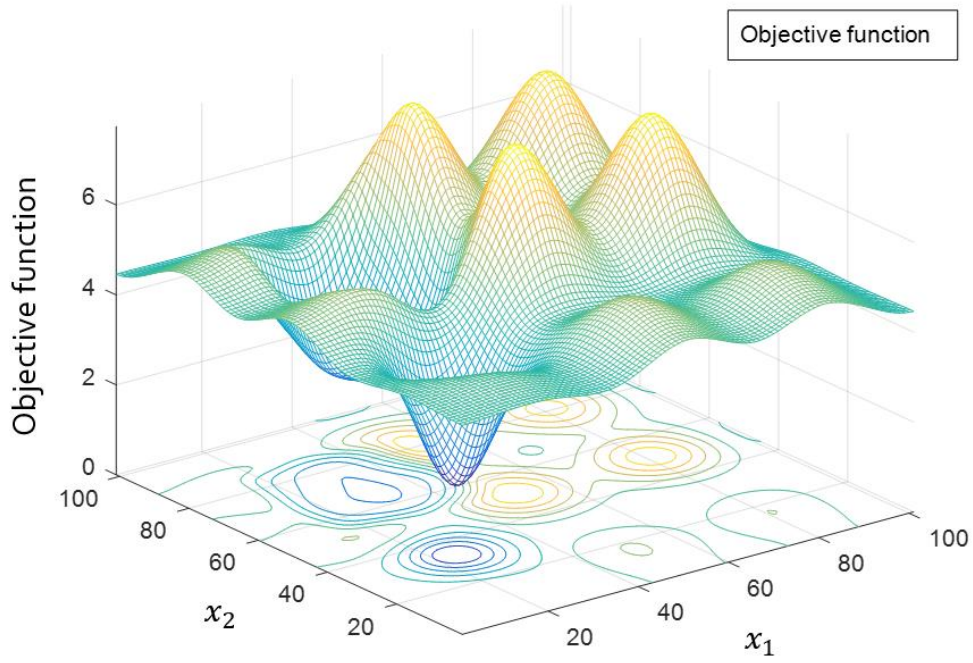


Figure 5.16. 2D-objective function.

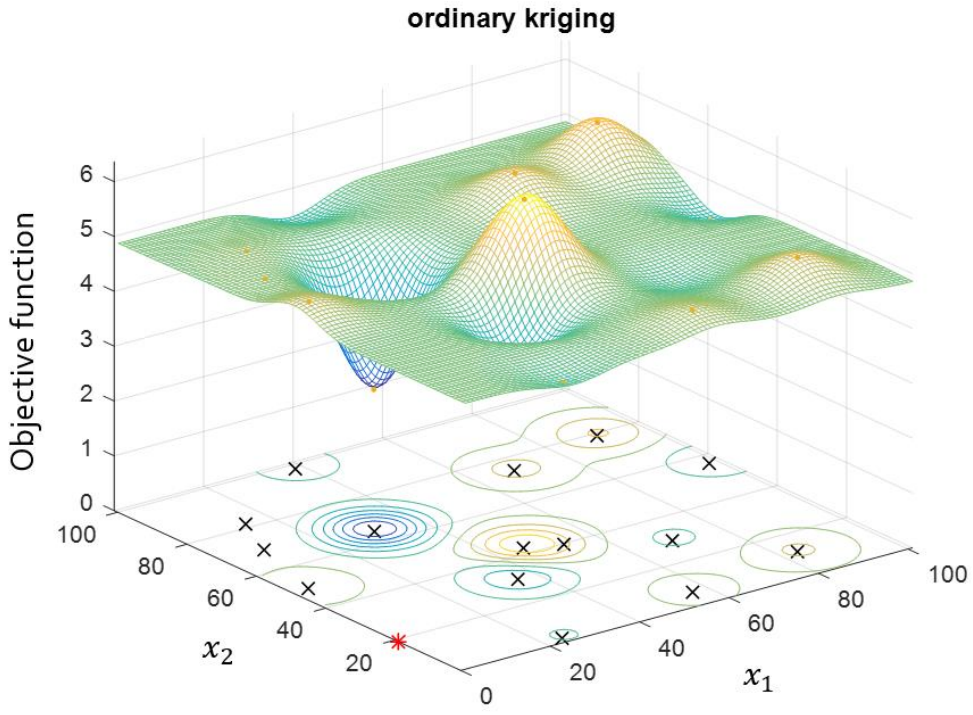


Figure 5.17. Ordinary kriging model (15 initial sampling points).

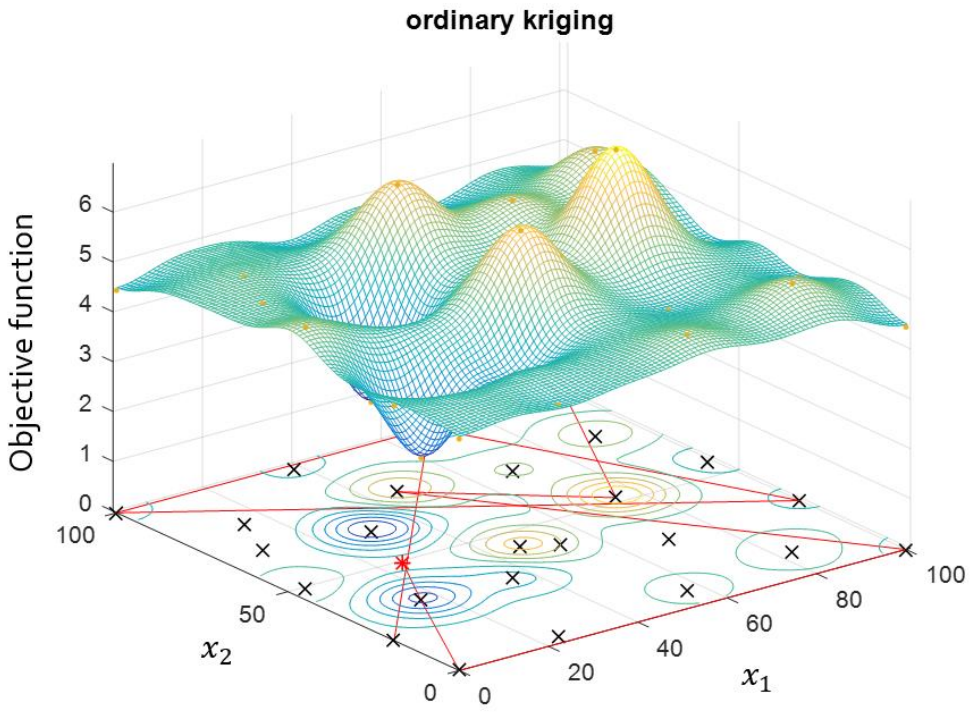


Figure 5.18. Ordinary kriging model (25 sampling points).

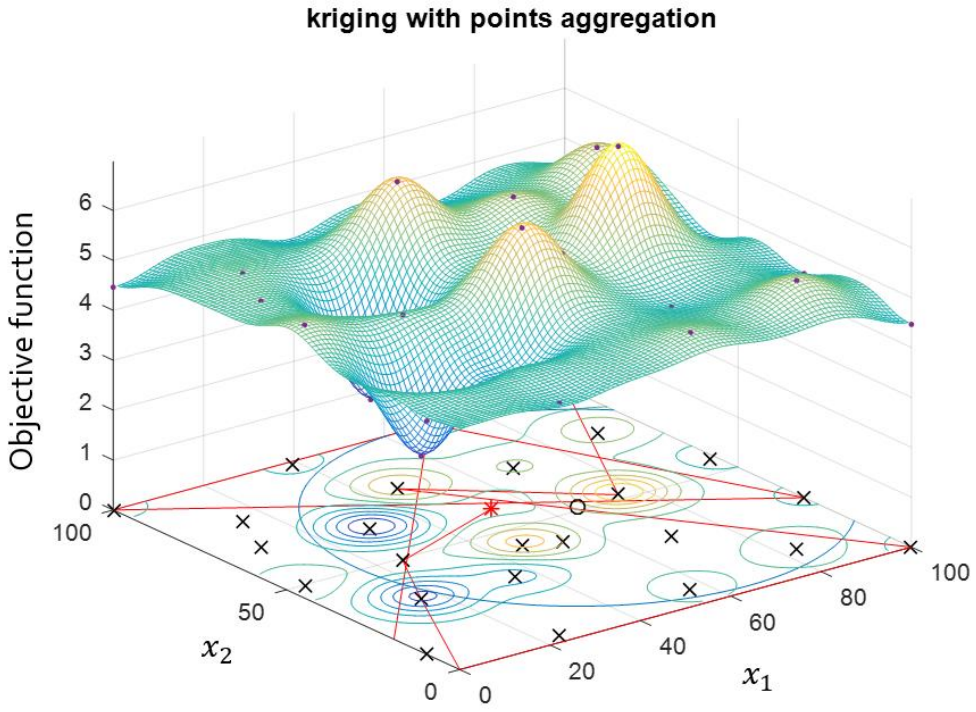


Figure 5.19. Kriging with points aggregation (26 sampling points, 25 nodes).

`

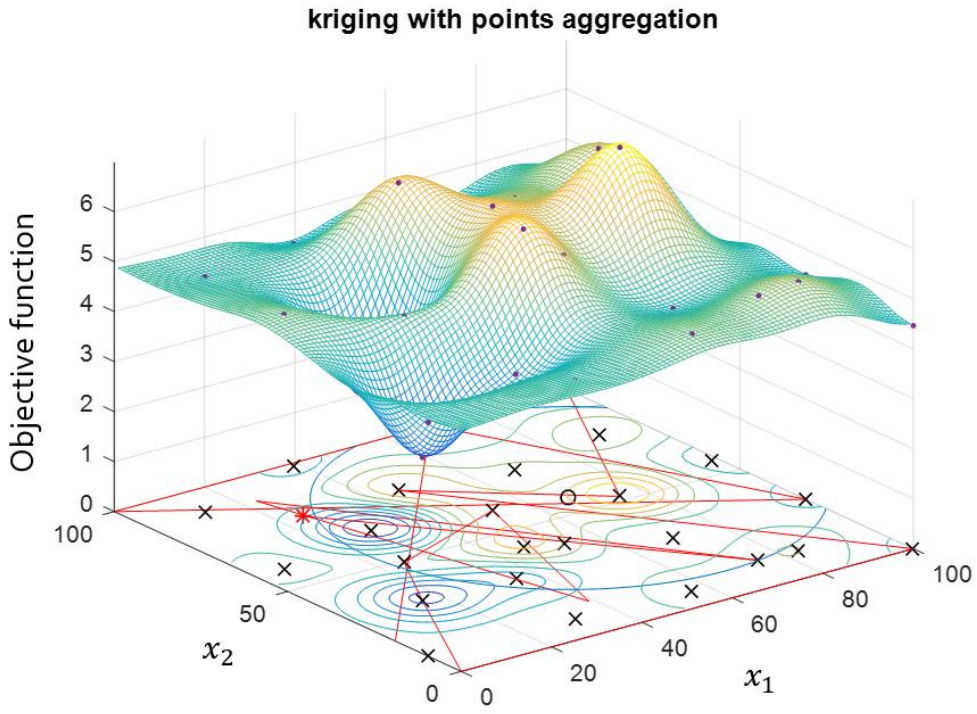


Figure 5.20. Kriging with points aggregation (30 sampling points, 25 nodes).

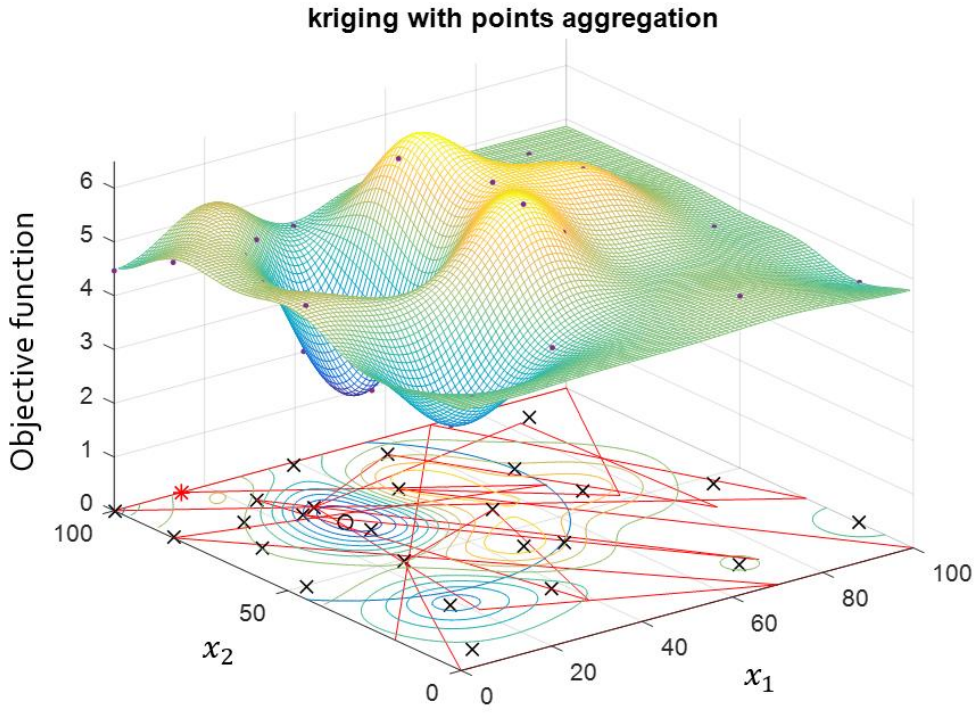


Figure 5.21. Kriging with points aggregation (40 sampling points, 25 nodes).

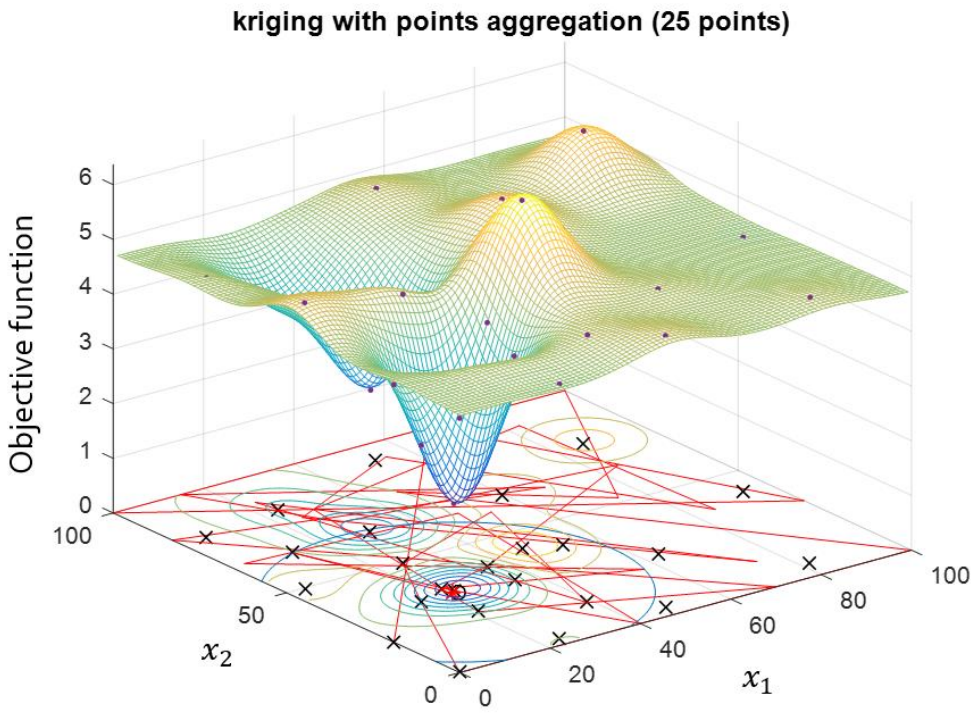
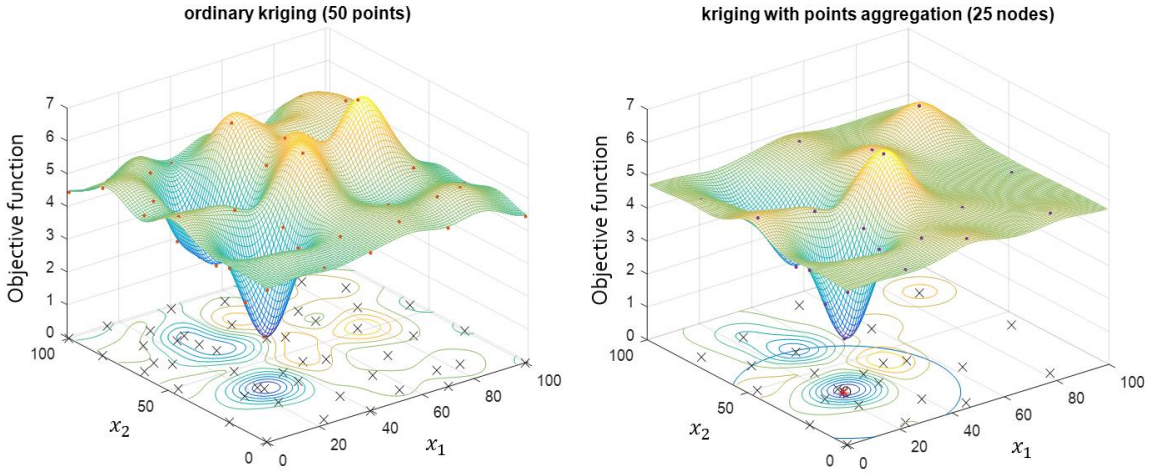


Figure 5.22. Kriging with points aggregation (50 sampling points, 25 nodes).



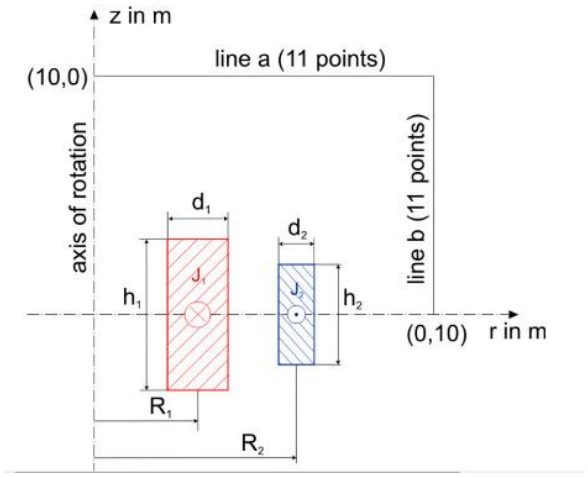
(a) (b)

Figure 5.23. (a) Ordinary kriging model (50 sampling points, 50 nodes), (b) Kriging with points aggregation (50 sampling points, 25 nodes)

`

5.3.4 Solving practical problem T.E.A.M. 22

The superconducting magnetic energy storage device in T.E.A.M. problem 22 consists of two superconducting coils, as already explained in Section 4.7.2. The design objective is to minimise the stray magnetic field while maintaining the stored energy at 180 MJ (see Figure 5.24, also Figure 4.10), subject to specified quench conditions and geometrical constraints [63]. This is an 8-parameter version of the problem.



$$OF = \frac{B_{stray}^2}{B_{norm}^2} + \frac{\left|E - E_{ref}\right|}{E_{ref}}$$

$$E_{ref} = 180\ MJ$$

$$B_{norm} = 200\ \mu T$$

$$B_{stray}^2 = \frac{\sum_{i=1}^{n=22}\left|B_{stray,i}\right|}{22}$$

,

Figure 5.24. The superconducting magnetic energy storage device (T.E.A.M. 22) [63].

Table 5.3. Optimal design variables found by different algorithms

Algorithm	R_1 (m)	R_2 (m)	h_1 (m)	h_2 (m)	d_1 (m)	d_2 (m)	J_1 (A/mm^2)	J_2 (A/mm^2)
PSO	1	2.2647	1.1076	1.7766	0.5225	0.3442	28.1779	−5.4921
Q-PSO	2.2947	2.6126	0.39	2.2704	0.3967	0.204	30	−21.293
E-QPSO	1	1.8	0.38	3.6	0.5155	0.2851	19.9975	−6.3571
GSA	1.939	2.823	0.37	1.101	0.399	0.195	22.5	−22.5
ES	1.99	2.931	0.421	0.94	0.29	0.188	26.6	−26.6
SAA	1.694	2.907	0.394	0.882	0.323	0.207	20.9	−20.9
Kriging standard	1	1.8	0.4	1.39	0.4	0.15	30	−30
Kriging proposed [1)]	3.272	3.573	1.819	1.106	0.195	0.154	26.932	−23.259
	1.11	2.319	3.193	0.28	0.259	0.734	22.5	−22.5
	1.103	2.318	3.193	0.288	0.259	0.734	22.5	−22.5
Original answer	1.296	1.8	2.178	3.026	0.583	0.195	16.955	− 18.91

`

Table 5.4. Performance comparison between different algorithms

Algorithm	Objective function	Constraints penalty [2]	No. of FEM calls
PSO	1.5673	85.0413	~6000
Q-PSO	2.4016	13.3456	~6000
E-QPSO	0.3464	0.3685	~6000
GSA	1.5547	0.195	17150
ES	0.4103	1.69235	4200
SAA	1.0087	1.09395	14000
Kriging standard	1.4065	32.9056	449
Kriging proposed [1]	0.0383	0.0159	500 [3]
	0. 0028	0.0018	826 [4]
	0.0014	0.0003	829 [4]
Original answer	0.0033 [5]	0	–

PSO: particle swarm optimisation [54], Q-PSO: quantum-behaved particle swarm optimisation [55-57], E-QPSO: QPSO with exponential probability distribution [57], GSA: global search algorithm [58], ES: evolution strategy [58], SAA: simulated annealing algorithm [58], CGM: conjugate gradient method [58]. Results for PSO, Q-PSO, E-PSO, GSA, ES and SAA taken from [54] and [58]. The comparison is for the 8 parameter continuous case [63].

Notes:

1) The new kriging algorithm offers significant savings in memory related to the correlation matrices; this has been achieved by aggregating the outside points.
2) Solutions from a number of previously published methods violated the quench condition; the degree by which this constraint was not met is given by the “penalty” (high values indicate severe violation). In some cases, neither the geometrical nor current density constraints were met.
3) For a fairer comparison of memory usage between standard kriging and the proposed kriging method, the maximum number of iterations was set to 500, while maintaining a maximum of 375 nodes; a memory savings of roughly 50% was achieved for the correlation function and moreover, a better optimum was found.
4) The proposed enhanced kriging method was allowed to continue the search with the number of nodes maintained at 500; improved results were thus achieved (better value for the objective function and lower constraint violation) after more iterations, at the modest expense of more FEM calls.
5) The value of the objective function in the original specification was slightly different; it was recalculated here using a consistent FEM model for comparison.

`

Table 5.5. Comparison of time cost between various optimisation methods

Algorithms	Time (s) per 10^3 iterations (algorithm)[1)]	Estimated time (ms) per iteration (algorithm)	Time (s) per FEM call	Number of iterations	Estimated overall time (h)
PSO	-	-	~22	~6000	~36.7
Q-PSO	-	-	~22	~6000	~36.7
E-QPSO	-	-	~22	~6000	~36.7
GSA	-	-	~22	17150	~105
ES	-	-	~22	4200	~25.7
SAA	-	-		14000	~85.6
Kriging standard	52.04	52.04	~22	449	~2.8
Kriging proposed	-	552[2)]	~22	500	~3.1
	-	447[2)]	~22	826	~5.2
	-	449[2)]	~22	829	~5.2

1) Experiments were carried out on a 3.4GHz PC. The MATLAB codes for PSO, Q-PSO, E-QPSO, GSA, ES and SAA were not available, but the times to run non-surrogate model-based optimisation algorithms are very short compared to the time required to execute each FEM, hence they can be neglected without affecting the comparison.

2) The proposed kriging approach involves an extra computation overhead, thus the time cost per iteration increases slightly as more infill points are added, while this extra computation overhead is independent of the FEM model. Simulated annealing is roughly constant at each iteration, for each 1000 evaluations; its runtime is around 2 seconds. The KNN algorithm has complexity of O(dkn) (d: problem dimension, k: number of nearby points, n: number of total points) and is proportional to the number of existing design points; for 1000 existing points in the 8-D parameter space, searching for the 40% of neighbours in each iteration takes around 0.002 seconds. Moreover, the computation time for fitness functions in the model centring step is negligible. The points aggregation involves optimizing a convex single variable objective function; this process takes 10-15 iterations and total time approximately 0.3 seconds for 1000 existing points in the 8-D parameter space. The overall computation overhead is therefore around 2 seconds. However, the computation overhead only occurs when the number of sampling points exceeds the maximum allowance. Therefore, for tests with different number of sampling points and the maximum nodes allowance, the average time cost per iteration is different.

In this example, a penalty constraint has been imposed to the objective function, because inequality constraint is critical to ensure the conductivity states; the penalty has been directly applied to the objective function treated as a single objective problem instead of a bi-objective problem. As can be seen from Table 5.4, the proposed kriging approach has found an optimum at the 500th iteration which is quite different to those found by other methods in the table. It is possible that the proposed kriging method has not yet converged or simply converged to a local optimum. At the 826th and 829th iteration, it then found two solutions that are closer to those obtained by other methods. Because the problem dimension is large in this case (8 parameters), convergence

towards a global optimum cannot be guaranteed with confidence, as the sampling points are too sparse in the search space with a large number of parameters and there will be many areas left unexplored. This also explains why the optimum location obtained by all the methods differ significantly.

The experiment results are summarised in Table 5.4. As on previous occasions, kriging has demonstrated its superiority by dramatically reducing the number of necessary function calls and thus, avoiding excessive use of the computationally expensive finite element software. Moreover, the addition of the points aggregation offers the additional flexibility of limiting the number of active points in the design space; this has the benefit of reducing the memory requirements for the solution without sacrificing the accuracy. Finally, the iterations were able to continue to achieve a better design with a modest increase of computational effort due to the need for more FEM calls.

The computation costs of various optimisation algorithms have also been displayed in Table 5.5. Both the standard kriging approach and the kriging approach with points aggregation have shown a significant advantage in terms of overall computation time over other direct optimisation algorithms.

A kriging-based optimisation approach for large datasets was proposed and its efficiency demonstrated using the T.E.A.M. 22 problem. The model centre positioning algorithm balances exploration and exploitation assisted by the use of a stochastic approach, which eliminates the risk of a deterministic criterion function being trapped in a local optimum. It was found that the size of the correlation matrices can be greatly reduced by applying points aggregation techniques. Indeed, the proposed approach can fit a large set of data into a limited size of memory and whereas information about remote points might be lost, this is alleviated by the use of points aggregation incorporating a new weighted clustering algorithm.

5.4 Conclusion

We briefly discussed the challenge of using the kriging model-based optimisation approach in handling problems with a large dataset and proposed two independent kriging-based approaches for handling such problems. In both cases, the problem was solved at the expense of some information, while the loss of information was kept as small as possible by utilising another layer of optimisation. Both methods presented in this chapter were tested against our test functions and T.E.A.M. practical benchmark problems and the results were compared to other algorithms. The test results indicate that the proposed methods provide a memory savings of 36% for the covariance matrix in the first case and 50% in the second case.

`

Chapter 6. Multi-objective optimisation

6.1 Introduction

In previous chapters, only single objective optimisation problems were considered. In many practical design scenarios however, designers often have to deal with multiple conflicting objectives, in which the improvement of one objective may not be possible without the deterioration of another. Hence, compared to single objective optimisation problems, multi-objective optimisation problems (MOOPs) are more complex and more difficult to solve, and unlike single objective optimisation problems, there often exist multiple solutions for which decisions must be made. Optimisation involving multiple objectives is referred to as multi-objective optimisation (MOO).

The development of multi-objective optimisation theory dates back to the late 1800s, when Francis Francis Ysidro Edgeworth (1845-1926) first defined the concept of trade-off for multiple conflicting objectives [59]. Throughout the 20th century, the development of multi-objective optimisation was largely brought forward by studies in the area of engineering, mathematical economics and political science.

Problem definition:

$$\text{Minimise } \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})\} \tag{6.1}$$

$$\text{Subject to } x \in \boldsymbol{D}$$

$$g_i(\boldsymbol{x}) \leq 0,$$

$$h_i(\boldsymbol{x}) = 0, \qquad i = 1, 2 \dots, n.$$

where n is the number of objectives, $f(x)$ is the objective function, $\boldsymbol{D}$ is a non-empty set of feasible design points, $g_i(x)$ are the inequality constraints and $h_i(\boldsymbol{x})$ are the equality constraints.

6.1.1 Scalarization methods

Scalarization methods are arguably the most widely used multi-objective approaches due to their simplicity. The general idea is to combine all the objectives to form a single objective problem. For problems with a priori knowledge of importance for each of the objectives; weights and scaling factors commonly used in scalarization methods can also reflect the designer's preference. In cases in which the importance of each objective is not clear, by systematically varying the method parameters, scalarization methods are also able to generate a set of Pareto points.

The most generic form of scalarization method is the weighted sum approach:

`

$$Y = \sum_{i=1}^{n} w_i F_i(x) \tag{6.2}$$

where i indicates the i^{th} objective, $F_i(x)$ is the i^{th} objective function and w_i is the weight applied to the i^{th} objective function.

Other scalarization methods include: the weighted exponential sum method, the weighted min-max method and the weighted product method. For a comprehensive review of the scalarization methods, as well as other non-EA based approaches, readers are referred to [60].

6.1.2 Evolutionary algorithm-based methods

An increasing number of evolutionary algorithm-based solution methods for MO problems have appeared in the literature since the mid-1980s. The vector evaluated genetic algorithm (VEGA) proposed in 1985 [61] is one of the earliest examples of multi-objective evolutionary algorithms (MOEAs). During the past 30 years, numerous MOEAs have been proposed. Some of the most often cited are: niched Pareto genetic algorithm (NPGA) [63], strength Pareto evolutionary algorithm (SPEA), strength Pareto evolutionary algorithm-2 (SPEA2), Pareto-archived evolution strategy (PAES), Pareto envelope-based selection algorithm (PESA), Pareto envelope-based selection algorithm-II (PESA-II), non-dominated sorting genetic algorithm (NSGA) [62], non-dominated sorting genetic algorithm NSGA-II [64] and particle swarm (PSO) based methods [65].

The general opinion favours EAs as advantageous in solving MOO problems by often being population based, thus multiple solutions can be obtained in a single run. However, solutions to practical problems are usually expensive in terms of computational time and effort. In the context of electromagnetic devices, the finite element method is a common design tool; it often takes hours or even days to obtain a single solution. Therefore, surrogate model-based algorithms are often preferred.

In addition to the aforementioned algorithms, an increasing number of indicator-based MO algorithms have been proposed in recent years; the indicator is used as a fitness measure for a set of Pareto points and – by optimising the indicator function – the MO problem essentially becomes a single objective optimisation problem, since the solver only needs to locate the optimal value of the indicator value and update the generation based on it. Examples include: epsilon indicator [66], R2 indicator [69], additive -ε indicator[67] and the hypervolume indicator[68]. Comparisons between indicators can be found in [66] [69] [70].

`

6.2 Hypervolume indicator

A hypervolume indicator, also known as a Lebesgue measure or S metric, is the most widely used indicator and has been successfully applied to both EAs and surrogate-based algorithms. Hypervolume is the Lebesgue measure bounded by the non-dominated solutions and the reference point, for a normalised objective space; as [1 … 1] are often used as the reference points, the dimension of the array is equal to the number of objectives. Figures 6.1 (a) and (b) present the known design sites in the objective space and the corresponding hypervolume (grey area) of this set of points. The black dots represent the Pareto points.

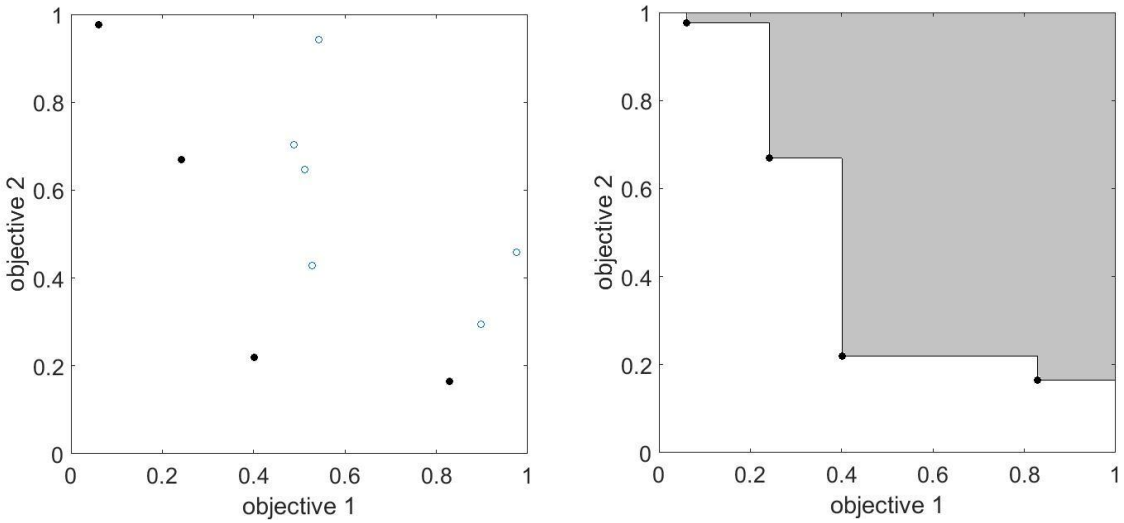


Figure 6.1. (a) Design points in the objective space (b) the hypervolume of the Pareto front.

In kriging-based multi-objective optimisation, the hypervolume indicator is modified based on the ideal expected improvement (EI) infill criterion and the hybridised version, widely known as the expected hypervolume improvement (EHVI). The formula for EHVI is exceedingly similar to EI, except the expected improvement is measured in the hypervolume instead of the value of objective function in the search space.

The hypervolume improvement (HVI) is the increment in hypervolume based on the new points and the old design points. Its analytical form is given by:

$$HVI = HV(\{PF, y\}, ref) - HV(PF, ref) \tag{6.4}$$

where PF stands for Pareto front, $HV(PF, ref)$ stands for the hypervolume bounded by the Pareto front and the reference point ref and y is the newly added design point.

The formula for EHVI is given in [71] [72]:

$$EHVI = \int_{-\infty}^{ref}\int_{-\infty}^{ref}\cdots\int_{-\infty}^{ref} HVI[f_1(x), f_2(x), \cdots, f_n(x)] \cdot \phi(F_1)\cdot\phi(F_2)\cdots\phi(F_n)\, dF_1 dF_2 \cdots dF_n \tag{6.3}$$

where $\phi(\cdot)$ is the probability density function and n is the number of objectives.

An example of a kriging-based multi-objective optimisation approach based on EHVI is illustrated in Figures 6.2 to 6.4. The dashed lines are the underlying test functions, the blue and orange lines represent the kriging model for two different objective functions at the 14th iteration. The location of the Pareto front is marked by red crosses on the test function.

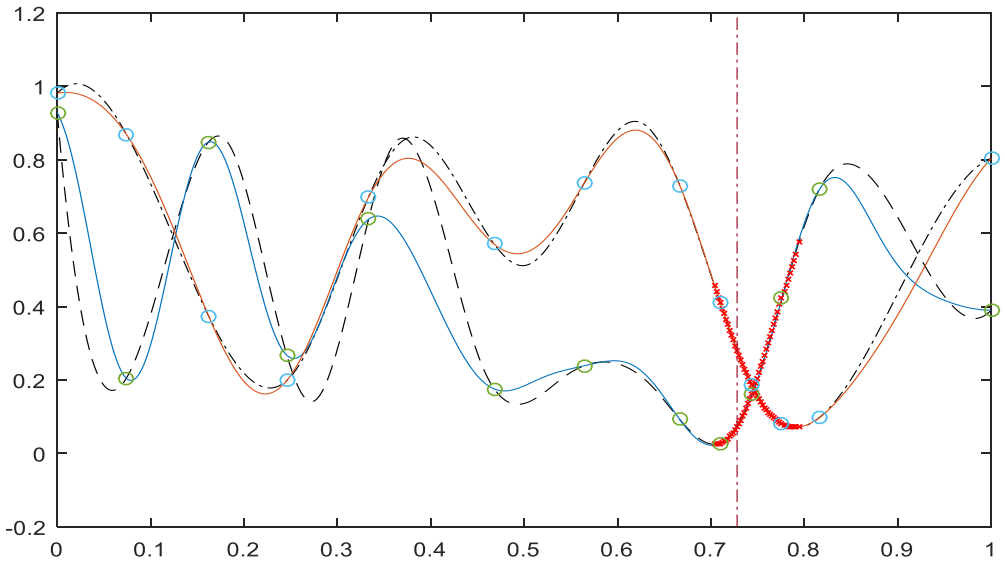


Figure 6.2. Kriging models in the search space.

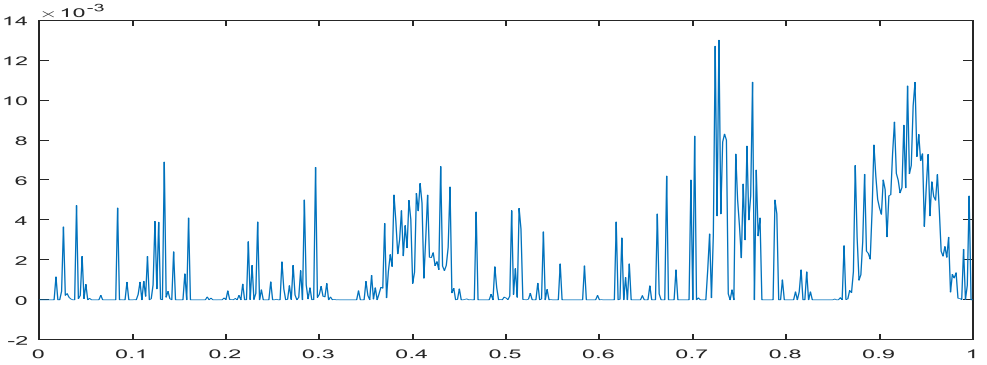


Figure 6.3. Expected hypervolume improvement in the search space.

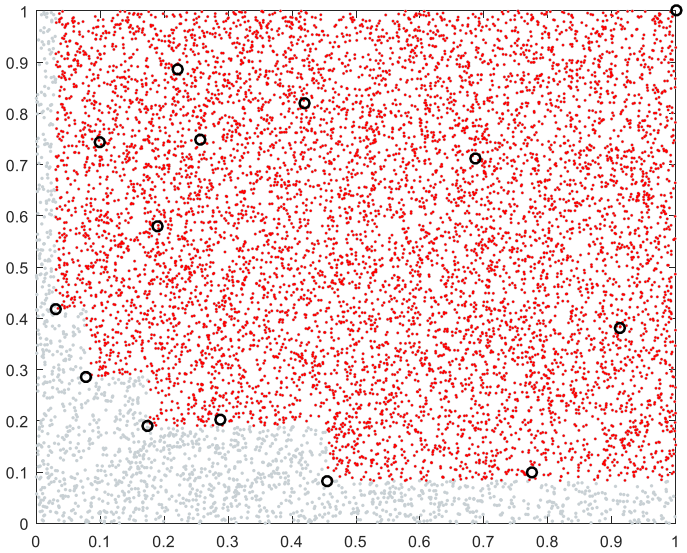


Figure 6.4. Calculating the hypervolume via the Quasi Monte Carlo approach.

As can be observed in Figure 6.4, the Monte Carlo approach only provides an approximation of the hypervolume measure; when the displacement of the new sampling points from the current set of Pareto points is small, its accuracy may not be sufficient. More sophisticated methods for computing the hypervolume have been proposed within the literature in recent years. Two algorithms (IIHSO and WFG) were proposed in the PhD thesis [74] and other examples can be found in [75], [76], [77], [78].

The most important feature of a hypervolume indicator is that this measure is strictly monotonic to Pareto improvement [79]. In other words, infill points resulting in a higher hypervolume measure always improve the current Pareto solution. Despite this favourable feature, its high computational costs for higher dimensions is also widely known. The computation of hypervolume is an NP-hard problem: the complexity increases exponentially in number of objectives.

6.3 Localised probability of improvement

In this section, we introduce a novel approach to kriging-based multi-objective optimisation by utilising a local probability of improvement as the infill sampling criterion and the nearest neighbour check to ensure diversification and uniform distribution of the Pareto fronts.

6.3.1 Probability of improvement

Kriging provided both the predicted mean and the associated mean square error at an unknown location. The probability of improvement PoI at any location is given by:

`

$$PoI(\boldsymbol{x}) = \Phi\left(\frac{y_t(\boldsymbol{x}) - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) \tag{6.5}$$

where y_t is the target of improvement, $\hat{y}$ is the kriging predicted mean at location $\boldsymbol{x}$, $\hat{s}^2$ is the mean square error at location $\boldsymbol{x}$ and $\Phi(\cdot)$ is the cumulative distribution function.

From the above equation, it can be observed that the value of PoI depends on the target y_t, predicted mean $\hat{y}$, and the mean square error $\hat{s}$, where the target can be modified to obtain a "manipulated" value of the PoI.

The algorithm presented in this section is a localised approach to calculate the probability of improvement at an unknown location. The term "localised" is added here because the redefined probability of improvement for each point is calculated based on their location in the objective space, and unknown points at different regions may have different location targets of improvement. Hence, for simplicity, we define the indicator as the $LPoI$.

$LPoI$ at a given location is an integrated measure calculated for two improvement targets. A target is calculated for a reference point and the reference point is taken based on the location of the unknown point $\boldsymbol{x}$.

6.3.2 The first improvement target

The first improvement target y_{ext} associates with the minimum value of each individual objective function, the subscript ext stands for "extreme value"; and y_{ext} is given as:

$${y_{ext}}^n = {y_{min}}^n \cdot (1 - p) \tag{6.6}$$

where ${y_{min}}^n$ is the known minimum value of the n^{th} objective function and p is the percentage of improvement to be defined; parameter p is discussed later in this section. The corresponding PoI is:

$${PoI_{ext}}^n(\boldsymbol{x}) = \Phi\left(\frac{{y_{ext}}^n - \hat{y}^n(\boldsymbol{x})}{\hat{s}^n(\boldsymbol{x})}\right) \tag{6.7}$$

where $\hat{y}^n$, $\hat{s}^n$, ${y_{ext}}^n$ and ${PoI_{ext}}^n$ are the corresponding measures of the n^{th}objective function.

For the first improvement target, we get n number of PoI, which equals the number of objectives because the PoI_{ext} is calculated based on the extreme value (maximum) of each objective functions. We consider the maximum potential improvement for all individual objectives, hence:

$$PoI_{ext}(\boldsymbol{x}) = maximize\{{PoI_{ext}}^n(\boldsymbol{x})\} \tag{6.8}$$

`

6.3.3 The second improvement target

The second improvement target ${y_{int}}^n(\boldsymbol{x})$ is associated with the reference point that is defined based on the location of $\boldsymbol{x}$. The subscript int stands for "intermediate" and y_{ref} is calculated as:

$$ {y_{int}}^n = {y_{ref}}^n \cdot (1 - p) \tag{6.9} $$

where y_{ref} is the calculated reference point.

To obtain the reference point y_{ref}, the algorithm finds the Pareto front for existing design sites using non-dominated sorting. For each closest set of Pareto points (the number points is equal to the number of objectives) it calculates the corresponding reference point. The coordinates for the reference point of each dimension is equal to the maximum value of the coordinates for these Pareto points in the same dimension. The coordinates for the corresponding reference point in the nth dimension $Ref(x^n)$ is given by:

$$ {y_{ref}}^n = \max\{Y^n\} \tag{6.10} $$

where Y^n is the collection of the n^{th} objective value for all of the points in that Pareto set.

Taking a bi-objective problem as an example, assuming the reference point y_{ref} is to be determined for Pareto points P_1 and P_2, the coordinates of P_1 and P_2 are hence denoted by $[P_1.x^1, P_1.x^2]$ and $[P_2.x^1, P_2.x^2]$, respectively. Note that x^n is the n^{th} objective value at the location in the search space associated with P. The x^1 and x^2 coordinates (in the objective space) of the reference point are thus described as follows:

$$ {y_{ref}}^1 = max\{P_1.x^1, P_2.x^1\} \tag{6.11} $$

$$ {y_{ref}}^1 = max\{P_1.x^2, P_2.x^2\} \tag{6.12} $$

and the corresponding PoI is given as:

$$ {PoI_{ref}}^n(\boldsymbol{x}) = \Phi\left(\frac{{y_{int}}^n(\boldsymbol{x}) - \hat{y}^n(\boldsymbol{x})}{\hat{s}^n(\boldsymbol{x})}\right) \tag{6.13} $$

where ${y_{ref}}^n$, $\hat{y}^n$, $\hat{s}^n$, ${PoI_{ref}}^n$ and ${PoI_{ext}}^n$ are the corresponding measures of the n^{th}objective function.

We obtained n number of PoI for the second improvement target. However, unlike the first improvement target, the second one uses a localised target. Therefore, consider using the minimum potential improvement for all individual objectives and hence:

$$ LPoI_{ref}(\boldsymbol{x}) = minimize\{{PoI_{ref}}^n(\boldsymbol{x})\} \tag{6.14} $$

`

6.3.4 Integrated improvement target

Finally, the proposed indicator $LPoI$ for any given point is the maximum of these two probability of improvement measures, as follows:

$$LPoI(\boldsymbol{x}) = maximize\{LPoI_{ref}, PoI_{ext}\} \tag{6.15}$$

PoI_{ext}, as described by (6.8), is due to the fact that the minimum of each individual objective function is always present in the Pareto front, hence the PoI at each location $\boldsymbol{x}$ over the optimal target of that function is always considered. This term also contributes to the diversification of the Pareto front.

Furthermore, $LPoI_{ref}$, as described by (6.14), can be treated as the maximum of the minimum potential improvement to a local target. This term helps to improve the Pareto front both toward the origin and in the direction of the objective value. It contributes to the diversification of the Pareto front.

To obtain the next infill sampling point, the algorithm finds the location $\boldsymbol{x}$ associated with the maximum $LPoI$ measure in the objective space.

6.3.5 Parameter $\boldsymbol{p}$

Parameter p, as seen in (6.6) and (6.9), is associated with the magnitude of target improvement; it controls the convergence rate of the algorithm. A smaller improvement amount will guide the solver toward existing Pareto points, while a larger value will encourage the exploration of the design space. It is crucial to use a proper p, since too small a value may lead to a false Pareto front, while a large value may result in a slow convergence rate or zero probability of improvement at all unknown sites. Thus, it is better to dynamically correct the value while monitoring the convergence.

We first provide a simple self-adjusted method for parameter p.

First, the initial improvement target percentage $p_{initial}$ is defined and then the parameter p is calculated as:

$$p = p_{initial} \cdot \max\{LPoI_{prev}\} \tag{6.16}$$

where $LPoI_{prev}$ is the complete set of $LPoI$ at the previous iteration.

The next infill point is taken at the location with maximum $LPoI$. Therefore, the solver tends to minimise the localised probability of improvement and converge toward the Pareto front. When the design space is well explored, or the p is especially small, the solver will converge toward existing Pareto fronts; at this stage, it is common for the $LPoI$ to equal or come close to one at multiple unknown sites (extremely likely to improve over the target point). In order to obtain a uniformly distributed Pareto front, the algorithm selects candidates which have the largest Euclidean distance to existing Pareto points compared to the next infill sampling points. For this reason, the maximum

`

value of $LPoI$ can be capped between 0.9 and 1 for faster exploitation of the existing Pareto front without degrading the overall performance

6.4 Test functions and examples

6.4.1 Bi-objective example

Considering the following two test functions plotted in black dotted lines in Figure 6.5, the circles are sampled design points and the red and blue lines are kriging models built on existing samples. The red crosses mark the locations of all of the Pareto points for each objective function. The true Pareto points can be obtained by mapping all the feasible solutions on to the objective space and running a simple non-dominated sorting method (given in the appendix). When the Pareto points are found in the objective space, their corresponding location of x in the search space is marked in the figure below.

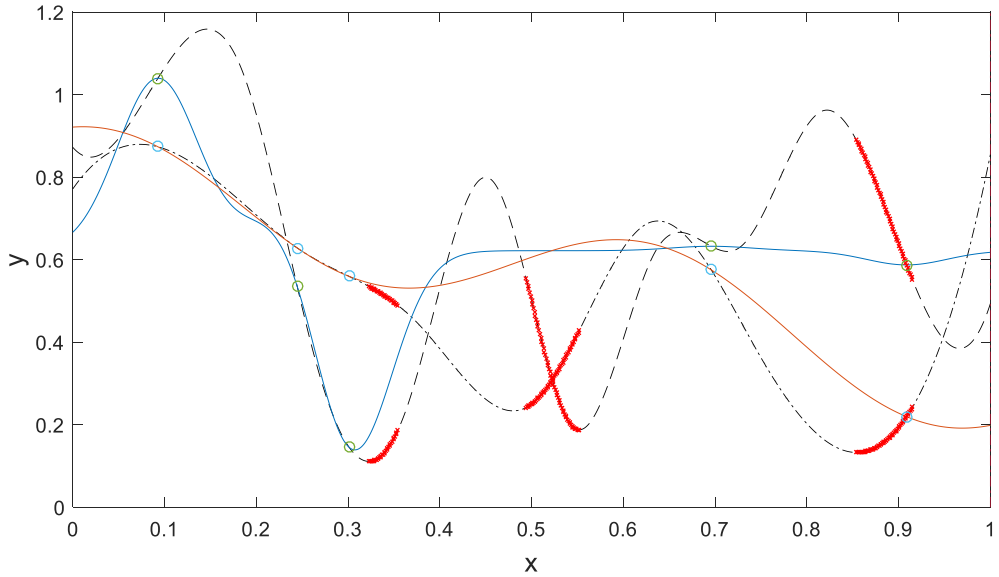


Figure 6.5. Kriging models at the 5th iteration.

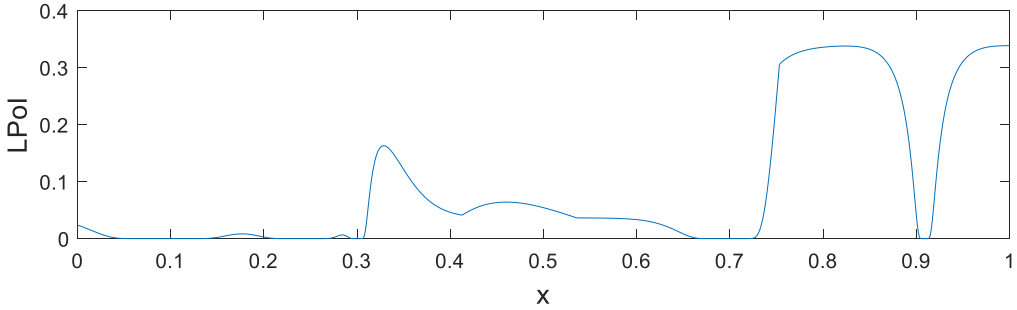


Figure 6.6. $LPoI$ criterion at the 5th iteration.

Figure 6.6 displays the $LPoI$ criterion in the search space. While the maximum location is not obvious in the figure, the location at x=1 is sampled in the next iteration.

The kriging model in the objective space is illustrated in Figure 6.7. The dashed line shows all the feasible solution in the objective space for $0 \leq x \leq 1$, the minor red crosses indicate the true Pareto front, and the blue solid line provides a direct mapping of two kriging models for the search space; it shows an estimation of all feasible solutions (when two kriging models are accurate enough, the blue solid line should coincide with the dashed line). Since the sampling points are too sparse at the early stage, the estimated model in the objective space is far from accurate and more infill sampling points need to be added.

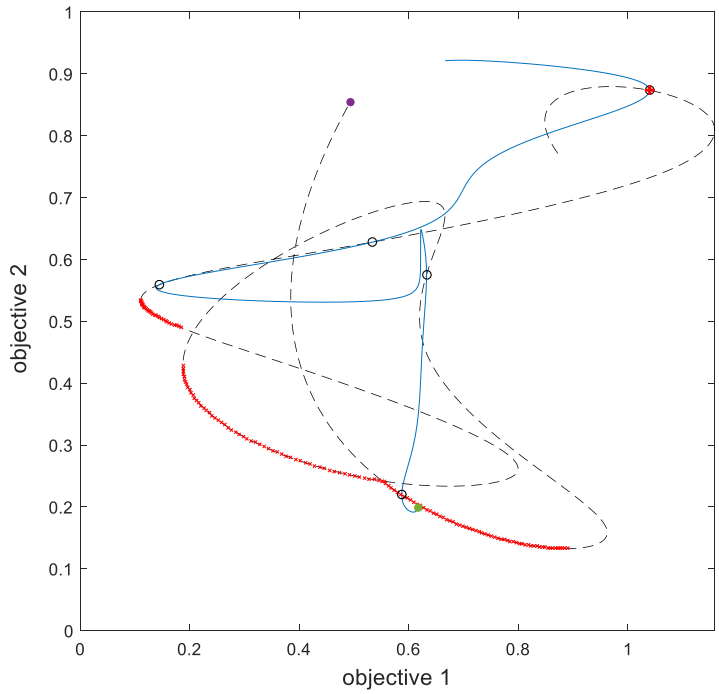


Figure 6.7. Kriging model in the objective space.

Figure 6.8 shows that at the 17th iteration, kriging models are reasonably accurate in the search space. As can be seen in Figure 6.9, the solver has started to converge toward the two Pareto clusters (marked with red crosses in Figure 6.8). The kriging model in Figure 6.10 does not look as accurate and has left the location away from the Pareto front unexplored, because the probability of improvement on the existing Pareto front in these locations are relatively small.

`

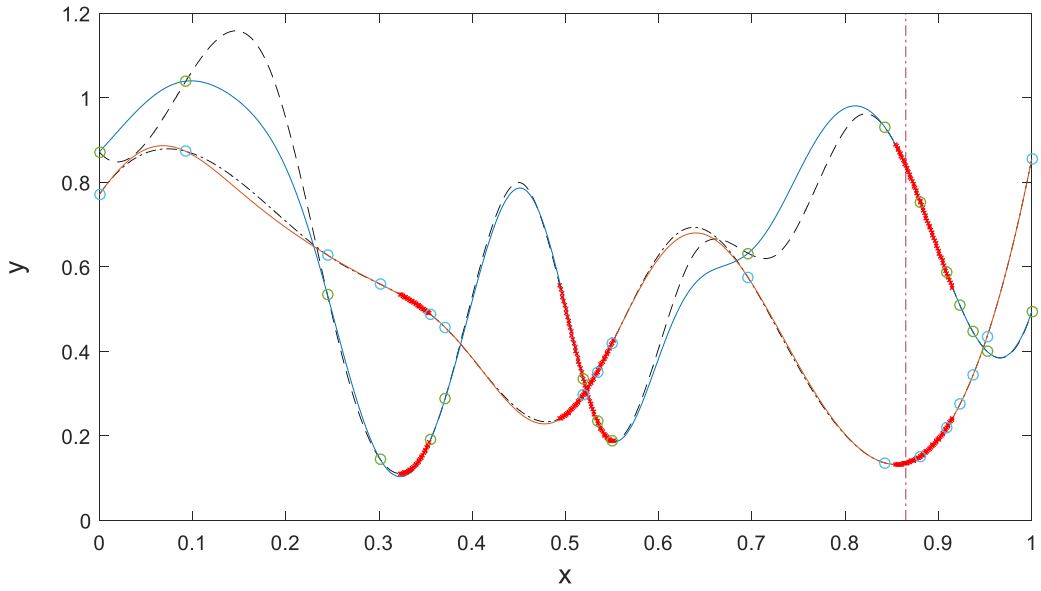


Figure 6.8. Kriging models at the 17th iteration.

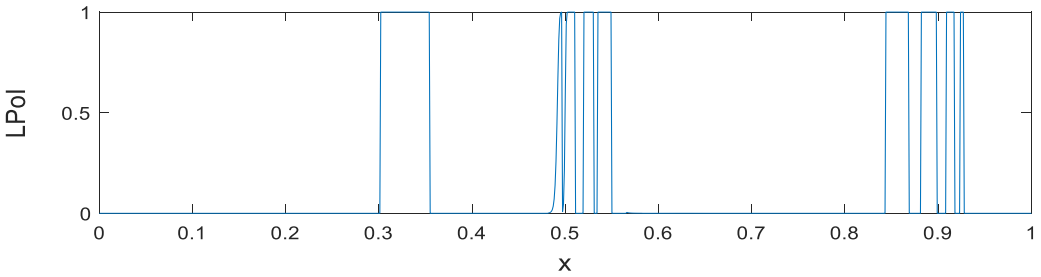


Figure 6.9. $LPoI$ criterion at the 17th iteration.

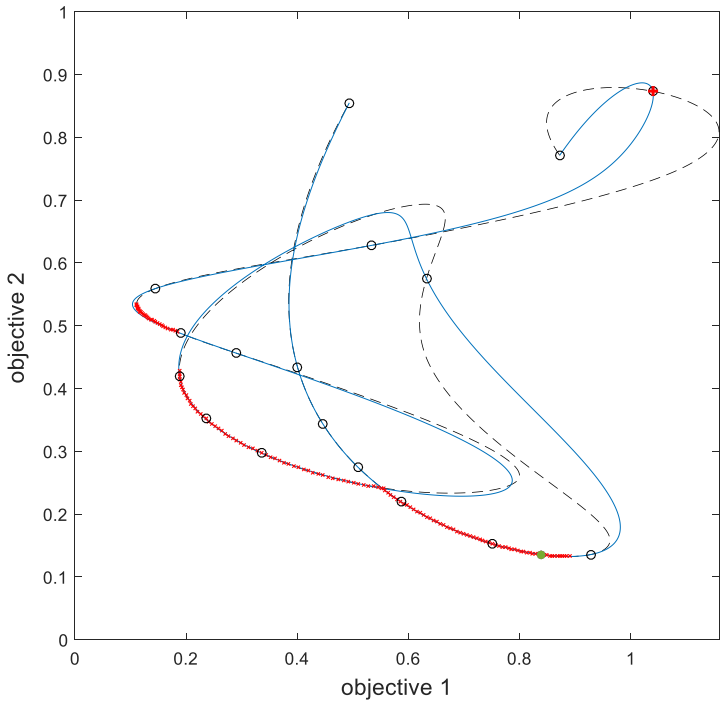


Figure 6.10. Kriging model in the objective space.

Figure 6.11 displays the kriging models at the final iteration (40th iteration) and all of the sample points are plotted in Figure 6.12, where the true Pareto front is marked with red crosses. The solver has successfully converged and infill points are taken uniformly along the Pareto front.

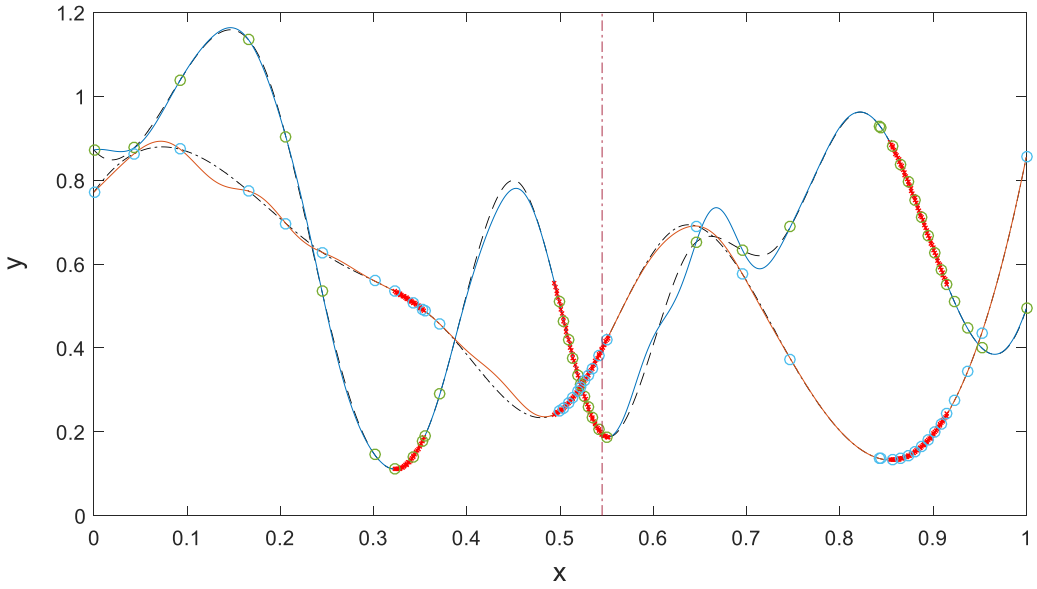


Figure 6.11. Kriging models at the 40th iteration.

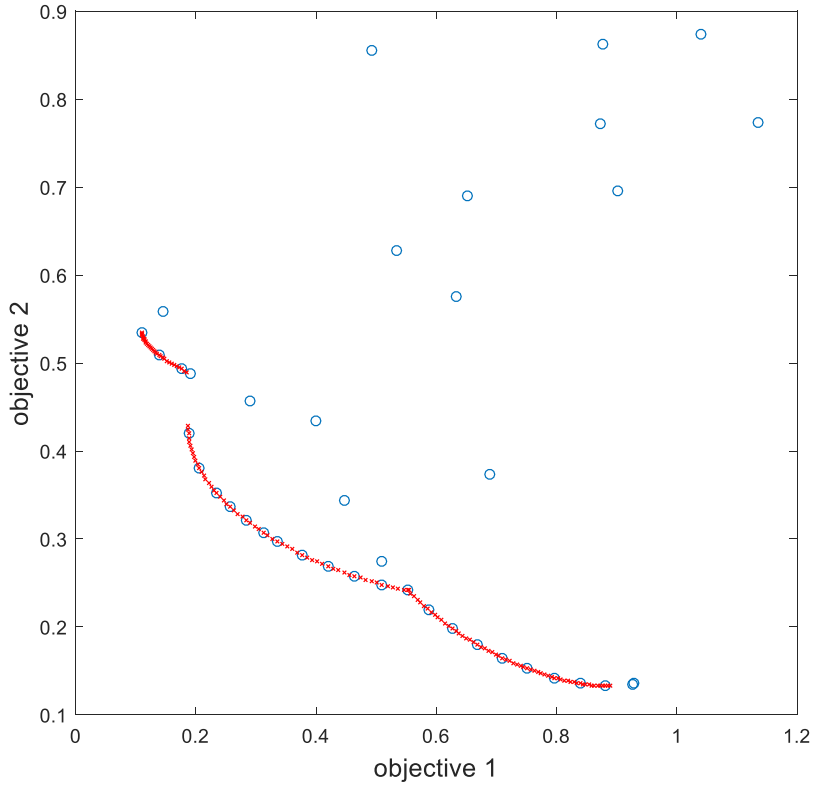
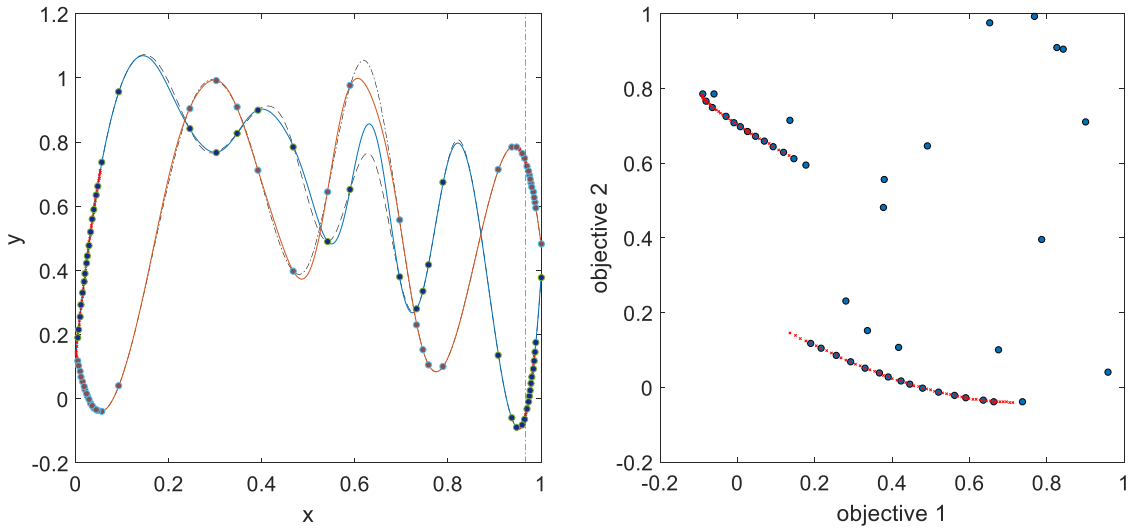


Figure 6.12. Design points in the objective space.

`

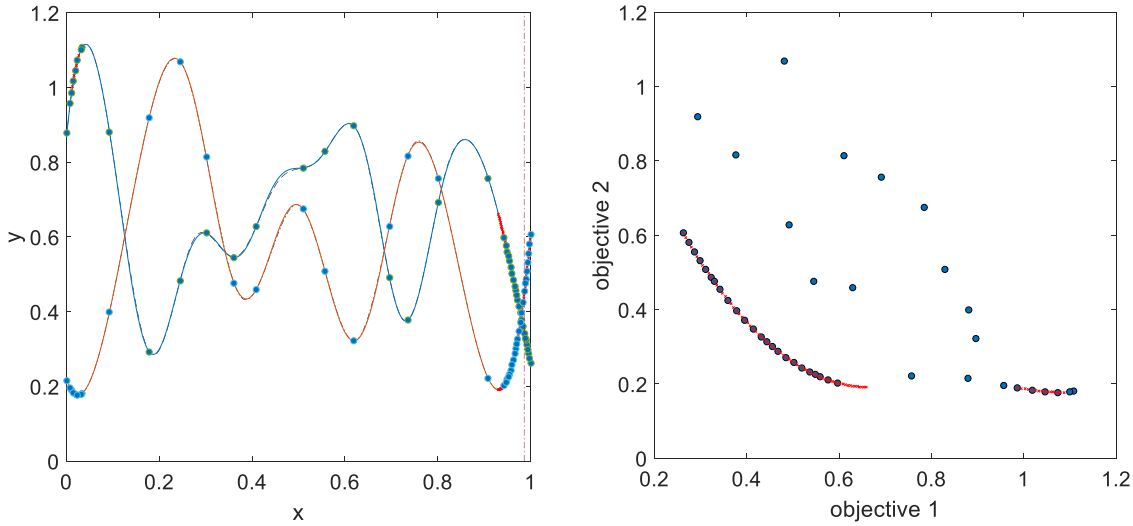
6.4.2 Bi-objectives problems and test results

A large number of tests were conducted with different, randomly generated bi-objective test functions. The examples below start with 5 initial design points and 45 maximum iterations. Figures 6.13 and 6.14 display the final results from two randomly generated test functions.



(a) (b)

Figure 6.13. (a) Kriging models in the search space, (b) design points in the objective space.



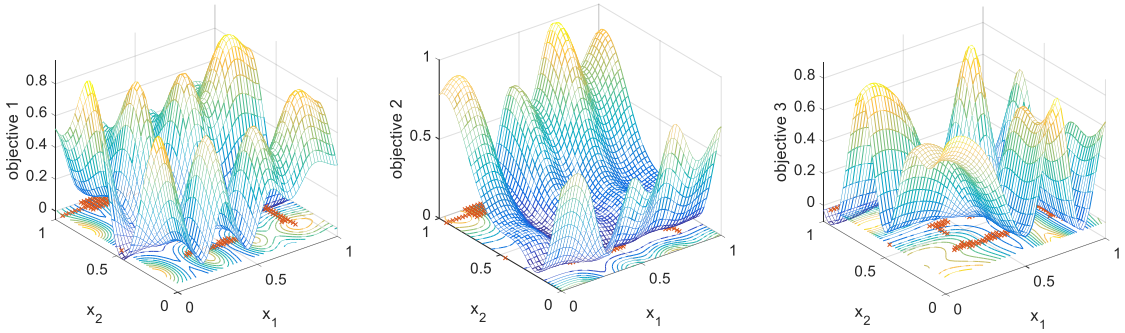
(a) (b)

Figure 6.14. (a) Kriging models in the search space, (b) design points in the objective space.

6.4.3 Three objective problems and test results

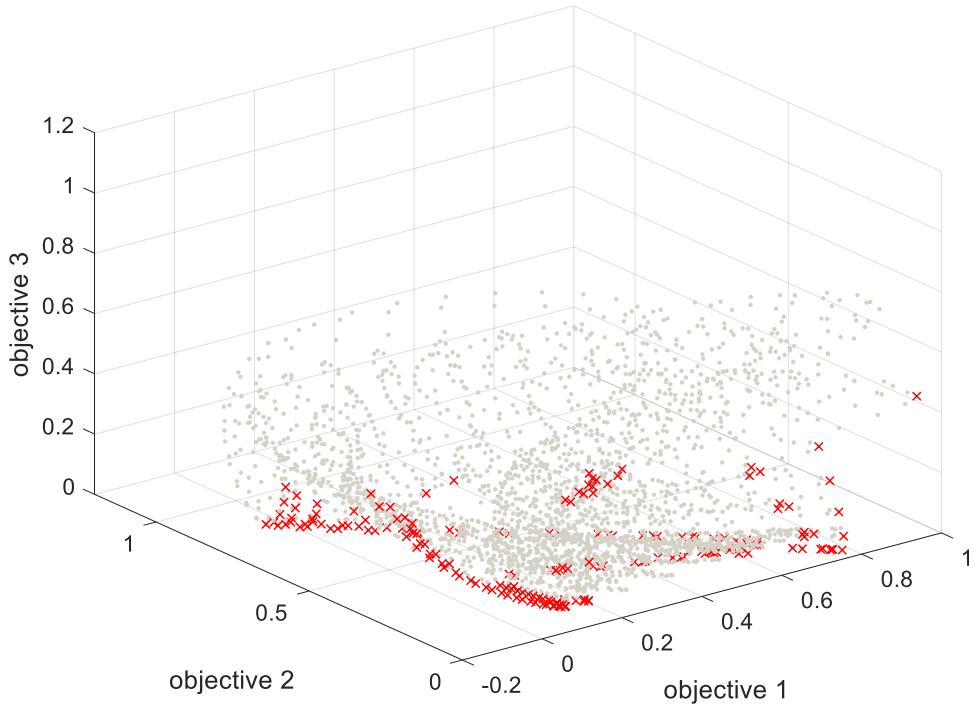
The problems in this section have three objective functions and each function has two variables. The maximum number of function evaluations is limited to 200. These test functions consist of random terms similar to those used in Section 3.3 in order to provide different response surfaces. For problems with higher dimensions and/or higher number of objectives, the Pareto points became sparser, hence locating all the Pareto points will be much harder.

Two examples have been provided below. Figures 6.15 and 6.17 show response surfaces of three objection functions in each example. In Figures 6.16 and 6.18, the grey dots are all possible (feasible) solutions of the problem, the red crosses describe the true Pareto front shown in Figures 6.16(a) and 6.18(a). In Figures 6.16(b) and 6.18(b) the orange squares indicate the sampling points in the objective space, while the orange triangles in Figures 6.16(c) and 6.18(c) mark the located Pareto points. As can be seen from the examples below, the proposed multi-objective optimisation approach has produced a reasonable number of diversified Pareto points within a limited number of function calls (200 sampling points).

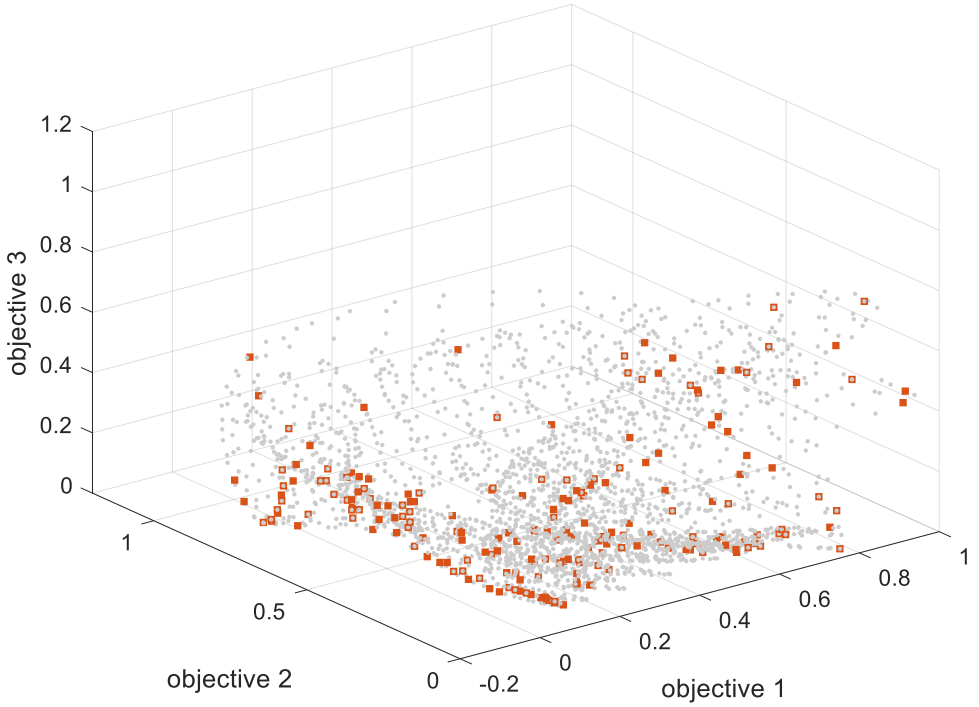


(a) (b) (c)

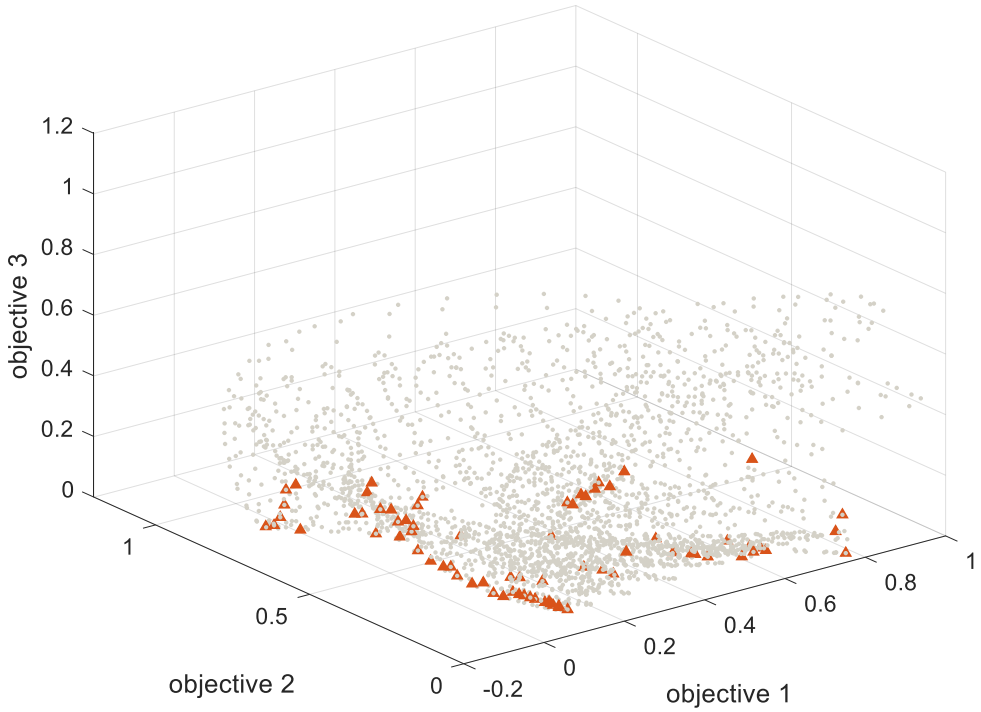
Figure 6.15. (a) Response surface of objective 1, (b) response surface of objective 2, (c) response surface of objective 3.


objective 3
1.2
1
0.8
0.6
0.4
0.2
0
objective 2
1
0.5
0
objective 1
-0.2
0
0.2
0.4
0.6
0.8
1

(a)

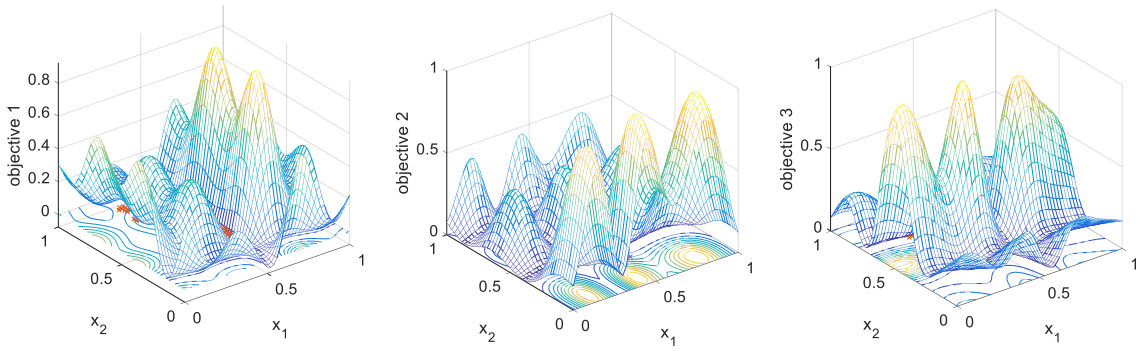

objective 3
1.2
1
0.8
0.6
0.4
0.2
0
objective 2
1
0.5
0
objective 1
-0.2
0
0.2
0.4
0.6
0.8
1

(b)



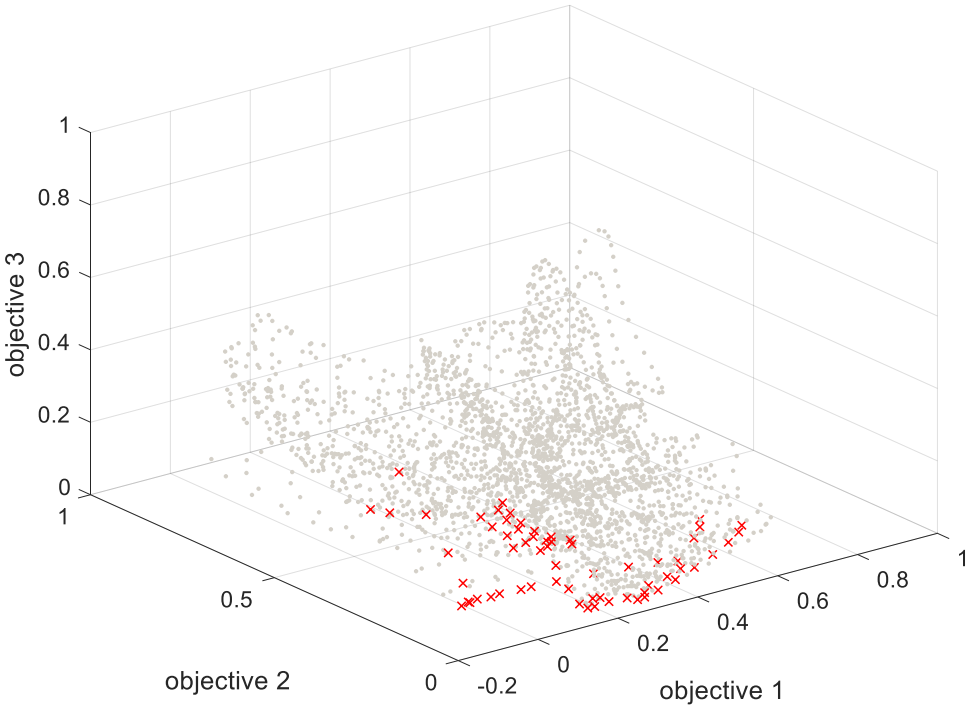
(c)

Figure 6.16. (a) Plot of the true Pareto front, (b) design points in the objective space, (c) located Pareto points.

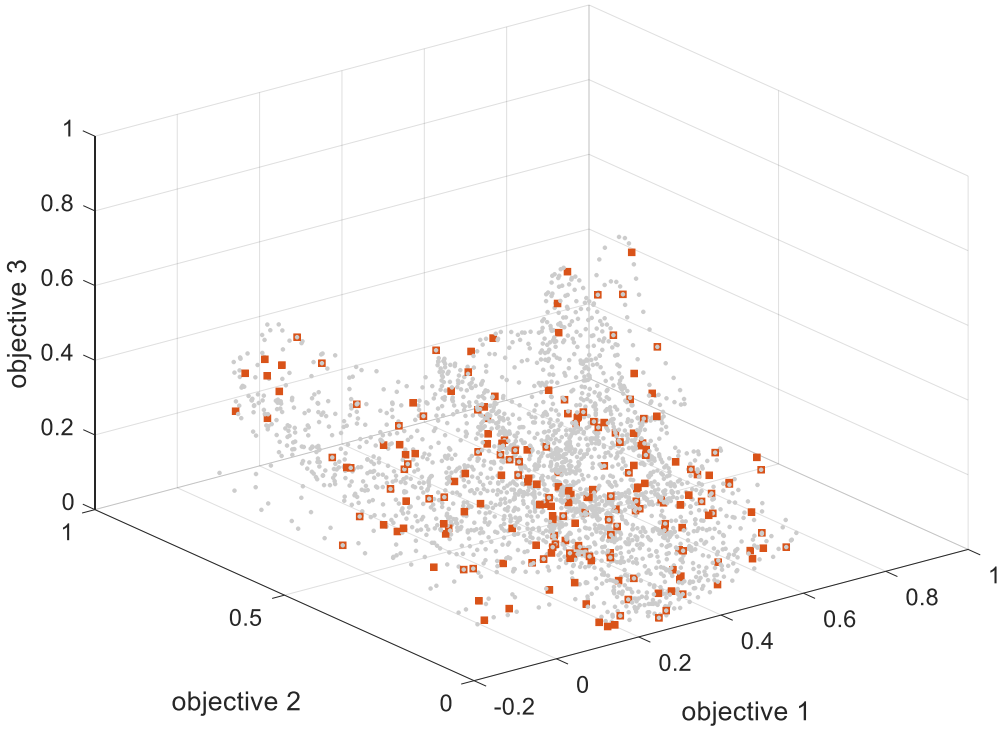


(a) (b) (c)

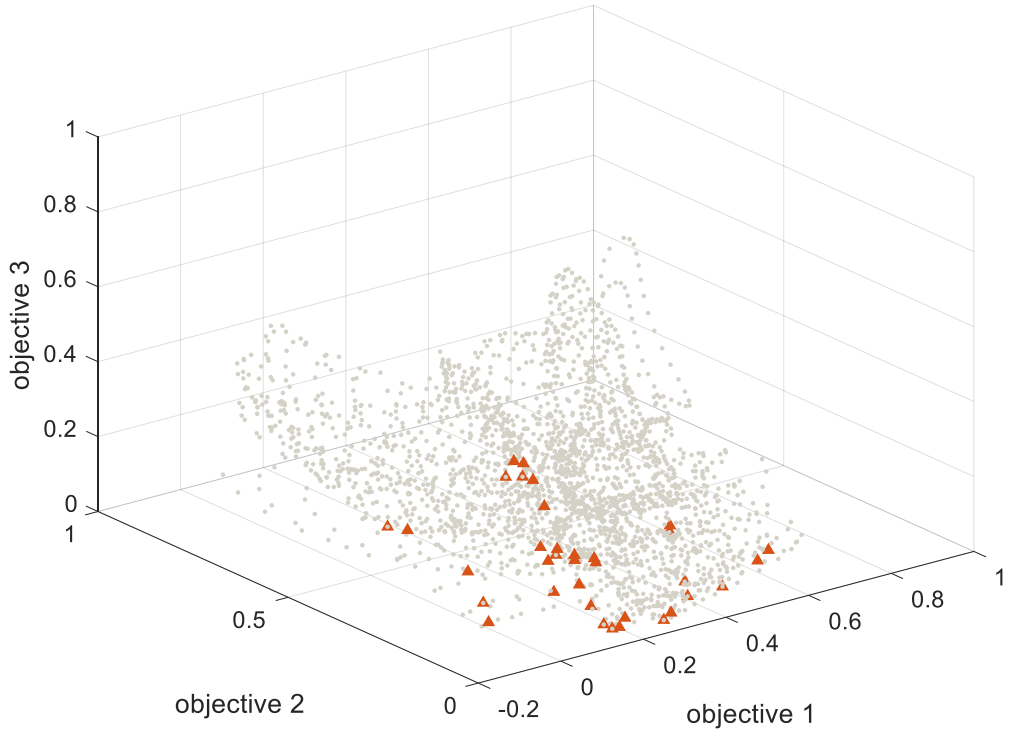
Figure 6.17. (a) Response surface of objective 1, (b) response surface of objective 2, (c) response surface of objective 3.


objective 3
1
0.8
0.6
0.4
0.2
0
objective 2
1
0.5
0
objective 1
-0.2
0
0.2
0.4
0.6
0.8
1

(a)


objective 3
1
0.8
0.6
0.4
0.2
0
objective 2
1
0.5
0
objective 1
-0.2
0
0.2
0.4
0.6
0.8
1

(b)



(c)

Figure 6.18. (a) Plot of the true Pareto front, (b) design points in the objective space, (c) located Pareto points.

6.5 ZDT benchmark problems

The method proposed in this chapter was tested on well-known ZDT multi-objective benchmark problems. The ZDT problems are a popular set of test functions proposed by Zitzler *et al.* (2000) [73] for benchmarking multi-objective optimisation algorithms. The ZDT test suit consists of 5 continuous multi-objective problems. The test function formulae are given in each section. Test results are plotted in Figures 6.19 to 6.23, the grey dots represent all the feasible solutions and the orange triangles are sampled design points. The result shows the sampling points distributed intensively around the Pareto region in the objective space, indicating the proposed kriging-based multi-objective optimisation solver provides a clear convergence tendency towards the Pareto front. In addition, the diversification and uniformity of the Pareto points have been maintained. For each problem, the dimension n has been set to $n = 2$, while the maximum number of function evaluations is limited to 200 and the number of initial sampling points equals to 10.

`

6.5.1 ZDT1

$$f_1(x) = x_1, f_2(x) = g(x)h(x) \tag{6.17}$$

where

$$g(x) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)}, h(x) = 1 - \sqrt{\frac{x_1}{g(x)}}, n = 2 \tag{6.18}$$

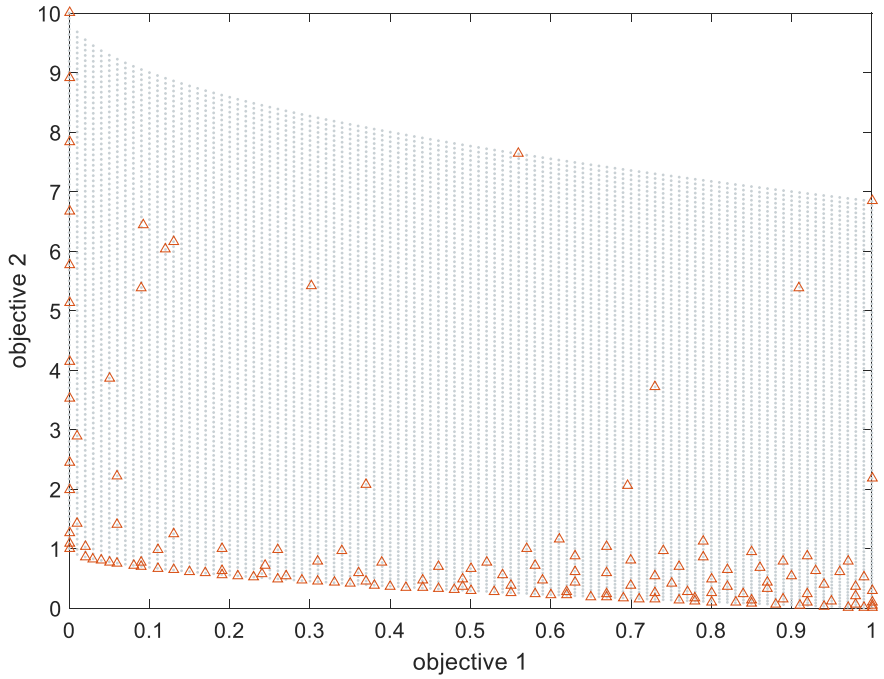


Figure 6.19. Design points in objective space (ZDT1).

6.5.2 ZDT2

$$f_1(x) = x_1, f_2(x) = g(x)h(x) \tag{6.17}$$

where

$$g(x) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)}, h(x) = 1 - \left(\frac{x_1}{g(x)}\right)^2, n = 2 \tag{6.18}$$

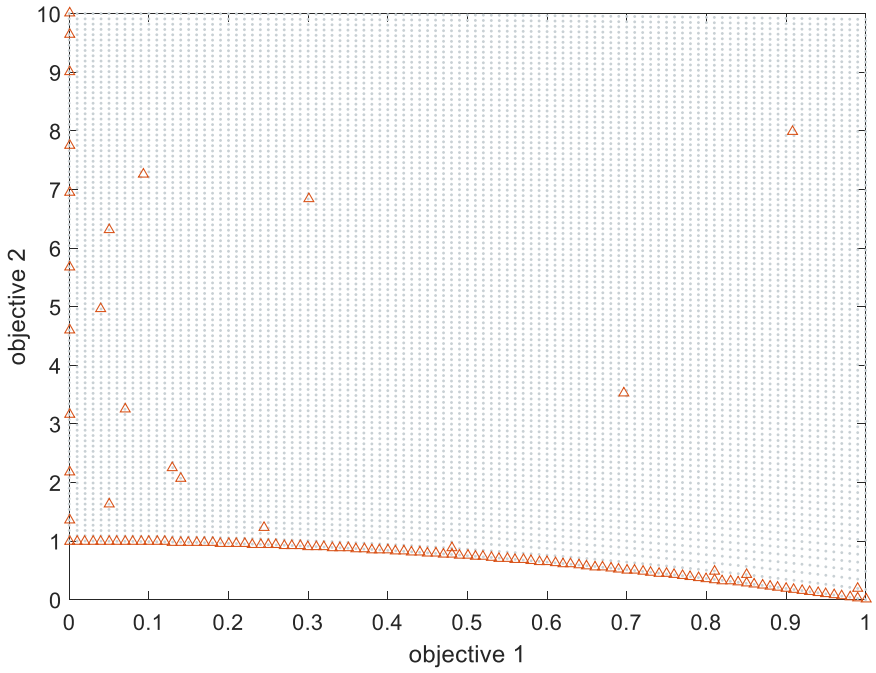


Figure 6.20. Design points in the objective space (ZDT2).

6.5.3 ZDT3

$$f_1(x) = x_1, f_2(x) = g(x)h(x) \tag{6.19}$$

where

$$g(x) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)}, h(x) = 1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\pi x_1), n = 2 \tag{6.20}$$

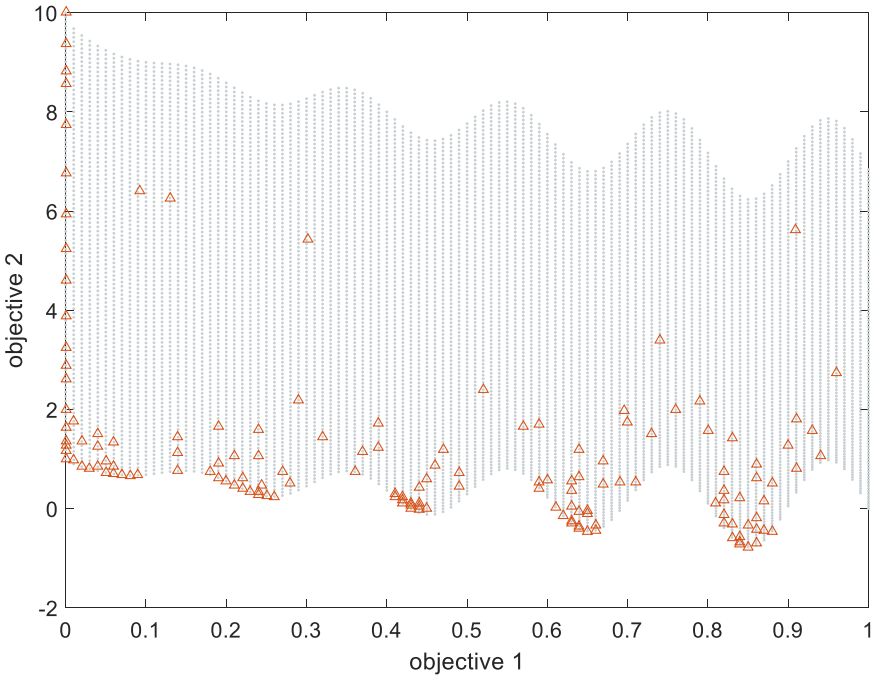


Figure 6.21. Design points in the objective space (ZDT3).

`

6.5.4 ZDT4

$$f_1(x) = x_1, f_2(x) = g(x)h(x) \tag{6.21}$$

where

$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[{x_i}^2 - 10\cos(4\pi x_i)], h(x) = 1 - \sqrt{\frac{x_1}{g(x)}}, n = 2 \tag{6.22}$$

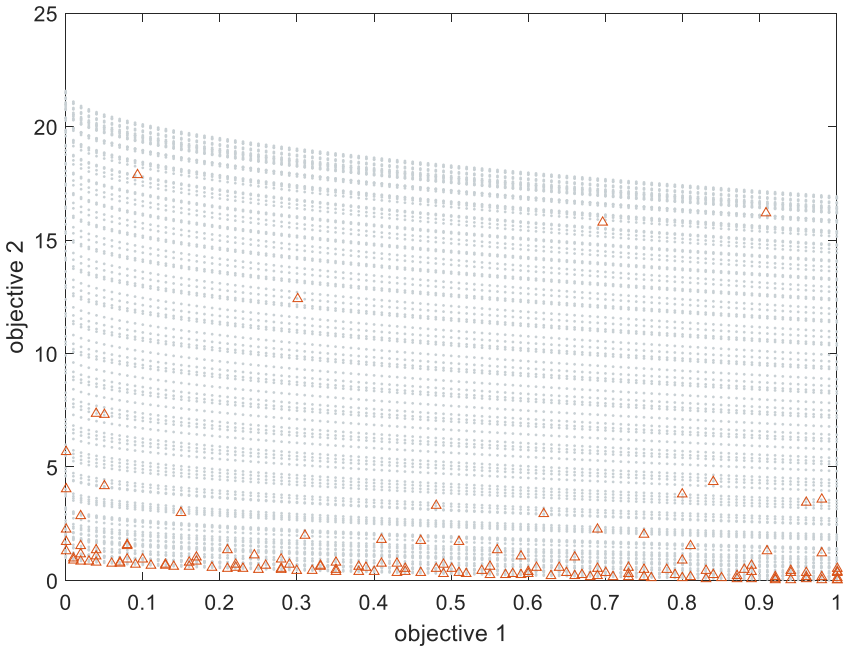


Figure 6.22. Design points in the objective space (ZDT4).

6.5.5 ZDT6

$$f_1(x) = 1 - e^{-4x_1}\sin^6(6\pi x_1), f_2(x) = g(x)h(x) \tag{6.23}$$

where

$$g(x) = 1 + 9\left(\frac{(\sum_{i=2}^{n} x_i)}{(n-1)}\right)^{\frac{1}{4}}, h(x) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2, n = 2 \tag{6.24}$$

`

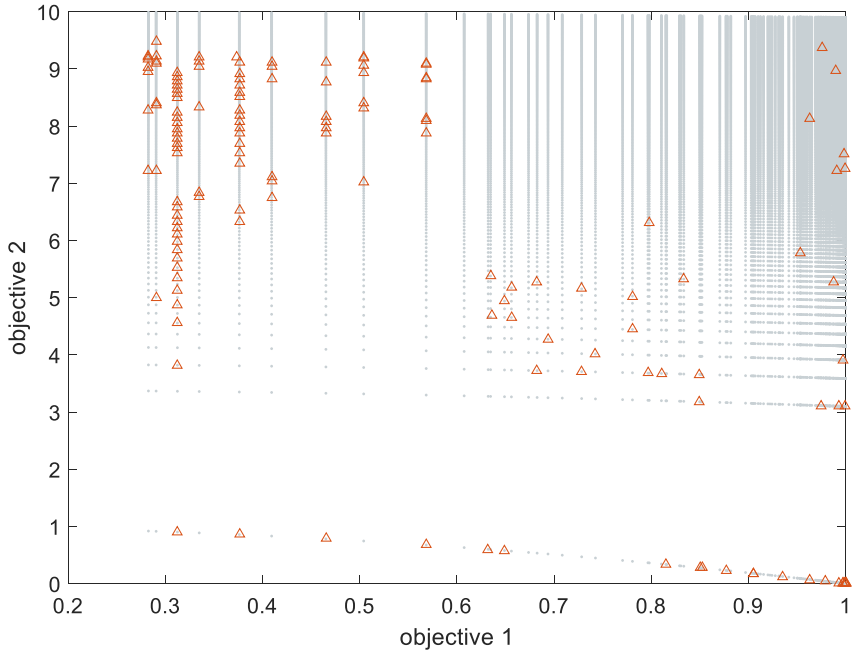


Figure 6.23. Design points in the objective space (ZDT4).

6.6 Conclusion

A novel approach to kriging-based multi objective optimisation was presented and details were discussed in this chapter. The proposed method was further tested against various random test function and the well-known ZDT benchmark test problems. Each of these test problems reproduces an interesting feature in multi-objective optimisation problems that may potentially cause difficulties for the optimisation algorithm. The proposed algorithm has successfully converged on the true Pareto front within a reasonable number of iterations. It has been shown that the solver is efficient and robust and provides a reasonably fast convergence rate toward the true Pareto front, while achieving both uniformity and divergence of the Pareto solution.

`

Chapter 7. Conclusion

7.1 Summary

This PhD program aimed at reviewing and improving the application of current kriging-based optimisation methods in robust and multi-objective design problems.

To comply with the research objectives, Chapter 1 reviewed surrogate-based optimisation, including the theory of the kriging model, initial sampling and infill sampling. Chapters 2 through 6 consist of studies on independent topics within the context of kriging-based optimisation; namely, infill sampling plan, robust optimisation, large datasets and multi-objective optimisation. Each chapter consists of a brief review of the topic and proposes new methods.

7.2 Contribution

- An improved infill sampling plan that incorporates a novel approach to dynamically balance exploration and exploitation is proposed. The method uses feedback information from the model quality to automate the parameter settings.
- A fast gradient-based infill criterion search method is proposed. The method significantly reduces the number of necessary infill criterion evaluations, consequently improving both the process' efficiency and accuracy.
- A two-step kriging-based robust optimisation approach for worst-case problems is proposed. By introducing an additional layer of optimisation during the infill process and a modified criterion for worst-case optimum allocation, the optimisation efficiency is increased.
- A dual kriging approach is introduced. By reallocating less important points to the secondary kriging, the covariance matrix size can potentially be reduced by 50%. Test results indicate that the proposed points reallocating criterion limits the impact of the removed points on the main kriging.
- A kriging method with points aggregation is proposed. The method locates the optimal model centre and aggregates a selection of points via optimisation, thus limiting the impact of information losses on interested search areas. This approach maintains constant memory usage at specified levels and therefore, enables the designer to intake a larger number of infill points.

`

- A kriging-based multi-objective optimisation method is proposed. The novel indicator localised probability of improvement ($LPoI$) addresses both the divergence and uniformity of the Pareto solution, while imposing a low computational cost. Test results reveal that the proposed method has a clear convergence tendency toward the Pareto front.

Referring back to the specific objectives formulated in Section 1.4 of the thesis it is therefore claimed that all of them have been met and some exceeded. While no single research program can completely answer the challenges of a specific problem, it can be argued that this work represents a step forward in efficient handling of kriging assisted design optimisation of electromagnetic devices.

7.3 Future work

The infill sampling and model-updating kriging-based optimisation is an iterative procedure that contributes to most time cost of the optimisation process, typically more than 99%. Currently, infill points are sampled sequentially. However, researchers may consider developing an infill criterion for multiple infill point intake in order to take advantage of parallel computational power. Undoubtedly, other programming issues also need to be solved, but the potential reward is significant. For a quad-core computer, parallel computation of basic arithmetics (i.e. all the threads are efficiently used and idle time is neglected) is approximately 3.4 times faster than single-core computation. In If designers can access more powerful computational resources, the algorithm's potential could be significant.

In addition to the difficulty in handling large datasets, kriging also has less flexibility in handling a wide range of problems: for example, problems with irregular or discontinued objective functions. Perhaps by designing an algorithm that maps the underlying problem to an unevenly scaled or modified search space, the kriging model could better fit the rescaled problem. For example, a response surface with a small complex area and a large flat area is mapped in a rescaled search space, where the complex area is enlarged while the flat area is narrowed. In this way, the complex region revives more "pixels" than other regions and both the sampling and model accuracy in the complex region could potentially improve.

There exist opinions which suggest that the domain-specific AI is the future in many industries, machine learning may also be introduced to the task of designing an optimisation algorithm. A simple design framework generates a large number of test problems that mimic the targeted practical problems and runs the optimisation algorithms multiple times on each of the test problems, while recording the detailed movement of the algorithm at the specific condition. These movements are then classified as either good movements or bad movements. Once the entire process is

`

complete, a computer-stored table of algorithm behaviours with past performance can be generated. Consequently, this table can be attached to the algorithm as a directional reference for similar problems. Optimisation and machine learning share many similarities in their logic, and developments from any of them may have potential applications in another. Hence, it would be interesting to combine these two topics.

The optimisation algorithms proposed in this thesis have been tested on various types of problems, including 1-D and 2-D test functions consist of multiple local optima, and practical benchmark problems T.E.A.M. 22 and T.E.A.M. 25. However the test did not expend to actual design problems, it would interesting and beneficial to implement the proposed algorithms to solve practical design optimisation problems.

Finally, it has to be recognised – as already mentioned in Section 1.4 – that in some practical design approaches the manufacturer does not necessarily aim at finding the global optimum and will be perfectly satisfied with a simple improvement to the current design, in particular to avoid expensive changes to manufacturing processes, tooling, materials, etc. It would indeed be an interesting follow up to this project how such approach could be phrased in mathematical terms so as to benefit from the kriging techniques developed in this thesis.

7.4 List of publications

The following list shows all papers published related to this thesis, including one digest accepted for a conference in Poland to take place in September 2017. Copies of the papers may be found in Appendix A.

Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2016. A Dual Kriging Approach With Improved Points Selection Algorithm for Memory Efficient Surrogate Optimization in Electromagnetics. *IEEE Transactions on Magnetics*, *52*(3), pp.1-4.

Li, Y., Rotaru, M. and Sykulski, J.K., 2016. Kriging based robust optimisation algorithm for minimax problems in electromagnetics. *Archives of Electrical Engineering*, *65*(4), pp.843-854.

Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2017. A Kriging-Based Optimization Approach for Large Data Sets Exploiting Points Aggregation Techniques. *IEEE Transactions on Magnetics*, *53*(6), pp.1-4.

Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2017. Localized probability of improvement for kriging based multi-objective optimization. *18th International Symposium on Electromagnetic Fields in Mechatronics, September 2017, Lodz, Poland*. (accepted)

`

`

Reference

[5] Rao, S.S. and Rao, S.S., 2009. *Engineering optimization: theory and practice*. John Wiley & Sons.

[6] Bäck, T., Fogel, D.B. and Michalewicz, Z., 1997. Handbook of evolutionary computation. *New York: Oxford*.

[7] Horst, R., Pardalos, P.M. and Van Thoai, N., 2000. *Introduction to global optimization*. Springer Science & Business Media.

[8] Floudas, C.A. and Gounaris, C.E., 2009. A review of recent advances in global optimization. *Journal of Global Optimization*, *45*(1), pp.3-38.

[9] Ashlock, D., 2006. *Evolutionary computation for modeling and optimization*. Springer Science & Business Media.

[10] Xiao, S., Rotaru, M. and Sykulski, J.K., 2012. Exploration versus exploitation using kriging surrogate modelling in electromagnetic design. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, *31*(5), pp.1541-1551.

[11] Xiao, S., Rotaru, M. and Sykulski, J.K., 2013. Adaptive weighted expected improvement with rewards approach in kriging assisted electromagnetic design. *IEEE Transactions on Magnetics*, *49*(5), pp.2057-2060.

[12] Xiao, S., Rotaru, M. and Sykulski, J.K., 2011. Exploration versus Exploitation Using Kriging Surrogate Modelling in Electromagnetic Design. In, *ISEF 2011 - XV International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering*, OS.1.6.

[13] Song, X., Li, Y., Rotaru, M. and Sykulski, J.K., 2014, March. Correlation matrices in kriging assisted optimisation of electromagnetic devices. In *IET Conference Proceedings*. The Institution of Engineering & Technology.

[14] Xiao, S., Rotaru, M. and Sykulski, J.K., 2013. Robust global optimization of electromagnetic designs utilizing gradient indices and kriging. In,*COMPUMAG 2013, 19th International Conference on the Computation of Electromagnetic Fields, Budapest, HU, 30 Jun - 04 Jul 2013.* 2pp.

[15] Xiao, S., Rotaru, M. and Sykulski, J.K., 2013. Strategies for balancing exploration and exploitation in electromagnetic optimisation. *COMPEL-The international*

`

journal for computation and mathematics in electrical and electronic engineering, *32*(4), pp.1176-1188.

[16] Xiao, S., Rotaru, M. and Sykulski, J.K., 2012. Strategies for balancing exploration and exploitation in electromagnetic optimisation. *XXII Symposium on Electromagnetic Phenomena in Nonlinear Circuits*, Pula, HR, 26 - 29 Jun 2012. 2pp, 1-2.

[17] Stephenson, J., Gallagher, K. and Holmes, C.C., 2004. Beyond kriging: dealing with discontinuous spatial data fields using adaptive prior information and Bayesian partition modelling. *Geological Society, London, Special Publications*, *239*(1), pp.195-209.

[18] Ankenman, B., Nelson, B.L. and Staum, J., 2010. Stochastic kriging for simulation metamodeling. *Operations research*, *58*(2), pp.371-382.

[19] Denison, D.G., 2002. *Bayesian methods for nonlinear classification and regression* (Vol. 386). John Wiley & Sons.

[20] Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P., 1989. Design and analysis of computer experiments. *Statistical science*, pp.409-423.

[21] Lghali, A., 2012. Surrogate Based Optimization Using Kriging Based Approximation. *VU BA paper*.

[22] Hoeting, J.A., Davis, R.A., Merton, A.A. and Thompson, S.E., 2006. Model selection for geostatistical models. *Ecological Applications*, *16*(1), pp.87-98.

[23] Song, H., Choi, K.K. and Lamb, D., 2013, May. A study on improving the accuracy of kriging models by using correlation model/mean structure selection and penalized log-likelihood function. In *10th world congress on structural and multidisciplinary optimization. Florida, Orlando*.

[24] Nielsen, H.B., Lophaven, S.N. and Søndergaard, J., 2002. DACE, a MATLAB Kriging toolbox. *Informatics and mathematical modelling. Lyngby–Denmark: Technical University of Denmark, DTU*.

[25] Xie, W., Nelson, B. and Staum, J., 2010, December. The influence of correlation functions on stochastic kriging metamodels. In *Proceedings of the Winter Simulation Conference* (pp. 1067-1078). Winter Simulation Conference.

[26] Giunta, A., Wojtkiewicz, S. and Eldred, M., 2003, January. Overview of modern design of experiments methods for computational simulations. In *41st Aerospace Sciences Meeting and Exhibit* (p. 649).

`

[27] Liu, J., Han, Z.H. and Song, W.P., 2012, September. Comparison of infill sampling criteria in kriging-based aerodynamic optimization. In *28th Congress of the International Council of the Aeronautical Sciences* (pp. 23-28).

[28] Han, Z.H. and Zhang, K.S., 2012. *Surrogate-based optimization*. Rijeka: INTECH Open Access Publisher.

[29] Niederreiter, H., 1992. *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied mathematics.

[30] Statistics and Machine Learning Toolbox User's Guide

[31] Jones, D.R., Schonlau, M. and Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, *13*(4), pp.455-492.

[32] Sasena, M.J., Papalambros, P. and Goovaerts, P., 2002. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering optimization*, *34*(3), pp.263-278.

[33] Forrester, A.I. and Keane, A.J., 2009. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, *45*(1), pp.50-79.

[34] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. *science*, *220*(4598), pp.671-680.

[35] Taguchi, G., 1986. *Introduction to quality engineering: designing quality into products and processes*.

[36] Ferreira, J., Gaspar-Cunha, A., Fonseca, C.M. and Covas, J.A., 2008. *Evolutionary multi-objective robust optimization*. INTECH Open Access Publisher.

[37] Beyer, H.G. and Sendhoff, B., 2007. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, *196*(33), pp.3190-3218.

[38] Schuëller, G.I. and Jensen, H.A., 2008. Computational methods in optimization considering uncertainties–an overview. *Computer Methods in Applied Mechanics and Engineering*, *198*(1), pp.2-13.

[39] Xiao, S., Li, Y., Rotaru, M. and Sykulski, J.K., 2015. Six sigma quality approach to robust optimization. *IEEE Transactions on Magnetics*, *51*(3), pp.1-4.

[40] Xiao, S., Li, Y., Rotaru, M. and Sykulski, J.K., 2014. Considerations of uncertainty in robust optimisation of electromagnetic devices. *International Journal of Applied Electromagnetics and Mechanics*, *46*(2), pp.427-436.

`

[41] Lee, K.H. and Kang, D.H., 2006. A robust optimization using the statistics based on kriging metamodel. *Journal of Mechanical Science and Technology*, *20*(8), pp.1169-1182.

[42] Marzat, J., Walter, E. and Piet-Lahanier, H., 2013. Worst-case global optimization of black-box functions through Kriging and relaxation. *Journal of Global Optimization*, *55*(4), p.707.

[43] ur Rehman, S., Langelaar, M. and van Keulen, F., 2014. Efficient Kriging-based robust optimization of unconstrained problems. *Journal of Computational Science*, *5*(6), pp.872-881.

[44] http://www.compumag.org/jsite/T.E.A.M..html

[45] Alotto, P.G., Baumgartner, U., Freschi, F., Jaindl, M., Kostinger, A., Magele, Ch., Renhart, W. and Repetto, M., *SMES Optimisation Benchmark: T.E.A.M. Workshop Problem 22*.

[46] Holland, J.H., 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

[47] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. *science*, *220*(4598), pp.671-680.

[48] Hu, N., 1992. Tabu search method with random moves for globally optimal design. *International Journal for Numerical Methods in Engineering*, *35*(5), pp.1055-1070.

[49] Santner, T.J., Williams, B.J. and Notz, W.I., 2013. *The design and analysis of computer experiments*. Springer Science & Business Media.

[50] Hajji, O., Brisset, S. and Brochet, P., 2004. A new tabu search method for optimization with continuous parameters. *IEEE transactions on magnetics*, *40*(2), pp.1184-1187.

[51] Li, Y., Xiao, S., Rotaru, M. and Sykulski, J.K., 2016. A Dual Kriging Approach With Improved Points Selection Algorithm for Memory Efficient Surrogate Optimization in Electromagnetics. *IEEE Transactions on Magnetics*, *52*(3), pp.1-4.

[52] Romary, T., 2013. Incomplete Cholesky decomposition for the kriging of large datasets. *spatial statistics*, *5*, pp.85-99.

[53] Cressie, N. and Johannesson, G., 2008. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(1), pp.209-226.

`

[54] Xiao, S., Rotaru, M. and Sykulski, J.K., 2015. Correlation matrices in kriging assisted optimisation of electromagnetic devices. *IET Science, Measurement & Technology*, *9*(2), pp.189-196.

[55] Haas, T.C., 1990. Lognormal and moving window methods of estimating acid deposition. *Journal of the American Statistical Association*, *85*(412), pp.950-963.

[56] Furrer, R., Genton, M.G. and Nychka, D., 2006. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, *15*(3), pp.502-523.

[57] Murtagh, F., 1983. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, *26*(4), pp.354-359.

[58] dos Santos Coelho, L. and Alotto, P., 2008. Global optimization of electromagnetic devices using an exponential quantum-behaved particle swarm optimizer. *IEEE Transactions on Magnetics*, *44*(6), pp.1074-1077.

[59] Hogg, T. and Portnov, D., 2000. Quantum optimization. *Information Sciences*, *128*(3), pp.181-197.

[60] Sun, J., Feng, B. and Xu, W., 2004, June. Particle swarm optimization with particles having quantum behavior. In *Evolutionary Computation, 2004. CEC2004. Congress on* (Vol. 1, pp. 325-331). IEEE.

[61] Krohling, R.A. and dos Santos Coelho, L., 2006, July. PSO-E: Particle swarm with exponential distribution. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 1428-1433). IEEE.

[62] Alotto, P., Kuntsevitch, A.V., Magele, C., Molinari, G., Paul, C., Preis, K., Repetto, M. and Richter, K.R., 1996. Multiobjective optimization in magnetostatics: a proposal for benchmark problems. *IEEE Transactions on Magnetics*, *32*(3), pp.1238-1241.

[63] Stadler, W., 1988. Fundamentals of multicriteria optimization. In *Multicriteria Optimization in Engineering and in the Sciences* (pp. 1-25). Springer US.

[64] Marler, R.T. and Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, *26*(6), pp.369-395.

[65] Schaffer, J.D., 1985, July. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international Conference on Genetic Algorithms* (pp. 93-100). L. Erlbaum Associates Inc.

`

[66] Srinivas, N. and Deb, K., 1994. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), pp.221-248.

[67] rey Horn, J., Nafpliotis, N. and Goldberg, D.E., 1993. Multiobjective optimization using the niched pareto genetic algorithm. *IlliGAL report*, (93005), pp.61801-2296.

[68] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.A.M.T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), pp.182-197.

[69] Parsopoulos, K.E. and Vrahatis, M.N., 2002, March. Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM symposium on Applied computing* (pp. 603-607). ACM.

[70] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M. and Da Fonseca, V.G., 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, *7*(2), pp.117-132.

[71] Hansen, M.P. and Jaszkiewicz, A., 1998. *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical Universityof Denmark.

[72] Zitzler, E. and Thiele, L., 1998, September. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International Conference on Parallel Problem Solving from Nature* (pp. 292-301). Springer Berlin Heidelberg.

[73] Brockhoff, D., Wagner, T. and Trautmann, H., 2015. R2 indicator-based multiobjective search. *Evolutionary Computation*, *23*(3), pp.369-395.

[74] Mansour, I.B. and Alaya, I., 2015. Indicator based ant colony optimization for multi-objective knapsack problem. *Procedia Computer Science*, *60*, pp.448-457.

[75] Luo, C., Shimoyama, K. and Obayashi, S., 2015. A study on many-objective optimization using the kriging-surrogate-based evolutionary algorithm maximizing expected hypervolume improvement. *Mathematical Problems in Engineering*, *2015*.

[76] Martínez-Frutos, J. and Herrero-Pérez, D., 2016. Kriging-based infill sampling criterion for constraint handling in multi-objective optimization. *Journal of Global Optimization*, *64*(1), pp.97-115.

[77] Zitzler, E., Deb, K. and Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, *8*(2), pp.173-195.

`

[78] Bradstreet, L., 2011. *The hypervolume indicator for multi-objective optimisation: calculation and use*. University of Western Australia.

[79] Bringmann, K. and Friedrich, T., 2010. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry*, *43*(6-7), pp.601-610.

[80] Bringmann, K. and Friedrich, T., 2009, April. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 6-20). Springer Berlin Heidelberg.

[81] Bringmann, K. and Friedrich, T., 2010. An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, *18*(3), pp.383-402.

[82] Lacour, R., Klamroth, K. and Fonseca, C.M., 2017. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, *79*, pp.347-360.

[83] Knowles, J. and Corne, D., 2002, May. On metrics for comparing nondominated sets. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* (Vol. 1, pp. 711-716). IEEE.

`

`

Appendix A

A Dual Kriging Approach with Improved Points Selection Algorithm for Memory Efficient Surrogate Optimization in Electromagnetics

Yinjiang Li, Song Xiao, Mihai Rotaru, and Jan K. Sykulski, *Fellow, IEEE*

Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom

The paper introduces a new approach to kriging surrogate model sampling points allocation. By introducing a second (dual) kriging during the model construction the existing sampling points are reallocated to reduce overall memory requirements. Moreover, a new algorithm is proposed for selecting the position of the next sampling point by utilizing a modified Expected Improvement criterion.

***Index Terms*— Kriging, global optimization, surrogate modelling, large datasets.**

I. Introduction

KRIGING offers significant advantages in computationally expensive optimization as a number of necessary objective function calls may be reduced. This matters in particular when each call entails time consuming simulations, such as frequently encountered in the design of electromagnetic devices [1]. Large data sets, however, tend to be produced by correlation matrices which arise when kriging models are produced and the amount of storage required is usually proportional to n^2 [2], where n is the number of sample points; this problem may become acute in the case of multi-parameter optimization when performed on small computers with limited memory [3], [4].

II. A Modified EI Sampling Criterion

Expected Improvement (EI) [5] is commonly used to guide the process of selecting the next point for evaluation (often with modifications [1]). The challenge is to balance exploitation and exploration in order to avoid the kriging model being trapped in a local optimum; moreover, the quality of the kriging prediction of the shape of the objective function may also be important in the context of the robustness of the design. In this paper we suggest a modification to the standard EI criterion with the aim to spread the 'infill' (new sampling) points more efficiently throughout the design space. Consider a simple illustration in Fig. 1 where the dotted line is the actual objective function. The range has been normalized between 0 and 1 while the values of the objective function have no actual meaning in this example.

The proposed sampling criterion calculates EI while taking the estimated error (the 'Mean Square Error' MSE [1]) between known sampling points into consideration

$$sampling\ criterion = \max\{EI\} \times \mathrm{MSE} \times weight + \max\{\mathrm{MSE}\} \times EI \qquad (1)$$

whereas scaling has been applied to account for different values of components and thus normalize the results. Moreover, a weight is added to the estimated error. The estimated error is provided by the kriging predictor together with the predicted value at any given point.

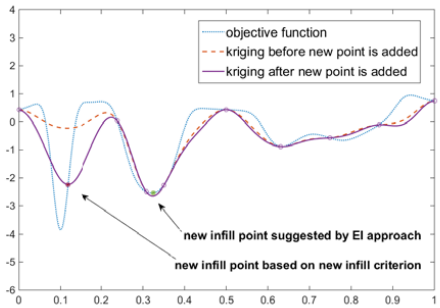


Fig. 1. The kriging model before and after a new point has been added.

The weight term is the ratio of the exponentially weighted standard deviation (between infill points and their previously predicted value) average and the uniformly weighted standard deviation average. This value decreases as the optimization process continues and the model quality increases. The exponentially weighted standard deviation average at the current iteration is calculated using formula:

$$ESDA = \frac{sd_1 + (1-\alpha)sd_2 + (1-\alpha)^2 sd_3 + \cdots}{1 + (1-\alpha) + (1-\alpha)^2 + \cdots} \qquad (2)$$

where α determines the weight on each standard deviation term, and $0 < \alpha < 1$; sd is the current standard deviation at a point.

The classical EI criterion itself would advocate exploitation of the area close to the recently simulated minimum (which in reality is a local minimum) and create a new point at x=0.347 (see Fig. 1), whereas this could be counterproductive during the exploration stage. The modified criterion, however, proposes more exploration and positions the new sampling point elsewhere with better chance of capturing the global minimum, as illustrated. Moreover, it is important in the context of robust optimization to predict not just the position of the optimum but also the shape of the objective function near the optimum.

Manuscript received June 23, 2015; revised ??? ??, 2015 and ??? ??, 2015; accepted ??? ??, 2015. Date of publication ??? ??, 2015; date of current version ??? ??, 2015. Corresponding author: J. K. Sykulski (e-mail: jks@soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier (inserted by IEEE).

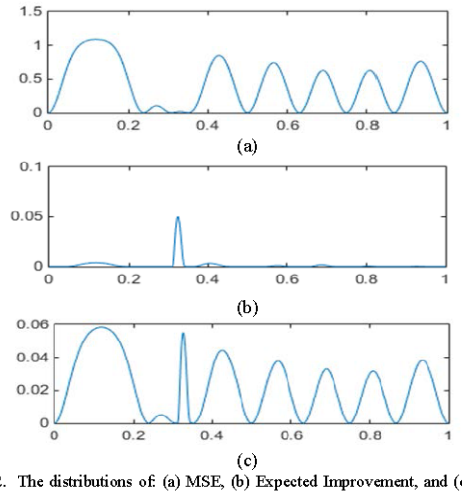


Fig. 2. The distributions of: (a) MSE, (b) Expected Improvement, and (c) the resultant sampling criterion for the example test function.

III. A Dual Kriging Approach

The main drawback of the kriging approach is the need for creating correlation matrices which – especially in the case of multi-parameter problems – may become very large and must be handled carefully. This was discussed in [4] where a possible solution was offered. Here we suggest a complementary approach resulting from an argument that once the surrogate model is advanced and the shape of the objective function is reasonably accurately predicted we really only need some sampling points, especially those close to the areas considered as potentially attractive. Thus, as the total number of sampling points increases and we are getting to the memory limit of the computer, in order to avoid computationally time consuming 'memory management' (e.g. page swapping) we may instead 'remove' some of the less attractive points in an attempt to keep the total number of points constant, or increasing slowly, while the removed points may be used to create a 'dual' kriging model. At the same time it should be noted that – as shown in Fig. 3 – the memory saving is the biggest when we operate at roughly between 20% and 30% of the reduced number of points (shaded area). For example, at 20% (that is the original kriging model preserving 80% of points) the memory saving is 32%.

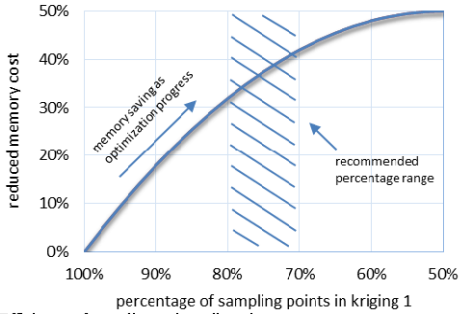


Fig. 3. Efficiency of sampling points allocation.

The simple criterion for removing a point is related to the triangular area formed by connecting this point and the two neighbors, as illustrated in Fig. 4 (here x=0.75 can be removed). Thus the points with the smallest associated area are removed and in fact used to create a dual kriging model. The current optimum is assigned a large area therefore will not be removed.

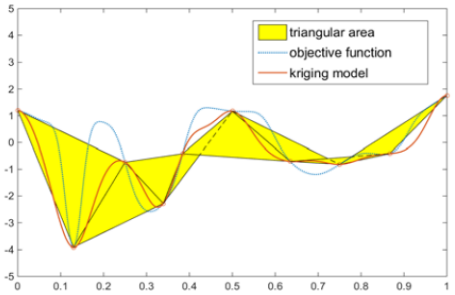


Fig. 4. Point removal criterion plotted as shaded triangular areas.

Consider the test function of Fig. 1 after 13 iterations and its kriging prediction as shown in Fig. 5 (note that iterations have not completed yet). At this stage six points have been removed from the main kriging to create the secondary kriging. The modified model now contains fewer but more important points, as illustrated by Figs. 5 and 6. In the example a saving of 49% of memory has been realized by reducing the correlation matrix.

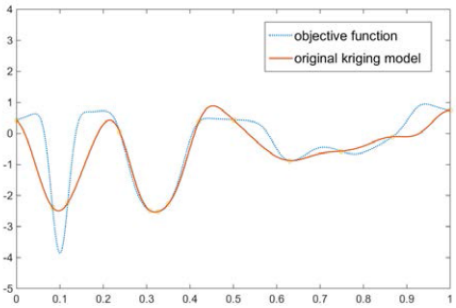


Fig. 5. A single kriging model with 13 sampling points (iterations incomplete).

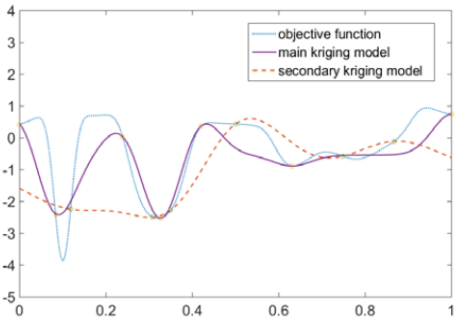


Fig. 6. The main kriging model (7 sampling points) and the secondary kriging model (6 sampling points).

IV. EI Sampling Criterion aided by Dual Kriging

While applying dual kriging could potentially save computer memory demands by as much as 50% compared to a single kriging approach, for the same number of sampling points, its major challenge lies in the selection of the location of the new infill point. If only the main kriging model were to be considered, there would be a risk of a new location to be at or close to a point which has just been removed. In order to inhibit such an unwelcome scenario, both kriging models and thus a modified MSE defined by a product of the principal and dual MSEs are used; consequently the EI criterion uses all points and no information is lost. This is illustrated by Fig. 7 showing the 15th iteration, with the point adding/removal process triggered after the eighth point had been added; thus there are 8 points in the main ('1') and 7 in the dual ('2') kriging models. The modified sampling criterion based on the combined MSE offers a better prediction than the original single kriging approach.

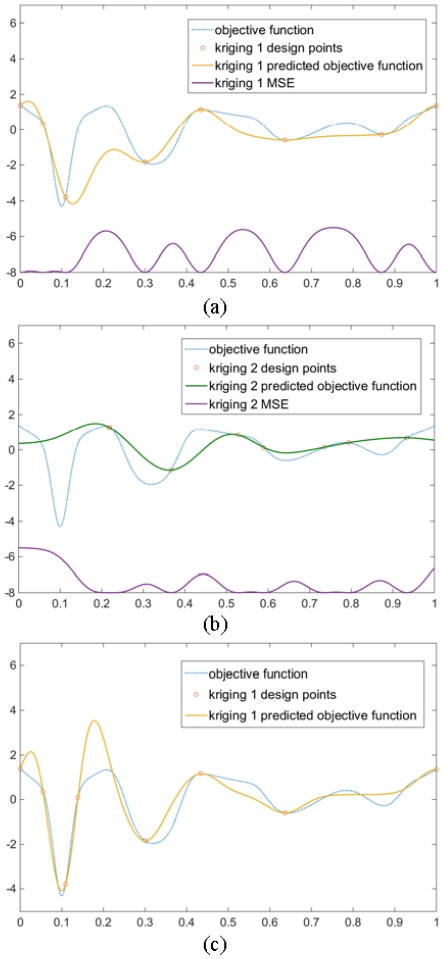


Fig. 7. (a) Main kriging and MSE, (b) dual kriging and MSE, (c) predicted function after a 'new' infill point is added and the 'old' point removed.

V. More on the Selection of Infill Points

The use of Expected Improvement (EI) as a criterion for selecting infill points is widely accepted in surrogate modelling, despite some drawbacks as reported for example in [12]; further modifications and improvements have been suggested. But there is one particular difficulty which is rarely mentioned but worth emphasizing. In order to locate reliably the point with maximum EI, all points within the design space need to be considered, requiring calculation of the predicted objective function and the corresponding MSE at all these points. This may seem straightforward for one dimensional problems, but could create another case of 'combinatorial explosion' in more practical multi-dimensional problems. Take four dimensions, for example, and a step of 1/100 in each direction: this would require 10^8 calculations to find the maximum EI, which even for very fast surrogate models would create a computational bother. A large step of say 1/10 would be easy to handle but unlikely to capture the actual optimum. In Fig. 8 a case is depicted where even a relatively small step of 1/50 might result in the algorithm missing the largest EI and hence the global optimum at x=0.131 would be overlooked.

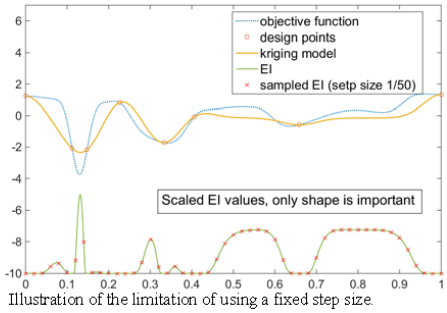


Fig. 8. Illustration of the limitation of using a fixed step size.

As an alternative to the exhaustive search using a predefined step size, the task of finding the maximum sampling criterion value could be treated as a small optimization problem in its own right. Any optimization method could be employed for that purpose and would effectively eliminate the problem of the step size mentioned above. A genetic algorithm has been utilized to search for the infill criterion function in the next section.

Finally, the algorithms described so far can be supplemented by efficient local routines, e.g. a new point could be added between any two adjacent existing points and the location of the maximum EI estimated using a local gradient-based method.

VI. Team Problem 25

To illustrate the proposed optimization methodology in the context of electromagnetic design, the TEAM problem 25 has been studied [13], which is a die press with an electromagnet for the orientation of magnetic powder, used to produce an anisotropic permanent magnet. The objective of this problem is to optimize the shape to minimize a particular objective. Each function evaluation requires a full finite element (FE) solution of a non-linear problem, which is computationally inefficient if used in combination with any optimization method, especially if extensive design space exploration is required. This is therefore a very appropriate practical case to be studied.

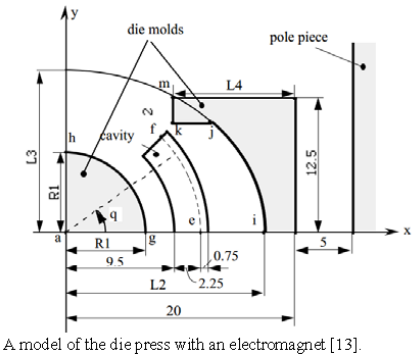


Fig. 9. A model of the die press with an electromagnet [13].

In Fig. 9, R1, L2, L3 and L4 are the design parameters to be optimized so that the objective function W is minimized

$$W = \sum_{i=1}^{n} \left\{ \left(B_{xip} - B_{xio}\right)^2 + \left(B_{yip} - B_{yio}\right)^2 \right\} \qquad (3)$$

where B_x and B_y are the x and y components of magnetic flux density at points along the curve e-f, while subscripts p and o denote the calculated and desired values, respectively. The constraints are listed in Table I and results in Table II.

TABLE I
PARAMETER CONSTRAINTS

Variable	Lower-boundary (mm)	Upper-boundary (mm)
R1	5	9.4
L2	12.6	18
L3	14	45
L4	4	19

TABLE II
COMPARISON OF PERFORMANCE

Algorithms	Optimal value (10^{-4})	R1 (mm)	L2 (mm)	L3 (mm)	L4 (mm)	No of function calls
GA	2.686	7.2996	14.174	14.001	14.326	3421
SA	1.622	7.2252	14.322	14.110	14.306	2145
HuTS	0.500	7.3780	14.613	14.371	14.204	1580
UTS	1.050	7.5487	14.908	14.506	14.416	931
NTS	0.648	7.4337	14.732	14.428	14.237	575
Kriging EI	0.452	7.2	14.1	14	14.5	265
Kriging AWEI	0.412	7.2	14	14	14.5	214
Dual kriging	0.323	7.007	13.891	14.035	14.270	242

Genetic algorithm GA [6], Stimulated Annealing SA [7], Tabu Search HuTS [8]; Universal Tabu search [9]; New Tabu Search NTS [10]; Kriging EI and Kriging AWEI (with specified step size) [11];

The first five results in Table II are taken from literature, the first two kriging values from our previous publications, the dual kriging approach is shown in the bottom row. Generally kriging is performing significantly better than other methods in terms of a better optimum, but primarily because of much reduced computing times, measured in the number of necessary FE calculations. The dual kriging required marginally more iterations, but has produced a slightly better result. In this sense all kriging models are similar and superior to other methods.

The dual kriging algorithm was triggered after 120 FE calls, with one sampling point removed when a new one was added in subsequent iterations. The sizes of the covariance matrix in the kriging model for the 'standard' AWEI [11] routine and the new dual algorithm are compared in Fig. 10 as optimization progresses. Memory savings on this occasion reached 36%.

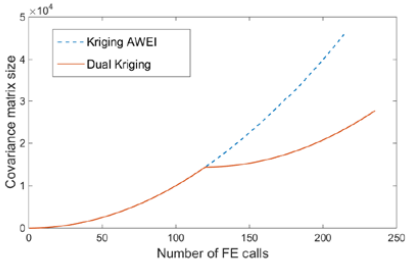


Fig. 10. Memory requirements for the covariance matrix at each iteration.

VII. CONCLUSION

A modified infill sampling criterion is proposed which can be applied to both the single and dual kriging methods. A simple but efficient automatic exploration vs exploitation adjustment based on model quality is developed. The limitations of an exhaustive search are noted and a global optimization aided EI search algorithm introduced. The dual kriging method is verified against a test function and an electromagnetic TEAM problem 25, with a 36% reduction in the covariance matrix size.

REFERENCES

[1] S. Xiao, M. Rotaru, and J. K. Sykulski, "Adaptive Weighted Expected Improvement with Rewards Approach in Kriging Assisted Electromagnetic Design," *IEEE Trans. Magn.*, vol. 49, no. 5, pp. 2057–2060, 2013.

[2] T. Romary, "Incomplete Cholesky decomposition for the kriging of large datasets," *GeoENV 2012*, Spain, p. 259, 2012.

[3] N. Cressie and G. Johannesson, "Fixed rank kriging for very large spatial data sets," *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 70, no. 1, pp. 209–226, 2007.

[4] S. Xiao, M. Rotaru, and J. K. Sykulski, "Correlation matrices in kriging assisted optimisation of electromagnetic devices," *IET Science, Measurement and Technology*, vol. 9, no. 2, pp. 189–196, 2015.

[5] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Cambridge, MA, USA: MIT Press, 1975.

[7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[8] N. Hu, "Tabu search method with random moves for globally optimal design," *Int. J. Num. Methods Eng.*, vol. 35, no. 5, pp. 1055–1070, 1992.

[9] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, New York, USA, Springer-Verlag, 2003.

[10] O. Hajji, S. Brisset, and P. Brochet, "A new Tabu search method for optimization with continuous parameters," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1184–1187, Mar. 2004.

[11] S. Xiao, M. Rotaru, and J. K. Sykulski, "Six Sigma Quality Approach to Robust Optimization," *IEEE Trans. Magn.*, vol. 51, no. 3, 2015.

[12] A. I. J. Forrester and D. R. Jones, "Global optimization of deceptive functions with sparse sampling," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, Canada, Sept. 2008.

[13] N. Takahashi, K. Ebihara, K. Yoshida, T. Nakata, K. Ohashi, and K. Miyata, "Investigation of simulated annealing method and its application to optimal design of die mold for orientation of magnetic powder," *IEEE Trans. Magn.*, vol. 32, no. 3, pp. 1210–1213, 1996.

ARCHIVES OF ELECTRICAL ENGINEERING VOL. XX(Y), pp. XXX-YYY (YEAR)

DOI 10.1515/aee-YEAR-00XY

Kriging based robust optimisation algorithm for minimax problems in electromagnetics

YINJIANG LI, MIHAI ROTARU, JAN K. SYKULSKI

Electronics and Computer Science
University of Southampton
Southampton, United Kingdom
e-mail: jks@soton.ac.uk

(Received: DD.MM.YEAR, revised: DD.MM.YEAR)

Abstract: The paper discusses some of the recent advances in kriging based worst-case design optimisation and proposes a new two-stage approach to solve practical problems. The efficiency of the infill points allocation is improved significantly by adding an extra layer of optimisation enhanced by a validation process.

Key words: worst-case optimisation, minimax problems, kriging, robust design

1. Introduction

It is often the case that sampling in modern engineering design may be constrained due to considerations of computational costs, time and thus available resources. In the context of electromagnetic design – where nowadays numerical simulation tools need to be used, such as the finite element method (FEM) – repetitive evaluation of the objective function may take hours or days of computation making the design process impractical. When extensive sampling is not available, surrogate based optimisation may significantly improve the design efficiency.

2. Worst-case problem specification

Robust optimisation is a relatively new term, its history can be dated back to 1989 when Taguchi first introduced the concept of design quality [1], and since then optimisation involving uncertainties has been increasingly drawing more attention. Due to the complexity of the optimisation problems in engineering design, the high level of non-linearity means these problems cannot be closely approximated by single linear or quadratic functions. Therefore, these problems are often solved by using direct search global optimisation algorithms. When evaluation of the underlying problem is expensive in terms of time or cost, surrogate modelling

techniques are often implemented as an approximation and optimisation is applied to the surrogate model instead of the original problem.

The output f of a black-box function, when the input variable x contains deterministic type of uncertainties, can be expressed by a simplified equation (ignoring possible other sources of uncertainties and assuming the uncertainty ε is independent of the input variable x) [2]

$$f = f(x + \varepsilon) \tag{1}$$

where $\varepsilon[-\varepsilon, \varepsilon]$, the distribution of uncertainty ε, is unknown, but the magnitude is bounded by a given range ε.

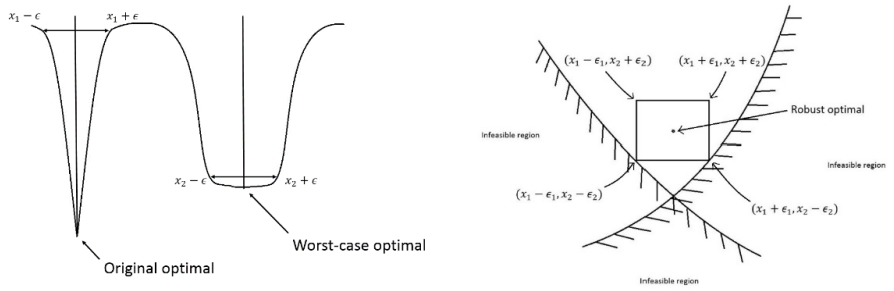


Fig. 1. Worst-case examples. 1 – 1D example, 2 – 2D example with constraints

Two simple examples are given in Fig. 1. The 1D illustration shows that, depending on the size of the uncertainty ε, the preferred worst-case robust optimum may differ from the original (theoretical) optimal point, as when the parameter varies within the specified limits (e.g. imposed by manufacturing tolerances) the performance worsens significantly, whereas the robust optimum ensures good performance throughout. The 2D example illustrates that the worst-case optimum needs to consider design constraints, as the optimum solution with uncertainties must not violate the infeasible region.

3. A brief review of existing approaches

For a deterministic type of uncertainties, the basic approach is to transform the robust optimisation problem into a standard optimisation problem by optimising the worst-case of the original objective function, where multiple objective function evaluations are needed at each design stage. The number of objective function calls may be significantly increased and many unimportant and possibly nearly duplicated design points will be allocated during this process and thus making the optimisation extremely inefficient. This large number of function calls will be of particular concern to designers, especially when the objective function is expensive to evaluate, which is often the case in electromechanical or electromagnetic design where the main tool for field modelling involves numerical computation (such as finite elements).

Recently, some more efficient kriging based approaches for solving worst-case optimisation problems have been proposed in literature. The authors of [3] use the mean and variants to assess the robustness, while their proposed strategy utilises the gradient information computed from the kriging model. In [4] the Expected Improvement (EI) infill sampling approach is combined with a relaxation procedure based on a kriging model. In [5] the EI infill sampling approach is applied to the worst-case response surface calculated based on the kriging model.

4. A two stage approach

In this paper, we propose a two-stage approach for solving computationally expensive worst-case optimisation problems. We focus on maximising the usage of available information while delaying the calculation of the worst-case value at sampling points to achieve a more efficient sampling scheme for the worst-case type of robust design optimisation.

The worst-case optimisation problem is often referred to as the minimax problem, with an extra 'layer' of optimisation, therefore the infill sampling criteria for global optimisation are often found inappropriate in the context of the worst-case optimisation problems. The worst-case value of the objective function at any given point does not depend on information given by that point alone (including the kriging prediction, mean squared error MSE, gradient etc.), as information from its neighbouring points also needs to be taken into account.

The algorithm consists of two stages: the first one is to update the kriging model by sequentially adding infill points at each iteration based on the worst-case expected improvement (WCEI) – this expected improvement measure is recalculated from standard EI, by taking the minimal EI value within the worst-case region of that design point (design site)

$$WCEI(x) = \max\left\{\min_{x+\varepsilon \in X}[EI(x+\varepsilon)],0\right\} \tag{2}$$

where X is a set of points located within the worst-case region of the unknown point x. A 1D example is illustrated in Fig. 2, where the boundary ε of the worst-case design is ± 0.3.

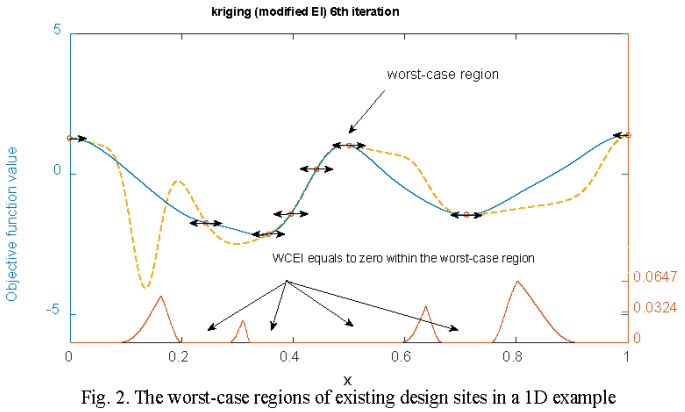


Fig. 2. The worst-case regions of existing design sites in a 1D example

The extra layer of the minimax problem is embedded within the WCEI; the new infill sampling point will be located where the minimal expected improvement around the target point is the largest. The WCEI is equal to zero at the locations within the worst-case region of existing design sites; consequently, these areas are banned as future infill locations at the model updating stage. During the process of model updating, the worst-case estimation of the objective function is computed simultaneously based on the kriging model constructed using the existing design sites at that iteration.

The second stage is triggered when the maximum WCEI within the design space becomes less than a predefined value or stage one has exceeded its allowance, if such limit has been imposed. An exploitation process takes place in stage two; the worst-case region around the worst-case optimum is exploited and validated using a modified EI approach, where instead of calculating the improvement, an expected 'deterioration' is estimated to give an indication where the maximal worsening is located within the worst-case region of the worst-case optimum

$$E[D(x)] = \begin{cases} (\hat{y}(x) - y_{wc})\Phi(u(x)) + \hat{s}\phi(u), & s > 0 \\ 0, & s = 0 \end{cases} \tag{3}$$

$$u = \frac{\hat{y}(x) - y_{\min}}{\hat{s}(x)}$$

This process is repeated until the value of the expected deterioration is zero or smaller than a predefined value; at this stage the location of the worst-case estimated optimum is added as the next infill point and the associated objective function is evaluated. When the range of the underlying objective function surface is large, both the location and value of the actual worst-case optimum can differ from the estimated one; therefore, the above validation process provides a more accurate prediction within the area of interest, and thus helps the program to locate the best worst-case optimum both efficiently and accurately.

5. Example

The worst-case optimisation routine following the two-stage approach is first illustrated using a one-dimensional test example. Figure 3 shows the original test function and its associated worst-case distribution, where the 'boundary' ε of the worst-case design was assumed to be ±0.3. It can be observed that both the landscape and in particular the position of the optimum differ noticeably between the original function and the worst-case version.

In Figures 4 (a) to (f) the yellow dotted line depicts the objective function (the worst-case version in (b) and (f), the original shape elsewhere), while the blue bold line shows the kriging prediction. Figure 4(a) shows the kriging model after the 11th iteration. As the maximum WCEI within the design space is less than the predefined value of 10^{-3}, the program enters stage two and the region around the worst-case estimated optimum at $x = 0.12$ (see Fig. 4(b)) is exploited. The infill criterion in Fig. 4(c) illustrates the value of expected deterioration within this region. The estimated worst-case optimum is updated during the validation process and the exploitation area is then moved to a region around the new estimated worst-case optimum at $x = 0.353$. In Fig. 4(d) the value of expected deterioration equals to zero at the 15th iteration; the final worst-

case optimum is then located at $x = 0.303$ in Fig. 4(e), while Fig. 4(f) shows the final shape of the estimated worst-case objective function, as well as the actual worst-case objective function.

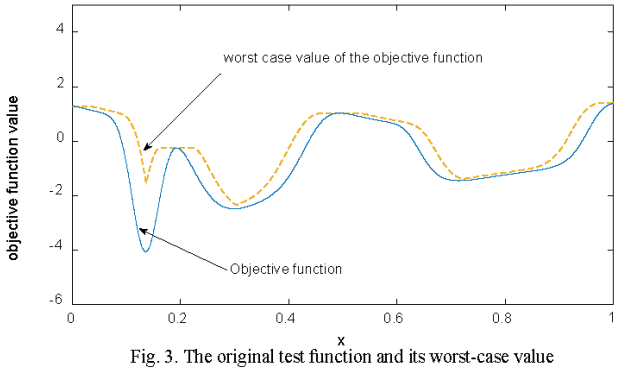


Fig. 3. The original test function and its worst-case value

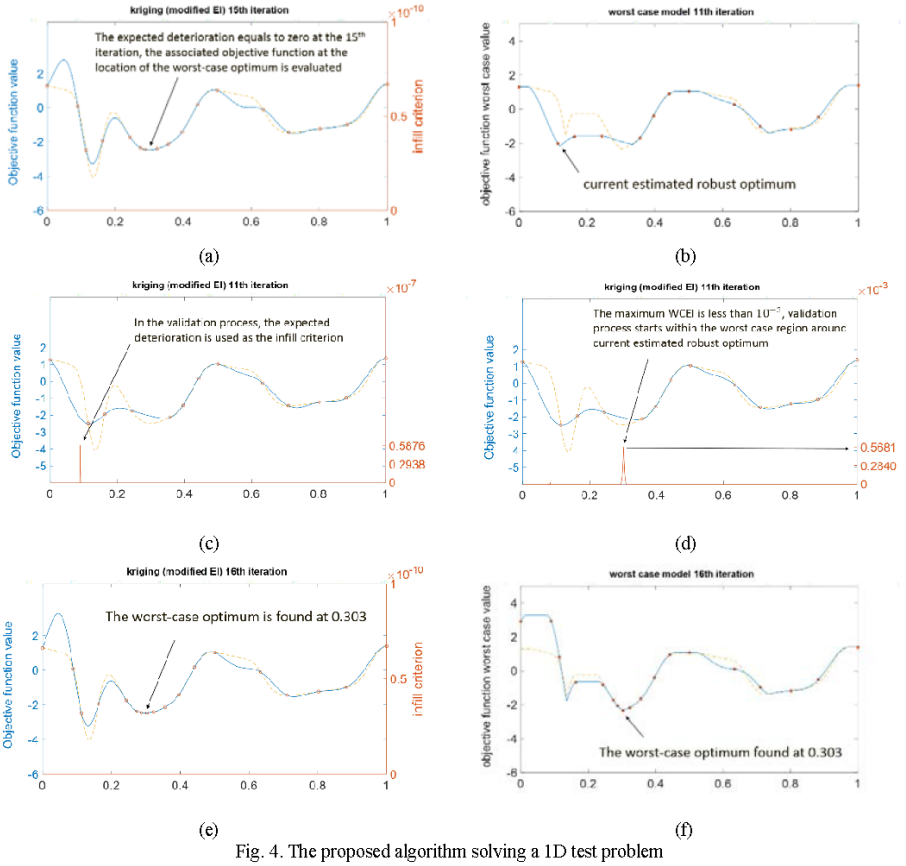


Fig. 4. The proposed algorithm solving a 1D test problem

6. Gradient decent with multi-start algorithm for infill points selection

In the context of global optimisation based on surrogate modelling, the location of the next sampling point is based on the infill point sampling scheme which contains a set of rules or formulae. Taking standard Expected Improvement (EI) approach as an example, it takes the predicted function value, estimated error and the current optimal value as the input variables, and yields the expected improvement for that particular point; the point with the largest expected improvement will be the next infill point. In order to locate reliably the point with the maximum EI, all points within the design space need to be considered, necessitating the calculation of the predicted objective function and the corresponding MSEs at all points. Using an exhaustive method to locate the point with the maximum EI is relatively straightforward in a 1D scenario, but may not be acceptable (or practical) in multidimensional cases, where the number of EI calculations increases exponentially if the same sampling interval of EI is maintained.

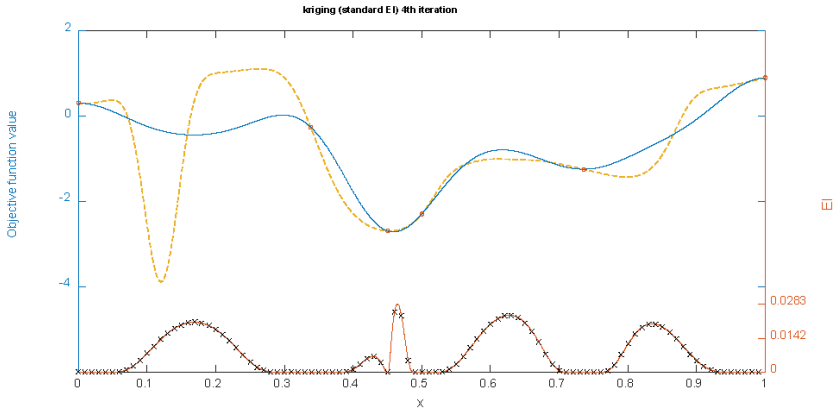


Fig. 5. EI at predefined sampling locations for a 1D test function

Figure 5 illustrates a 1D example problem, showing the original objective function (dotted yellow line) and the kriging model (bold blue line), while the bold cinnabar line at the bottom (with crosses on top) is the true EI curve. 101 EI sampling locations have been evenly distributed within the design region with a fixed interval of 0.01, with the small black crosses on top of the EI curve marking the sampled EI at those predefined locations. As can be seen from the figure, the sampled EI is a reasonable approximation of the true EI curve, but not very accurate at the 'critical' location around the maximum EI.

The EI curve in Fig. 5 exhibits certain useful characteristics: it is differentiable and the local maximum between two existing design points is often close to their mid-point. We could take advantage of these features in order to significantly speed up the infill sampling process. Instead of using an exhaustive search of all predefined locations, a gradient decent approach with a starting point in the middle between two design points has been incorporated into the algorithm and is illustrated by an example in Fig. 6.

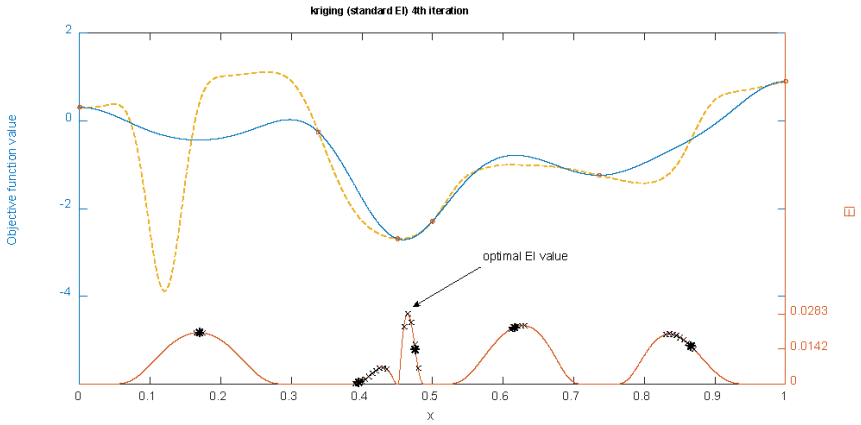


Fig. 6. A gradient decent approach for the EI search

In the example given in Fig. 6, the search for the maximum EI is not restricted by the predefined step size, the optimal EI is found with reasonable accuracy while reducing the number of EI calculations by half.

For a multidimensional design space, where the local maximum EI is located in between multiple existing design sites, a multi-start strategy can be applied to replace a single starting point, as in the 1D case, by simply generating starting points in the middle of each pair of two existing design sites. It is a combination problem and the number of starting points p for n known points is $p = nC2$. Hence for 100 existing design points, 4950 starting points for the gradient decent calculations are generated.

7. Solving practical problems

TEAM workshop problems [6] consist of a set of practical electromagnetic optimisation design problems for benchmarking the performance of algorithms. Here we test the proposed approach on two practical benchmark problems. For both problems, the uncertainty boundary for each design parameter (upper and lower limit) is defined as 1% of their given design range.

The superconducting magnetic energy storage device in TEAM problem 22 [7] contains two superconducting coils; the design objective is to achieve a minimal stray field while the stored energy should be equal to 180 MJ. The configuration of the inner coil is given in the 3 Parameter ('discrete') case, and therefore there are three parameters to be optimised, namely the radius R_2, height h_2 and thickness d_2 of the outer coil, as indicated in Fig. 7.

The objective function is given as

$$OF = \frac{B_{stray}^2}{B_{norm}^2} + \frac{\left|E - E_{ref}\right|}{E_{ref}} \tag{4}$$

where E_{ref} = 180 MJ, B_{norm} = 3 μT and B_{stray}^2 is defined as

$$B_{stray}^2 = \frac{\sum_{i=1}^{n=22} \left| B_{stray,i} \right|}{22} \tag{5}$$

subject to the following inequality constraint, known as the quench condition

$$|J| + 6.4|B| - 54.0 \le 0 \tag{6}$$

where J (in A/mm^2) is the current density and B the maximum magnetic flux density (Fig. 8).

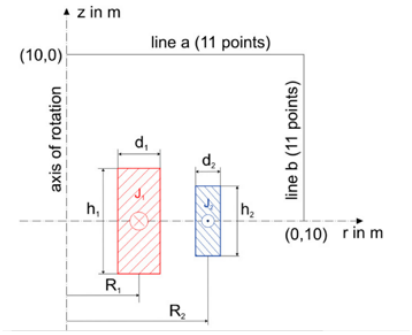


Fig. 7. The superconducting magnetic energy storage device [6] [7]

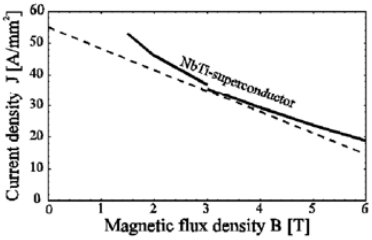


Fig. 8. Critical curve of an industrial superconductor [6] [7]

Table 1. Comparison of performance of various optimisation methods (TEAM 22 problem)

Algorithm	Optimal value	R_2	d_2	$h_2/2$	No of function calls
GA	0.134	3.040	0.386	0.240	2400
SA	0.098	3.078	0.390	0.237	5025
HuTS	0.089	3.080	0.380	0.246	3821
NTS	0.089	3.080	0.370	0.254	1800
PBIL	0.101	3.110	0.421	0.241	3278
Kriging EI	0.0875	3.090	0.394	0.236	211
Kriging AWEI	0.0875	3.090	0.400	0.232	323
Kriging WCEI (worst case)	0.1459	3.021	0.391	0.250	277

Genetic algorithm GA [9]; Stimulated Annealing SA [10]; Tabu Search HuTS [11]; Universal Tabu search [12]; New Tabu Search NTS [13]; Kriging EI [14]; Kriging AWEI [14]

Table 1 summarises the findings, citing the results from other publications and adding the robust 'worst-case' design. The kriging assisted optimisation has performed consistently well by achieving a marginally better solution with much reduced effort (the number of necessary function calls reduced by almost an order of magnitude). The robust optimum is – perhaps not surprisingly – a little different, slightly worse in terms of the value of the achieved objective function but assuring the robustness within the specified 1% uncertainty of parameters, while preserving the efficiency of computation (small number of function calls).

The device in Problem 25 [6, 8] is a die press with an electromagnet for the orientation of magnetic powder to produce an anisotropic permanent magnet. The objective is to optimise the shape of the model (Fig. 9). The shape g-h of the inner die mold is assumed to be a circle, the inside shape i-j-m of the outer die mold is elliptical and the line j-k is parallel to the *x*-axis; the ampere-turns of the coil are chosen as 4253A.

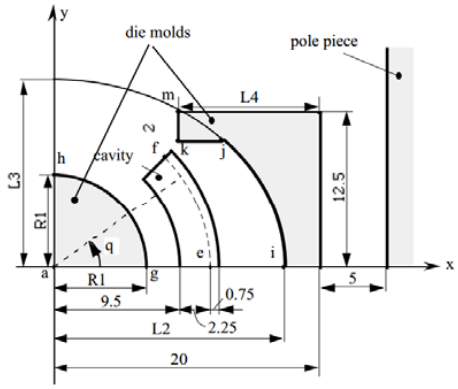


Fig. 9. A model of the die press with an electromagnet [6, 13]

In Fig. 9, R1, L2, L3 and L4 are the design parameters to be optimised so that the objective function W is minimized

$$W = \sum_{i=1}^{n} \left\{ \left(B_{xip} - B_{xio}\right)^2 + \left(B_{yip} - B_{yio}\right)^2 \right\} \tag{7}$$

where B_x and B_y are the x and y components of magnetic flux density at points along the curve e-f, while subscripts p and o denote the calculated and desired values, respectively. The constraints are listed in Table 2 and results in Table 3.

TABLE 2. Constraints of the parameters

Variable	Lower-boundary (mm)	Upper-boundary (mm)
R1	5	9.4
L2	12.6	18
L3	14	45
L4	4	19

TABLE 3. Comparison of performance of various optimisation methods (TEAM 25 problem)

Algorithm	Optimal value (10^{-4})	R1 (mm)	L2 (mm)	L3 (mm)	L4 (mm)	No of function calls
GA	2.6861	7.2996	14.174	14.001	14.326	3421
SA	1.6223	7.2252	14.322	14.110	14.306	2145
HuTS	0.5009	7.3780	14.613	14.371	14.204	1580
UTS	1.0501	7.5487	14.908	14.506	14.416	931
NTS	0.6482	7.4337	14.732	14.428	14.237	575
Kriging EI	0.4527	7.2	14.1	14	14.5	265
Kriging AWEI	0.4125	7.2	14	14	14.5	214
Dual kriging	0.3231	7.1	13.9	14.014	14.273	234
Kriging WCEI (worst case)	5.4442	7.104	13.891	14.035	14.270	453

Genetic algorithm GA [9]; Stimulated Annealing SA [10]; Tabu Search HuTS [11]; Universal Tabu search [12]; New Tabu Search NTS [13]; Kriging EI [14]; Kriging AWEI [14]; Dual kriging[15]

The main observations resulting from the TEAM 25 study are broadly in line with what was demonstrated before, although on this occasion the robust optimum is clearly somewhat 'worse' (in absolute terms, but of course it is more robust) and the algorithm is computationally a little less efficient, although still comfortably outperforms other non-kriging methods. It is also interesting to note that for both TEAM problems the originally published results (when the problems were first suggested) appear to be reasonably robust, more so than the subsequently offered solutions. The most important conclusion, however, resulting from this study is that the kriging assisted optimisation is very reliable and offers superbly efficient computation, both for the 'traditional' (global) optimisation and the robust formulation. Finally, the worst case (minimax) approach appears to be a very helpful methodology for robust optimisation.

8. Conclusions

A two-stage approach to worst-case optimisation problems has been proposed and details of the algorithm discussed. The suggested method does not compute the worst-case value, or the corresponding robustness measure, for any design site during the model updating stage, in order to avoid the objective function evaluation at a location that would contribute less to the overall model landscape, which would have taken place if the worst-case value had been evaluated for the newly added infill point. Instead, the explicit search for the robust optimum takes place in the second stage after the model updating process has completed, with a validation process added to exploit the region around the estimated worst-case optimum. A more efficient infill criterion selection algorithm has been introduced. The proposed optimisation method has been validated using simple test functions and two multi-dimensional practical electromagnetic design problems TEAM 22 and TEAM 25. The test results indicate that, with the aid of kriging surrogate modelling techniques, the proposed methodology significantly reduces the number of FEM function calls compared to other methods and thus is computationally very efficient for both global and robust optimisation.

References

[1] G. Taguchi, *Introduction to Quality Engineering*, American Supplier Institute (1989).

[2] Hans-Georg Beyer and Bernhard Sendhoff, *Robust optimisation - A comprehensive survey*, Comput. Methods Appl. Mech. Engrg. 196, pp. 3190-3218, 2007.

[3] Kwon-Hee Lee and Dong-Heon Kang, *A Robust Optimisation Using the Statistics Based on Kriging Metamodel*, Journal of Mechanical Science and Technology *(KSME Int. J.),* Vol. 20, No. 8, pp. 1169-1182 (2006).

[4] Julien Marzat, Eric Walter, Helene Piet-Lahanier, *Worst-case global optimisation of black-box functions through Kriging and relaxation*, Journal of Global Optimisation, Springer Verlag, 55 (4), pp. 707-727 (2013).

[5] Samee ur Rehman, Matthijs Langelaar, Fred van Keulen, *Efficient Kriging-based robust optimisation of unconstrained problems,* Journal of Computational Science, Volume 5, Issue 6, pp. 872–881 (2014).

[6] http://www.compumag.org/jsite/team.html

[7] P.G. Alotto, U. Baumgartner, F. Freschi, M. Jaindl, A. Kostinger, Ch. Magele, W. Renhart, M. Repetto, *SMES Optimisation Benchmark: TEAM Workshop Problem 22*.

[8] N. Takahashi, K. Ebihara, K. Yoshida, T. Nakata, K. Ohashi, and K. Miyata, *Investigation of simulated annealing method and its application to optimal design of die mold for orientation of magnetic powder*, IEEE Trans. Magn., vol. 32, no. 3, pp. 1210–1213 (1996).

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Cambridge, MA, USA: MIT Press (1975).

[10] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, *Optimisation by simulated annealing*, Science, vol. 220, no. 4598, pp. 671–680 (1983).

[11] N. Hu, *Tabu search method with random moves for globally optimal design*, Int. J. Num. Methods Eng., vol. 35, no. 5, pp. 1055–1070 (1992).

[12] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, New York, USA, Springer-Verlag (2003).

[13] O. Hajji, S. Brisset, and P. Brochet, *A new Tabu search method for optimisation with continuous parameters*, IEEE Trans. Magn., vol. 40, no. 2, pp. 1184–1187, Mar (2004).

[14] S. Xiao, M. Rotaru, and J. K. Sykulski, *Six Sigma Quality Approach to Robust Optimisation*, IEEE Trans. Magn., vol. 51, no. 3 (2015).

[15] Y. Li, S. Xiao, M. Rotaru, and J. K. Sykulski, *A Dual Kriging Approach with Improved Points Selection Algorithm for Memory Efficient Surrogate Optimisation in Electromagnetics*, IEEE Transactions on Magnetics, 52, (3), 1-4 (2015).

A kriging based optimization approach for large datasets exploiting points aggregation techniques

Yinjiang Li, Song Xiao, Mihai Rotaru, and Jan K. Sykulski, *Fellow, IEEE*

Electronics and Computer Science, University of Southampton, Southampton, UK, jks@soton.ac.uk

***Abstract*—A kriging based optimization approach is proposed for problems with large datasets and high dimensionality. Memory usage is maintained via model centering aided by minimizing the impact of information loss on accuracy of new point prediction using points aggregation techniques. The 8-parameter TEAM problem 22 is revisited in the context of computational efficiency and accuracy.**

***Index Terms*—kriging, surrogate optimization, clustering, large datasets.**

I. Introduction

Surrogate modelling techniques are helpful tools in design optimization, especially when the underlying problem is computationally expensive. This situation frequently arises in the design of electromagnetic devices where time consuming finite element simulations are often necessary to ensure accurate performance prediction, or physical modelling is called upon. Kriging based methodologies have been shown to be particularly useful and offer good accuracy of the estimation while reducing the number of required objective function calls. Unfortunately the complexity of the algorithm increases as solving the kriging model involves the inversion of a correlation matrix, which results in O(n^3) computation cost and O(n^2) storage cost. Consequently, the otherwise efficient application of kriging is often limited to smaller scale design problems. Much effort has been devoted to address this bottle-neck of kriging methods when applied to large datasets, some examples include zooming-in modelling [1], moving-window kriging [2], covariance tapering [3] and fixed rank kriging [4].

In this paper points aggregation is proposed. The method involves locating the most interesting search area for the next infill point, then aggregating points outside this center area, and finally building a kriging model for infill point search within the identified center area.

II. Center Positioning

The objective is to locate a center around which a prescribed number of 'interior' points will not be aggregated and where the next infill point will be located inside this region. For an arbitrary location inside the design space a corresponding $Criterion_c$ can be introduced based on the following equations, then the location with the maximum $Criterion_c$ will be defined as the model center

$$Criterion_c(x) = \begin{cases} C_1 + C_2, & random(0,1) < v \\ C_3, & otherwise \end{cases} \quad (1)$$

$$C_1 = {R_{C_1}}^{-1} \times \sum_{i=1}^{k} \|\boldsymbol{x}_i - \boldsymbol{c}\|^{\frac{1}{2}}, \quad R_{C_1} = k \times \max\left\{\|\boldsymbol{x}_i - \boldsymbol{c}\|^{\frac{1}{2}}\right\} \quad (2)$$

$$C_2 = {R_{C_2}}^{-1} \times \sqrt{\frac{\sum_{i=1}^{k}(y_i - \mu)^2}{k}}, \quad R_{C_2} = \frac{range(\mathbf{y})}{2} \quad (3)$$

$$C_3 = {R_{C_3}}^{-1} \times \left(max(\boldsymbol{Y}) - \frac{\sum_{i=1}^{k} -w_i y_i}{\sum_{i=1}^{k} w_i}\right), \quad (4)$$

$$R_{C_3} = range(\boldsymbol{Y}), \text{ and } w = e^{-v^{-5}(x_i - \boldsymbol{c})} \quad (5)$$

where $\boldsymbol{c}$ denotes the center, $\boldsymbol{x}_i$ is the location of its i^{th} closest point, k defines the number of closest neighborhood points around $\boldsymbol{c}$, y_i denotes the objective function values of the i^{th} closest neighborhood point, w_i is the weight term which has an inverse relationship with the distance from point i to the center $\boldsymbol{c}$, μ is the mean of $\boldsymbol{y}$, and v is the calculated probability.

C_1 in (1) is the sum of square roots of Euclidean distances between the hypothetical center $\boldsymbol{c}$ and k nearest points around it. The value of C_1 is a measure of a sample rate within the region; it determines how close a hypothetical center $\boldsymbol{c}$ is located in relation to its nearest k points, while the square root de-emphasizes the importance of remote points. C_2 is the weighted standard deviation of the objective function values of all the neighborhood points. Finally, C_3 is the weighted mean of the objective function values of all the neighborhood points. Each point is weighted by an exponential function, whose gradient is controlled by the parameter v. The smaller values of v apply less weight on remote points.

The C_1 and C_2 terms will encourage exploration of the under-sampled and rough areas, respectively, while C_3 focuses on exploitation of the current optimum region.

The probability of exploration and exploitation is controlled by a parameter v, whose value is related to the root-mean-square deviation (RMSD) of the kriging model. Instead of a deterministic mixture of exploration and exploitation terms, a stochastic approach has been applied to search the objective at different stages, while eliminating the risk of the deterministic criterion function being trapped in a local optimum.

Manuscript received November ?, 2016; revised ?? ??, 201? and ?? ?, 201?; accepted ?? ?, 201?. Date of publication ?? ?, 201?; date of current version ?? ?, 201?. Corresponding author: J. K. Sykulski (e-mail: jks@soton.ac.uk).
Digital Object Identifier (inserted by IEEE).

The predictor deviation d in iteration $iter$ is defined as

$$d_{iter} = f(\boldsymbol{x}_{iter}) - Pred_{iter-1}(\boldsymbol{x}_{iter}) \quad (4)$$

where $\boldsymbol{x}_{iter}$ is the location of the infill point in the $iter^{th}$ iteration, $f(\boldsymbol{x})$ is the evaluated objective function at location $\boldsymbol{x}$ and $Pred_{iter-1}(\boldsymbol{x})$ is the predicted objective function value at location $\boldsymbol{x}$ in iteration $iter-1$.

The deviation d_{iter} is calculated and recorded whenever a new infill point is defined. Finally, the historical root-mean-square deviation (RMSD) is

$$RMSD = \sqrt{\frac{\sum_{iter=1}^{m} {d_{iter}}^2}{m}} \quad (5)$$

where m is the most recent iteration.

To obtain a generalized weight term, an exponentially weighted RMSD is applied in this case in order to put more weight on recent results; the aim is to emphasize the recent prediction error to reflect on the optimization progress. The exponentially weighted RMSD is calculated using the formula

$$RMSD_{weighted} = \sqrt{\frac{\sum_{iter=1}^{m}(1-\alpha)^{m-iter} \times {d_{iter}}^2}{\sum_{iter=1}^{m}(1-\alpha)^{m-iter}}} \quad (6)$$

where α is the decay parameter and $0 < \alpha < 1$. A larger α will put less weight on past prediction errors and vice versa. When $\alpha = 0$, $RMSD_{weighted}$ is identical to $RMSD$.

A generalized weight term that represents the current optimization progress in terms of model prediction deviations may be defined by taking a ratio of the exponentially weighted RMSD and regular RMSD of historical prediction errors

$$v = weighted\_RMSD/RMSD \quad (7)$$

The parameter v can be regarded as a measure of model quality at any stage, v often ranges between α and $1+\alpha$, and α controls the gradient of the exponential weight function; as model deviation decreases v will gradually move towards zero.

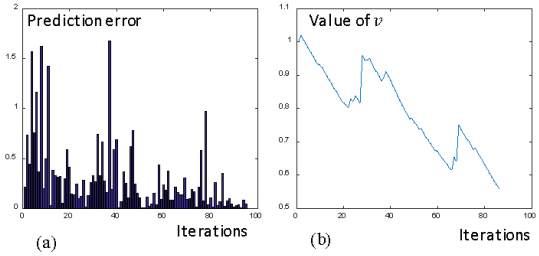


Fig. 1. (a) The history of the prediction error, (b) the history of the value of v, as iterations progress.

Regardless of which criterion function is used, this optimization stage involves finding the number of closest neighborhood points k around $\boldsymbol{c}$ and calculating the parameter v, which requires extra computation resources, therefore finding the hypothetical center $\boldsymbol{c}$ using exhaustive search is not practical. It may be seen as a global optimization problem with the input $\boldsymbol{c}$, the output $Criterion_c$, and the objective function $Criterion_c = (C_1 + C_2)$ or $Criterion_c = C_3$. Because the center only defines an area for the new infill point, an approximate solution will suffice at this step; thus a stochastic sequential global optimization method of simulated annealing has been utilized. It may be argued that as the precision of estimating the intermediate optimum is not that important, a sequential method will have the advantage of fewer function calls and more flexibility over population base methods.

The response surface of a 2D function is plotted in Fig. 2, with the red crosses at the bottom marking the location of existing design points. Figs. 3 (a) and (b) illustrate the criterion function for exploration and exploitation terms at the 90th iteration, respectively. As can be seen from the figures, the C_1+C_2 term encourages search in less sampled and non-smooth areas around x=[0.92,0.79], while the C_3 function suggests exploration of the area around the minimum at x=[0.20 0.27].

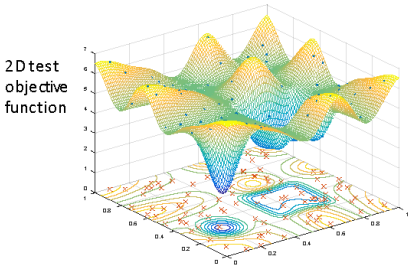


Fig. 2. A 2D test function and existing design points (with normalized axes).

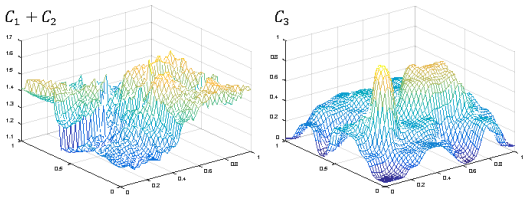


Fig. 3. (a) Exploration functions $C_1 + C_2$, (b) exploitation function C_3.

III. Outside Points Aggregation

The objective of this step is to aggregate design points further from the model center into a smaller number of nodes (the 'knots'), so that the total number of nodes and points in the model can be fitted into the memory. The problem of outside points aggregation involves hierarchical cluster analysis (HCA) [5] and a single variable optimization design. The objective of HCA is to group outsider points into a set of clusters so that the number of clusters is equal to the number of nodes. The value of the node is equal to the weighted mean of aggregated points.

There is rich literature related to cluster analysis, in particular in the field of data science, and many algorithms have been published. The points aggregation can be treated as a k-mean clustering problem where there are significantly more clusters to be identified compared to conventional clustering problems.

In this paper we developed a sequential algorithm for weighted points clustering, the pseudo code of which is given below

```
for m = 1: sample size
    for n = 1: number of clusters
        · calculate new cluster centroid
        · calculate weighted Euclidean distance
          to centroid
    end
    · find the cluster(x) with minimum weighted
      Euclidean distance e
    if e < weighted dissimilarity
        · add point(m) to cluster(x)
    else
        · create a new cluster
    end
end
```

The cluster's centroid o of a set of m points x is given by

$$o(x) = \frac{1}{p} \times \sum_{i=1}^{p} x_i \tag{8}$$

where p is the number of points to be considered.

The Euclidean distance is weighted by the distance between the cluster's centroid o and the model's center c, based on a correlation function, while the dissimilarity is weighted too, as a larger distance results in lower correlation and therefore information loss due to points aggregation will have a smaller impact on the prediction result within the center area. The design space is normalized and each cluster's centroid is weighted by the Gaussian function.

The Gaussian correlation function in the kriging model is used to calculate the weight w; the original function is given by

$$f(x_i, x_j) = e^{-\theta d_{ij}^2} \tag{9}$$

Because the hyperparameter θ needs to be tuned during the model construction process, and is unknown at the stage of outside points aggregation, we specify $\theta = 2$ as this provides a smoother decay in correlation and gives generally good results when the underlying problem is unknown.

The optimization problem is defined as $OF(d) = (n - q)^2$, where d is the input variable dissimilarity, n is the number of nodes/clusters generated during the clustering process, q is the number of outside nodes that can be fitted into the memory. The pseudo code provided above shows a basic workflow of the clustering process; to speed up the process, clusters with the minimum value of $n - c$ are kept in memory and a new clustering iteration starts with these existing clusters. The clustering process is terminated when the sum of the number of existing clusters and the number of unclassified points is less than the number of nodes calculated previously.

The following example illustrates outside points aggregation applied to a 2D problem. Fig. 4(a) shows the clustering without Gaussian weights, while Fig. 4(b) illustrates the clustering with Gaussian weight terms applied. The problem consists of 500 observations, assuming that the memory can build a kriging model up to 250 design points. We specify that 40% of the memory will be used to store the interior points within the model's central area, while the remaining 60% of memory is used to store nodes related to outside points. The 400 points outside the center area are aggregated into 150 nodes.

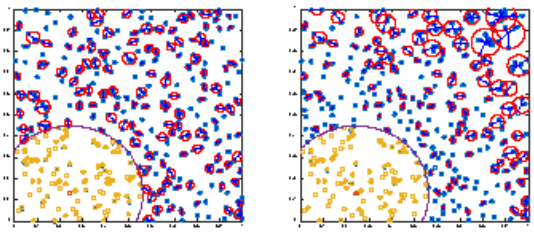


Fig. 4. (a) Clustering without Gaussian weight functions, (b) Clustering with Gaussian weight functions (with normalized axes).

IV. A 2D EXAMPLE

The point aggregation technique is illustrated in Figs. 5 and 6 by a 2D example. The kriging model in Fig. 5 is built based on 100 design points, while the model in Fig. 6 contains 60 nodes, including 20 inside points within the center area and 40 nodes outside the center area, where 62 original points have been involved in the aggregation. Memory allocation for the correlation matrix is 36% of the original size, thus considerable savings in memory requirements have been accomplished.

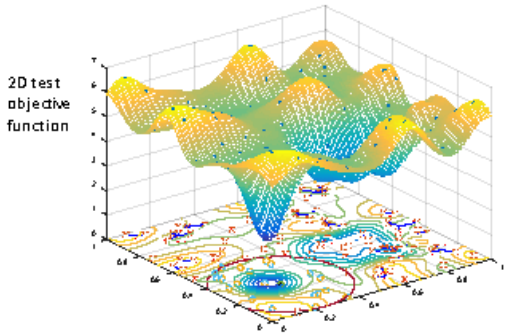


Fig. 5. Kriging estimate of a 2D test function (100 design points).

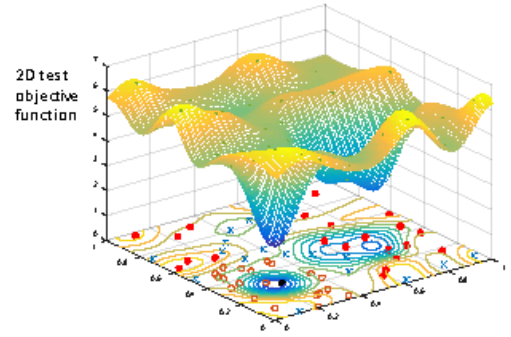


Fig. 6. Kriging estimate after point aggregation (60 nodes).

TABLE I
PERFORMANCE COMPARISON BETWEEN DIFFERENT ALGORITHMS

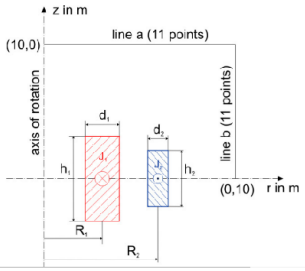
Algorithm	R_1 (m)	R_2 (m)	h_1 (m)	h_2 (m)	d_1 (m)	d_2 (m)	J_1 (A/mm^2)	J_2 (A/mm^2)	Objective function	Constraints penalty [2]	No. of FEM calls
PSO	1	2.2647	1.1076	1.7766	0.5225	0.3442	28.1779	–5.4921	1.5673	85.0413	~6000
Q-PSO	2.2947	2.6126	0.39	2.2704	0.3967	0.204	30	–21.293	2.4016	13.3456	~6000
E-QPSO	1	1.8	0.38	3.6	0.5155	0.2851	19.9975	–6.3571	0.3464	0.3685	~6000
GSA	1.939	2.823	0.37	1.101	0.399	0.195	22.5	–22.5	1.5547	0.195	17150
ES	1.99	2.931	0.421	0.94	0.29	0.188	26.6	–26.6	0.4103	1.69235	4200
SAA	1.694	2.907	0.394	0.882	0.323	0.207	20.9	–20.9	1.0087	1.09395	14000
Kriging standard	1	1.8	1.56	1.39	0.4	0.15	30	–30	1.4065	32.9056	449
Kriging proposed [1]	3.272	3.573	1.819	1.106	0.195	0.154	26.932	–23.259	0.0383	0.0159	500 [3]
	1.103	2.318	3.193	0.288	0.259	0.734	22.5	–22.5	0.0014	0.0003	829 [4]
Original answer	1.296	1.8	2.178	3.026	0.583	0.195	16.955	– 18.91	0.0033 [5]	0	–

PSO: particle swarm optimization [6], Q-PSO: quantum-behaved particle swarm optimization [6-8], E-QPSO: QPSO with exponential probability distribution [9], GSA: global search algorithm [10], ES: evolution strategy [10], SAA: simulated annealing algorithm [10]. Results for PSO, Q-PSO, E-PSO, GSA, ES and SAA taken from [6] to [10]. The comparison is for the 8 parameter continuous case [11]. Notes:

1) The new kriging algorithm offers significant savings in memory related to the correlation matrices; this has been achieved by aggregating the outside points.
2) Solutions from some previously published methods have violated the quench condition; the degree by which this constraint has not been met is given by the 'penalty' (high values indicate severe violation). In some cases the geometrical or current density constraints have not been met either.
3) For a fairer comparison of memory usage between standard kriging and the proposed kriging method, the maximum number of iterations was set to 500, while maintaining a maximum of 375 nodes; a memory saving on correlation function of ~50% was achieved and – as a bonus – a better optimum was found.
4) The proposed enhanced kriging method may be allowed to continue the search with the number of nodes maintained at 500; improved results have been achieved (better value of objective function and lower constraint violation) after more iterations, at the modest expense of more FEM calls.
5) The value of the objective function in the original specification was a little different; it was recalculated here using a consistent FEM model for comparison.

V. PRACTICAL EXAMPLE TEAM 22

The superconducting magnetic energy storage device in TEAM problem 22 consists of two superconducting coils. The design objective is to minimize the stray magnetic field while maintaining the stored energy at 180 MJ (see Fig. 7), subject to specified quench conditions and geometrical constraints [11].



$$OF = \frac{B^2_{stray}}{B^2_{norm}} + \frac{|E - E_{ref}|}{E_{ref}}$$

$$E_{ref} = 180\ MJ,$$
$$B_{norm} = 200\ \mu T$$

$$B^2_{stray} = \frac{\sum_{i=1}^{n=22} |B_{stray,i}|}{22}$$

Fig. 7. The superconducting magnetic energy storage device (TEAM 22) [11].

The results are summarized in Table I. As on previous occasions kriging has shown its superiority by dramatically reducing the number of necessary function calls and thus avoiding excessive use of the computationally expensive finite element software. Moreover, the addition of the points aggregation has offered an additional flexibility of limiting the number of active points in the design space; this has the benefit of reducing the memory requirements for the solution without sacrificing the accuracy. Finally, the iterations may be allowed to continue to achieve a better design with a modest increase of computational effort due to the need for more FEM calls.

VI. CONCLUSIONS

A kriging based optimization approach for large datasets has been proposed and its efficiency demonstrated using the TEAM 22 problem. The model center positioning algorithm balances exploration and exploitation assisted by the use of a stochastic approach, which eliminates the risk of a deterministic criterion function being trapped in a local optimum. It has been found that the size of the correlation matrices can be greatly reduced by applying points aggregation techniques. It is shown that the proposed approach can fit a large set of data into a limited size of memory and whereas some loss of information about remote points may be experienced this is alleviated by the use of points aggregation incorporating a new weighted clustering algorithm.

REFERENCES

[1] Xiao, Song, Rotaru, M. and Sykulski, J.K., "Adaptive Weighted Expected Improvement With Rewards Approach in Kriging Assisted Electromagnetic Design," *IEEE Trans. Magn.*, 49, (5), Spring Issue, 2057-2060, 2013.
[2] Timothy C. Haas, "Lognormal and Moving Window Methods of Estimating Acid Deposition," *Journal of the American Statistical Association*, vol. 85, no. 412, pp. 950-963, 1990.
[3] Reinhard Furrer, Marc G. Genton and Douglas Nychka, "Covariance Tapering for Interpolation of Large Spatial Datasets," *Journal of Computational and Graphical Statistics*, 15:3, pp. 502-523, 2006.
[4] Noel Cressie and Gardar Johannesson, "Fixed rank kriging for very large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70: 209–226, 2008.
[5] Murtagh F., "A Survey of recent advances in hierarchical-clustering algorithms," *The Computer Journal*, 26:354-359, 1983.
[6] Coelho, L.S., Alotto, P., "Global optimisation of electromagnetic devices using an exponential quantum-behaved particle swarm optimizer," *IEEE Trans. Magn.*, 44, (6), pp. 1074–1077, 2008
[7] Hogg, T., Portnov, D.S., "Quantum optimization," *Inform. Sci.*, 128, (3–4), pp. 181–197, 2000.
[8] Sun, J., Feng, B., Xu, W., "Particle swarm optimisation with particles having quantum behavior," Proc. *Congress Evolution of Computation*, Portland, OR, pp. 325–331, 2004.
[9] Krohling, R.A., Coelho, L.S., "PSO-E: Particle swarm with exponential distribution," Proc. *IEEE Congresss Evolution Computation*, Vancouver, BC, Canada, pp. 5577–5582, 2006.
[10] Kuntsevitch, A.V., Magele, C., Molinari, G., "Multiobjective optimization in magnetostatics: a proposal for benchmark problems," *IEEE Trans. Magn.*, 32, (3), pp. 1238–1241, 2002.
[11] P.G. Alotto, U. Baumgartner, F. Freschi, M. Jaindl, A. K¨ostinger, Ch. Magele, W. Renhart, M. Repetto, "SMES Optimisation Benchmark: TEAM Workshop Problem 22," 1996.

Localized Probability of Improvement for Kriging based Multi-Objective Optimization

Yinjiang Li, Song Xiao, Mihai Rotaru, and Jan K. Sykulski, *Fellow, IEEE*
Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
yl10e09@ecs.soton.ac.uk, jks@soton.ac.uk

***Abstract*—The paper introduces a new approach to kriging based multi-objective optimization by utilizing a local probability of improvement as the infill sampling criterion and the nearest neighbor check to ensure diversification and uniform distribution of pareto fronts. The proposed method is computationally fast and linearly scalable to higher dimensions.**

Keywords—kriging; multi-objective optimization; pareto front; surrogate-based optimization.

I. Introduction

Research on multiple objective optimization (MO) has been attracting significant attention of the engineering community since 1980s; with the aid of fast computers solutions to many complex optimization problems have been made possible. The Vector Evaluated Genetic Algorithm (VEGA) [1] is one of the earliest examples of Multi-Objective Evolutionary Algorithms (MOEAs). The more recent developments include NSGA-II [2] and its modified versions as well as Particle Swarm based methods [3]. For a comprehensive review of problem definitions and non-EA based solution methods, readers are referred to [4].

There is an increasing number of indicator-based MOEAs that have been proposed in recent years; the indicator is used as a fitness measure for a set of pareto points, and – by optimizing the indicator function – the MO problem essentially becomes a single objective optimization problem as the solver only needs to locate the optimal value of the indicator value and update the generation based on it. One of the best-known indicators is the hypervolume [5]; it has been successfully applied to both EAs and surrogate-based algorithms. Despite its unique feature of being strictly monotonic to pareto improvement [6], its high computational cost for higher dimensions is also widely known.

The general opinion favors EAs as advantageous in solving MO problems by often being population based, thus multiple solutions can be obtained in a single run. However, solutions to practical problems are usually expensive in terms of computational time and effort. In the context of electromagnetic devices the finite element method is a common design tool; it often takes hours or even days to obtain a single solution, therefore surrogate model based algorithms are often preferred.

In this study we propose a novel indicator focused Localized Probability of Improvement (LPoI) approach for MO problems; its implementation requires the predicted mean and mean square errors to be available, hence it is not applicable to other EAs, but for Gaussian based surrogate models (including those relying on kriging) it has the advantage of being linearly scalable to problems with higher number of objectives.

II. Localized Probability of Improvement

Compared to other surrogate modeling methods, kriging has the advantage of providing both the predicted mean and the associated mean square error (MSE) at an unknown location. The probability of improvement PoI at any location is given by

$$PoI(\boldsymbol{x}) = \Phi\left(\frac{y_t(\boldsymbol{x}) - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) \tag{1}$$

where y_t is the target of improvement, $\hat{y}$ is the kriging predicted mean at location $\boldsymbol{x}$, $\hat{s}$ is a square root of the mean square error at location $\boldsymbol{x}$ and $\Phi(\cdot)$ is the cumulative distribution function.

The algorithm presented in this paper uses a localized approach to define the probability of improvement at an unknown location; for simplicity we define the indicator as the LPoI which at a given point is calculated as follows

$$PoI_{ref}{}^{n}(\boldsymbol{x}) = \Phi\left(\frac{y_t{}^{n}(\boldsymbol{x}) - \hat{y}^{n}(\boldsymbol{x})}{\hat{s}^{n}(\boldsymbol{x})}\right) \tag{2}$$

$$PoI_{ext}{}^{n}(\boldsymbol{x}) = \Phi\left(\frac{y_{ext}{}^{n} - \hat{y}^{n}(\boldsymbol{x})}{\hat{s}^{n}(\boldsymbol{x})}\right) \tag{3}$$

$$LPoI_{ref}(\boldsymbol{x}) = \min\{PoI_{ref}(\boldsymbol{x})\} \tag{4}$$

$$LPoI(\boldsymbol{x}) = \max\{LPoI_{ref}, PoI_{ext}\} \tag{5}$$

where $y_t{}^{n}$, $\hat{y}^{n}$, $\hat{s}^{n}$, $y_{ext}{}^{n}$, $PoI_{ref}{}^{n}$ and $PoI_{ext}{}^{n}$ are the corresponding measures of the n^{th}objective function, $PoI_{ref}{}^{n}$ is the probability of improvement calculated on the basis of reference points, $PoI_{ext}{}^{n}$ is the probability of improvement calculated based on minimum objective function, y_t is the target improvement using reference points, y_{ext} is the target improvement over each individual objective, and finally PoI_{ref} and PoI_{ext} are collections of $PoI_{ref}{}^{n}$ and $PoI_{ext}{}^{n}$, respectively.

The $PoI_{ext}{}^{n}$ term, as described by (3), is due to the fact that the minimum of each individual objective function is always present in the pareto front, hence PoI at each location $\boldsymbol{x}$ over the optimal target of that function is always considered. This term also contributes to the diversification of the pareto front, while the associated improvement target y_{ext} is given by

$$y_{ext}{}^{n} = y_{min}{}^{n} \cdot (1 - p) \tag{6}$$

where $y_{min}{}^{n}$ is the known minimum value of the n^{th} objective function and p is the percentage of improvement, the analytical form of which is shown later by (9).

The term ${PoI_{ref}}^n$ in (2) is calculated at different locations of $\boldsymbol{x}$ based on the local improvement targets; a target is calculated for a reference point, and the reference point is taken based on the location of $\boldsymbol{x}$. In order to obtain the reference points, the algorithm finds the pareto front for existing design sites, using non-dominated sorting, and then for each closest set of pareto points (the number of points equals to the number of objectives), calculates the corresponding reference point. The coordinate of the reference point for each dimension is equal to the maximum value of these pareto points in the same dimension. Hence the reference point in the n^{th} dimension is given by

$$ {y_{ref}}^n = \max\{Y^n\} \tag{7} $$

where Y^n is the collection of the n^{th} objective values of all points in that Pareto set. The improvement target ${y_t}^n$ associated with each reference point is then given by

$$ {y_t}^n = {y_{ref}}^n \cdot (1 - p) \tag{8} $$

The percentage of improvement p controls the convergence rate of the algorithm and is given by

$$ p = p_{initial} \cdot \max\{LPoI_{prev}\} \tag{9} $$

where $p_{initial}$ is the initial improvement target to be defined (it is set to 0.1 in our tests) and $LPoI_{prev}$ is the set of $LPoI$ from all points at the previous iteration. To obtain the next infill sampling point, the algorithm finds the location $\boldsymbol{x}$ associated with the maximum LPoI measure in the criterion space.

The solver based on the LPoI criterion will tend to minimize the localized PoI and converge towards the pareto front. When the design space is well explored, or p is very small, the solver will converge to existing pareto fronts; at this stage it is common for a multiple number of unknown sites to have the LPoI equal or close to 1 (very likely to improve over the target point). In order to obtain a uniformly distributed pareto front, the algorithm selects candidates which have the largest Euclidean distance to existing pareto points as the next infill sampling points. Because of this reason given above, the maximum value of LPoI can be capped at 0.95 or 0.99 for faster exploitation of the existing pareto front.

III. Test Examples and Results

In this section we illustrate the proposed method using a bi-objective example.

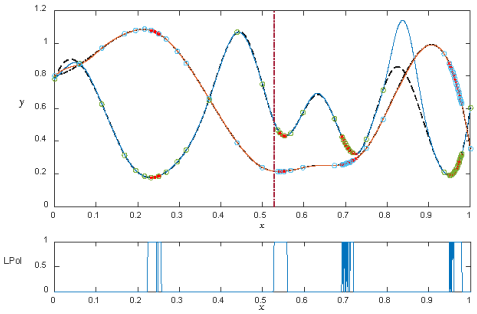
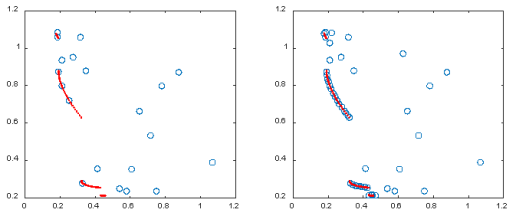


Fig. 1. The kriging model and the LPoI criterion in the search space.



(a) (b)

Fig. 2. (a) Existing design sites in the criterion space at 20th iteration; (b) Existing design sites in the criterion space at 45th iteration.

The top graph in Fig. 1 shows the kriging model (solid line) after the 45th iteration, with the red crosses plotted at the true pareto front, while the bottom plot shows the proposed indicator value for the unknown sites. As can be seen, the algorithm has correctly converged to all four pareto point clusters in the search space and thus further sampling will lead to more exploitation on the pareto front. The sampled design sites in the criterion space are plotted in Fig. 2, where the red dots indicate the location of the true pareto front.

IV. Solving the New TEAM Problem

A new TEAM problem is about to be proposed at the forthcoming Compumag conference in Korea, June 2017. This is devoted specifically to multi-objective optimization and the intention is to test our algorithm on this new benchmark problem; full details will be provided in the extended version.

V. Conclusion

A novel approach to kriging-based multi objective optimization is proposed and details are discussed in this paper. A bi-objective problem is illustrated, while the method will be tested against the new TEAM benchmark problem. The proposed method addresses efficiently both the diversification and uniformity of the pareto solution, is computationally efficient and is linearly scalable to higher number of objectives.

References

[1] J. David Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," Proceedings of the 1st International Conference on Genetic Algorithms, pp. 93-100, 1985.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in Schoenauer M. et al. (eds), Parallel Problem Solving from Nature PPSN VI, 2000.

[3] K.E. Parsopoulos and M.N. Vrahatis, "Particle swarm optimization method in multiobjective problems," SAC '02 Proceedings of the 2002 ACM symposium on Applied computing, pp. 603-607, 2002.

[4] R.T. Marler and J.S. Arora, "Survey of multi-objective optimization methods for engineering," Structural and Multidisciplinary Optimization, vol. 26, no. 6, pp. 369, 2004.

[5] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257-271, 1999.

[6] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," Evolutionary Computation, Proceedings of CEC'02, Honolulu, HI, pp. 711-716, 2002.

`

Appendix B

A summary of the codes used (a copy of all codes attached on a CD).

The MATLAB codes used in this thesis are provided in this section, the content of the media and brief comments are given below, folder names are underlined, while file names are presented in **bold**.

DACE

The software package DACE is a MATLAB kriging tool box; the program files contained in this folder form the prerequisite to most of the other programs in this section.

Chapter 2. Background review

initial_sampling.m

Contains code for random sampling and Quasi-Monte Carlo sampling approach

Kriging_EI.m

Contains code example of kriging with expected improvement (EI) as the infill criterion.

Func_Krig_EI.m

Sub function of **Kriging_EI.m** for calculating standard EI criterion.

Chapter 3. Efficient sampling plan

Kriging_ESS.m

Contains code example for using kriging and efficient sampling scheme as the infill criterion.

Func_Krig_EI.m

Sub function of **Kriging_ESS.m** for calculating standard EI criterion.

Chapter 4. Robust optimisation

Kriging_WCEI.m

Code example for the proposed minmax solver.

`

Func_Krig_WCEI.m

Sub function of **Kriging_WCEI.m** for calculating expected worst-case improvement.

Func_Krig_ED.m

Sub function of **Kriging_WCEI.m** for calculating expected deterioration.

Chapter 5. Kriging for larger datasets

Dualkriging.m

Program code for dual kriging approach and example.

Func_Krig_EI.m

Sub function of **Dualkriging.m** for calculating standard EI criterion.

AGkriging.m

The program code for kriging with aggregation and example.

TestFunc_2D.m

Sub function of **AGkriging.m**, for setting up the test example.

ffh_center.m

Function handler for the centre criterion, sub function of **AGkriging.m**. Use simulated annealing or other global optimisers to optimise this function.

Func_points_aggregation.m

Function handler for the dissimilarity function, sub function of **AGkriging.m**. Use fminbnd or **Script_clustering.m** to optimise dissimilarity.

Script_clustering.m

Script for optimizing dissimilarity; in case the fminbnd optimiser fails, the program will use this function to optimise dissimilarity.

Func_Krig_EI.m

`

Sub function of **AGkriging.m** for calculating standard EI criterion.

Chapter 6. Multi-objective optimisation

MOkriging_LPoI.m

Program code for kriging with the localised probability of improvement (LPoI) for multiobjective optimization problems.

non_dominated_front.m

Sub function of **MOkriging_LPoI.m** for dominate sorting, returning index. Author: Johannes W. Kruisselbrink, this file is a part of the Hypervolume Computation package available in MATLAB file exchange

Func_MO_LPoI.m

Sub function of **MOkriging_LPoI.m** for calculating the LPoI criterion.

Func_plot_v3.m

Sub function of **MOkriging_LPoI.m** for generating figures.

`