Subset selection from large datasets for Kriging modeling

Gijs Rennen

Received: 6 February 2008 / Revised: 6 June 2008 / Accepted: 28 July 2008 / Published online: 24 September 2008
© The Author(s) 2008. This article is published with open access at Springerlink.com

Abstract When building a Kriging model, the general intuition is that using more data will always result in a better model. However, we show that when we have a large non-uniform dataset, using a uniform subset can have several advantages. Reducing the time necessary to fit the model, avoiding numerical inaccuracies and improving the robustness with respect to errors in the output data are some aspects which can be improved by using a uniform subset. We furthermore describe several new and current methods for selecting a uniform subset. These methods are tested and compared on several artificial datasets and one real life dataset. The comparison shows how the selected subsets affect different aspects of the resulting Kriging model. As none of the subset selection methods performs best on all criteria, the best method to choose depends on how the different aspects are valued. The comparison made in this paper can be used to facilitate the user in making a good choice.

Keywords Design of computer experiments • Dispersion problem • Kriging model • Large non-uniform datasets • Radial basis functions • Robustness • Space filling • Subset selection • Uniformity

1 Introduction

1.1 Motivation

Kriging is an interpolation technique which finds its roots in geostatistics. The method is named after Krige, a South-African mining engineer, and is based on his work at the Witwatersrand reef complex (Krige 1951). In the 1960s, the French mathematician Matheron formalized Krige's method (Matheron 1963). Besides geostatistics, Kriging has also found numerous applications in other fields. Sacks et al. (1989) applied Kriging in the field of deterministic simulation for the design and analysis of computer experiments. Since then many others followed; see Jones et al. (1998), Jones (2001), Koehler and Owen (1996), Santner et al. (2003), Stehouwer and den Hertog (1999). A basic description of Kriging can be found in Appendix 5.

When building a Kriging model, the general intuition is that using more data will always result in a better model. Therefore, a large dataset is regarded as a good starting point for building a model. However, when the dataset we can use is already given, there are situations where using only a subset has certain advantages. Especially when the large given dataset is non-uniformly distributed over the whole design space, problems can occur. In this paper, we analyze and test methods to select subsets in order to reduce these problems.

Large non-uniform datasets can occur in several situations. The first situation we can think of is when we have a set of legacy data (Srivastava et al. 2004). Legacy datasets contain results of experiments, simulations or measurements performed in the past. These results are stored for future use to avoid having to generate them again. This is especially useful if there are many results

G. Rennen (✉)
Department of Econometrics and Operations Research,
Tilburg University, P.O. Box 90153, 5000 LE Tilburg,
The Netherlands
e-mail: G.Rennen@uvt.nl

or if the expenses to obtain these results are high. As this data is not generated especially for making a global model, it is often not uniformly distributed over the whole design space. Another reason for non-uniformity of legacy datasets can be that they contain measurements of a system which cannot be fully controlled.

A second situation is when the data is the result of a sequential optimization method. These methods often generate more data points near potential optima than in other regions (Booker et al. 1999). If we want to use this data to fit a global metamodel, we thus have to take into account that it contains clusters of points.

Thirdly, non-uniform data occurs if models are coupled (Husslage et al. 2003). We call two models coupled if the output of the first model is input for the second. If we want to construct a metamodel of both models, we can use a space-filling design of experiments for the first model. It could be argued that we could also do this for the second model. In some cases, however, it is better to use the output of the first model. Although the input of the first model is space-filling, its output is often not. When we want to construct a metamodel of the second model, we thus have to use a non-uniform dataset.

These are just a number of situations where we can come across large non-uniform datasets. Using these sets directly to build our model can impose problems which can often be resolved by using a uniform subset. We call a subset uniform if the input data of the data points in the subset are “evenly spread” over the entire design space. Whether or not using a uniform subset is better, depends not only on the dataset but also on the chosen modeling method.

Important reasons for using a subset instead of the complete dataset are the following:

- **Creating training and validation set**

The most common reason for using only part of the data as training data is validation. Splitting a dataset into a training and a validation set is a well-known validation method and is often done randomly (Cherkassky and Mulier 1998; Golbraikh and Tropsha 2002). In the field of design of computer experiments (DoCE), however, a consensus has been reached that designs used for deterministic computer experiments should be space-filling (Simpson et al. 2001). A design is called space-filling if it fills the whole design space. For a given number of design points, the design space is best filled if the design points are evenly spread over the whole design space. Therefore, it would be a good idea to also take a uniform subset when selecting a training set from an existing dataset (Golbraikh et al. 2003).

- **Time savings**

A second reason could be the reduction in time necessary to fit the Kriging model. This is certainly an important issue as its time-consumption is generally regarded as one of the main drawbacks of the Kriging method (Jin et al. 2001). With current implementations and computing power, it is sometimes even impossible to fit a Kriging model using all available data. For instance, when using the Matlab toolbox DACE provided by Lophaven et al. (2002) on a PC with a 2.4-GHz Pentium 4 processor, we encountered problems for datasets containing 3000 points or more.

Especially the inversion of the correlation matrix can impose problems as this requires much time and memory capacity. Kriging is, however, not the only method which can benefit from using less data. Genetic programming could also significantly benefit as it requires a lot of models to be fitted to a training set (Koza 1992; Banzhaf et al. 1998). Often this forms a large part of the total computation time and is thus worthwhile to reduce.

Besides fitting the Kriging model, the prediction of new points also requires less time because using less training data results in simpler models. This is mainly because the number of terms in the Kriging model depends on the size of the training set. The same holds for Lagrange interpolation. Especially in situations where the model is used for on-line monitoring and optimization, finding a fast and simple model is important (Kordon 2006).

- **Avoiding numerical inaccuracies**

A common property of large datasets is that they are non-uniform, which implies that they can contain points that lie very close together. This property can make the corresponding correlation matrix ill-conditioned (Davis and Morris 1997; Booker et al. 1999). Solving a linear system with an ill-conditioned matrix can cause significant numerical inaccuracies. The optimization of the Kriging parameters requires solving a linear system and can thus be inaccurate when data points lie close together. Removing certain points from the dataset can avoid the correlation matrix from becoming ill-conditioned and can, therefore, improve the numerical accuracy of the Kriging model.

- **Improving robustness**

Robustness of the Kriging model with respect to errors in the output data can be negatively influenced when data points lie close together. Siem and Den Hertog (2007) show that points that are close together can get assigned relatively large Kriging weights. Due to these large weights, errors in the

output values at these points can have a large influence on the output value of the Kriging model at other points. Removing some of the points can result in lower weights for the remaining points and thus a smaller effect of errors. Therefore, we can sometimes improve robustness by using only a subset instead of the whole dataset.

These different motivations for subset selection require us to look at different performance criteria. In Section 4.2, we describe the criteria used in this paper to measure the effects of subset selection on these different factors.

For most of the mentioned motivations, it is important that the selected subset is uniform. Selecting a uniform set of points also occurs in other problems like Design of Computer Experiments (DoCE) and the dispersion problem. Although these are different problems, we can use some ideas from the fields of DoCE and dispersion problems in determining our subset. To see why and how we can use these ideas, we look at the similarities and differences between the problems.

1.2 Design of computer experiments

The most important reason why we can use ideas from DoCE is that it has the same aim as subset selection. In both problems the aim is to select the training set that will produce the model that most accurately approximates the function or process underlying the data. However, in practice we do not have an explicit description of the underlying function or process, which makes it impossible to directly optimize this objective. Instead, we optimize certain properties of the training set that generally improve the quality of the resulting model. Of these properties, the most frequently used is space-fillingness (Simpson et al. 2001).

Another similarity is that in both problems we often have a restriction on the number of points we can use. In DoCE the restriction is caused by the large computation time per design point. The reasons for limiting the number of points by subset selection are given in Section 1.1.

Besides similarities, there are also differences between the two problems. The first difference is the set of points we can choose from. With DoCE, we can select all points in the design space. Sometimes we choose to restrict ourselves to points with a certain structure or property, like a Latin hypercube (McKay et al. 1979; Stein 1987) or an orthogonal array (Owen 1992; Tang 1993), but even then we are free in selecting which structure or property. With subset selection, we can only choose from the points in the original dataset. This

implies, for instance, that the subset can only cover the design space as good as the original dataset does. It could be argued that sometimes additional points can be evaluated to improve for instance uniformity. Nevertheless, we will not take this option into account as it is beyond the scope of this paper.

The second difference is that the output values of all possible points are known in the case of subset selection. This in contrast to DoCE where we have to select our points without knowing their output values. Even in sequential DoCE, only the output values of previously selected and evaluated points are known. Using the output information in subset selection will most likely result in training sets which give more accurate models.

These differences show that we cannot directly use results from DoCE. In the paper by Srivastava et al. (2004), however, a method is described that uses a space-filling DoCE to determine a uniform subset. The main idea is to take a randomized orthogonal array and to select for each point of this orthogonal array, the data point closest to it. This idea can also be applied to other types of space-filling DoCEs. However, to limit the number of methods compared in this paper, we will only consider orthogonal arrays. In Section 3.1, the method of Srivastava et al. (2004) is described in more detail.

1.3 Dispersion problems

The dispersion problem can be described as follows. Given n possible locations, locate $m < n$ facilities such that some function of the distances between the facilities is maximized (Ravi et al. 1994). Two commonly used functions are the minimum and the sum of the distances (Erkut and Neuman 1989). The first case is often referred to as MAXMIN and the second as MAXSUM. Both versions of the dispersion problem are strongly NP-hard. This was independently proven by Erkut (1990) and Ghosh (1996) for the MAXMIN problem and by Hansen and Moon (1994) and Kuo et al. (1993) for the MAXSUM problem. MAXMIN is also used as a measure of space-fillingness in DoCE. When we see locations as points, we can thus reformulate the problem as follows. Given n possible points, select a uniform subset of m points.

If we only focus on uniformity when selecting our subset and use MAXSUM or MAXMIN to measure the uniformity then subset selection is the same as the MAXSUM or MAXMIN dispersion problem. Even so, this does not mean that we can directly apply all facility location algorithms and heuristics to the problem of subset selection. The two main reasons for this are the following.

Firstly, we have to consider the dimensionality of the design space. In dispersion problems, the locations are often points in two- or three-dimensional space. Subset selection on the other hand applies to datasets of any dimension. In practice, the dimension is often considerably larger than 3. This means that we have to determine whether an algorithm for the dispersion problem extends to higher dimensions, before we can use it in subset selection.

Secondly, many algorithms only give a solution in a reasonable amount of time for sets containing relatively few (a couple of dozen or a few hundred) points (Agca et al. 2000; Pisinger 2006). Subset selection on the other hand is generally applied when there are thousands of points. Ideas which give no computational problems for many dispersion problems could require too much time or memory capacity to be useful for subset selection problems.

Keeping these two aspects in mind, we can apply some methods for the dispersion problem directly to subset selection. The greedy MAXMIN method (Ravi et al. 1991) is one such method. It starts by selecting the two points furthest away from each other. Then iteratively, the point furthest away from the already selected points is added to the subset. Although this method is quite simple, Ravi et al. (1991) have shown that if the triangle inequality holds, the heuristic gives an approximation ratio of 2. This means that we can guarantee that the MAXMIN distance of the subset we obtain is at most twice the MAXMIN distance of the optimal subset. Furthermore, due to its simplicity the method is quite fast and thus suitable for large datasets. Therefore, we also use this method for uniform subset selection and describe it in more detail in Section 3.3.

1.4 Overview

The structure of the rest of this paper is as follows. In Section 2, we show with a simple example that taking a subset can indeed improve the aspects mentioned in this introduction. New and current methods to select a subset are described in Section 3. The main difference between the new and current methods is that the new methods also use the output data. To determine if this results in subsets which produce better Kriging models, we make a comparison between the methods in Section 4. Furthermore, we make a comparison with radial basis function (RBF) models fitted to the complete dataset. The advantage of RBF-models is that they have shown good fits to both stochastic and deterministic functions (Powell 1987) and that they require significantly less time to be fitted than Kriging models (Jin et al. 2001). RBF-models can therefore be fitted to

datasets for which Kriging models would be too time-consuming. In Section 4, we thus also compare the performance, in terms of time-consumption and accuracy, of Kriging models fitted to a subset and RBF-models fitted to the complete dataset. A basic description of RBF-models can be found in Appendix 6. Finally, Section 5 contains the conclusions and suggestions for further research.

2 Example

To show that selecting a subset can really improve the aspects mentioned in Section 1.1, we introduce the following simple artificial example. We try to approximate the six-hump camel-back function (Dixon and Szegö 1978):

$$f(x) = 4x_1^2 - 2.1x_2^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

with $x_1 \in [-2, 2]$ and $x_2 \in [-1, 1]$. As our dataset, we take a maximin LHD of 20 points (van Dam et al. 2007) with four additional points close to an existing point as depicted in Fig. 1. By adding these four points, we have created a cluster of points in the dataset. As mentioned in Section 1.1 this can cause several problems.

We test the effect of selecting a uniform subset by fitting Kriging models to the datasets with and without the four additional points. To measure the effects of taking a subset on the different aspects, we use the performance measures which are described in Section 4.2. The results of the performance measures of both Kriging models are given in Table 1. For all measures it holds that a lower value is better.

Fig. 1 Maximin LHD of 20 points with four additional points

Table 1 Performance of Kriging models fitted to the datasets with and without the four additional points

	Without four additional points	With four additional points
RMSE	0.48	0.51
Maximum error	2.03	2.46
Condition number	76	1,491,564
Average robustness	0.97	8.36
Max. robustness	1.57	100.66

Both the RMSE and the Maximum Error show that the accuracy of the Kriging model without the four points is better than that of the model with the additional four points. The latter Kriging model focusses more on fitting accurately in the region of these additional points and is, as a result, more accurate in this region. However, this additional accuracy is at the expense of accuracy in other regions, which deteriorates the overall accuracy.

When we compare the condition numbers, we see a very large difference. This shows that the additional 4 points make the resulting Kriging model much more susceptible to numerical inaccuracies. Finally, the larger maximal and average robustness values indicate that this model is also less robust with respect to errors in the output data.

This simple example thus shows that removing some points can improve the quality of the resulting Kriging model. Determining which points to remove is quite easy in this case. In practice when the number of points is much larger, this becomes less straightforward. Therefore, we introduce in the next section some current and new methods to select points from a dataset.

3 Subset selection methods

3.1 Orthogonal array selection

In the paper by Srivastava et al. (2004), the problem is discussed of selecting 500 or fewer points from a dataset containing 2,490 points in 25-dimensional space. The selected points are used to create a Kriging model and the remaining points are used to check the accuracy of this model. Results are compared for different numbers of selected points and with results obtained by using quadratic response-surface models.

To select the points, first a randomized orthogonal array is constructed. Then for each point of the orthogonal array a “nearest neighbor” is determined, i.e. the data point closest to the orthogonal array point. All “nearest neighbors” together form the set of selected

points. It is possible that this set contains less points than the orthogonal array because one point in the dataset can be the “nearest neighbor” of multiple points of the orthogonal array. As we do not know in advance how often this happens, we cannot set the subset size exactly. However, we do know that the size of the orthogonal array is an upper bound for the subset size.

The idea of selecting “nearest neighbors” to points of an orthogonal array can be extended to other types of space-filling DoCEs. We could for instance also use maximin LHDs. In order to limit the number of methods compared in this paper, we do not use Maximin LHD Selection as a separate method. We do, however, use it to generate starting solutions for the Sequential Selection methods described in Section 3.5. The (approximate) maximin LHDs we use for this are obtained with the ESE-algorithm of Jin et al. (2005).

Another reason for using Orthogonal Array Selection is that it enables us to compare our results with those found by Srivastava et al. (2004). It is, however, unclear which method they exactly used to construct the randomized orthogonal arrays. We have decided to use the method described on page 131 in Hedayat et al. (1999) to construct 61 and 113-dimensional orthogonal arrays of respectively 250 and 686 points. Using a uniform distribution, we randomly select six or 25 dimensions to obtain randomized orthogonal arrays that are suitable for our test problems. As different orthogonal arrays can result in different subsets, the choice of orthogonal array could affect the quality of the resulting subset. We therefore determine for each dataset ten subsets using ten different orthogonal arrays. We thus hope to determine whether the choice of orthogonal array indeed affects the quality of the subset and the resulting model.

Notice, that this method requires us to find a suitable orthogonal array. This could be a problem as an orthogonal array of the desired number of points and dimensions might not be known. However, due to the flexibility in the desired number of points of the orthogonal array, this problem can in most cases easily be resolved.

3.2 Fast exchange algorithm

The Fast Exchange Algorithm was introduced by Lam et al. (2002) to select a subset from a very large database containing characteristics of molecules. The aim of the algorithm is to select a subset such that it covers the numerical space described by the database. However, as this space is often high-dimensional, nearly all of these databases are sparse. This makes it impossible to

densely cover the whole space with a reasonably sized subset of points. The algorithm therefore focusses on selecting the subset such that it is space-filling in low-dimensional projections of the space.

The algorithm globally works as follows. First, sets of bins are constructed for all 1-D, 2-D and 3-D projections. These bins form a partition of the projected design spaces and their sizes depend partly on the distribution of the data. The aim is now to select a subset such that each bin contains approximately the same number of points. To measure how close a certain subset is to this aim, the uniform cell coverage criterion is used. This criterion has the advantage that we can easily calculate the effect of adding or removing a particular point from the subset on the criterion value.

The Fast Exchange Algorithm now tries to find the best subset by exchanging points that are inside and outside the subset. This is done in two steps. In the first step, the best point to add to the current subset is determined. The resulting subset thus has a size of $n + 1$. In the second step, the best point to remove from this subset is determined. The main difference with the basic exchange algorithm is in these two steps. In the basic exchange algorithm, all possible points outside the subset are tried in the first step and all points inside the subset in the second step. As a result, the basic exchange algorithm loops through all points in the dataset before an exchange is made. The Fast Exchange Algorithm, on the other hand, uses a distribution of the improvements to select the exchange. At the begin of the algorithm, 100 additions and removals of points are tried to estimate the two distributions of the improvements in Steps 1 and 2. During the algorithm, when more additions and removals are tried, this estimated distribution is updated. A point is now added or removed if it belongs to the upper tail of this distribution. This often implies that we loop through a lot less points before an exchange is made.

3.3 Greedy MAXMIN selection

The greedy MAXMIN method comes from the field of dispersion problems. In the description of the algorithm, k -dimensional data points are therefore seen as points in k -dimensional space. As the name indicates, it seeks to maximize the minimal Euclidean distance between any two points in the chosen subset. It does this in essentially the same way as the “furthest point outside the neighborhood” heuristic described in Steuer (1986).

Let us denote the total dataset by N and the Euclidean distance between points i and j by $d_{i,j}$. We

can then describe this heuristic as follows: (Ravi et al. 1991)

1. Take $S = \emptyset$.
2. Let (i, j) be such that $d_{i,j}$ is maximal.
3. Add i and j to the set S .
4. Find a point $i \in N \setminus S$ such that $\min_{j \in S} d_{i,j}$ is maximum among the points in $N \setminus S$.
5. Add point i to S .
6. Repeat Step 4 and 5 until the set S contains the desired number of points.

In Step 4, we thus determine the point furthest away from the already chosen points. Although the heuristic is quite simple, Ravi et al. (1991) have shown that if the triangle inequality holds, the heuristic gives an approximation ratio of 2. Furthermore, they have proven that, unless $P=NP$, no polynomial-time relative approximation algorithm can provide a better performance.

Notice, that if the domains of the input variables are of different magnitude, it is better to normalize the domains in order to obtain a uniform subset. For all datasets used in this paper, we will therefore normalize the domains before applying this method.

3.4 Greedy DELETION algorithm

Besides MAXMIN Selection, we use another simple greedy algorithm. This greedy method constructs a subset by iteratively removing one point of the pair of points with the smallest Euclidean distance between them. To decide which of the two points should be removed, we look for each point at the distance to its second closest point. The point for which this distance is smallest is removed from the dataset.

Using the same notation as for MAXMIN Selection, we can describe the DELETION Algorithm as follows:

1. Take $S = N$.
2. Let (i, j) be such that $d_{i,j}$ is minimal.
3. Determine $c_i = \min_{k \in S \setminus \{i, j\}} d_{i,k}$ and $c_j = \min_{k \in S \setminus \{i, j\}} d_{j,k}$.
4. Remove the point with the lowest c value from the set S .
5. Repeat Steps 2, 3 and 4 until the set S contains the desired number of points.

As this method constructs a subset by removing points from the total dataset, the method requires less iterations for larger subsets than for smaller. This in contrast to for instance MAXMIN and Sequential Selection which build a subset by adding points to an initially empty set. Furthermore, notice that for this method it is useful to normalize the domains of the variables in order to obtain a uniform subset.

3.5 Sequential selection

In the three methods discussed in the previous section, the output values of the points are not used in the selection process. However as the selected points are used to determine a model of the output, it seems a good idea to explicitly use the output information in selecting the training points. As previously mentioned, this is an important difference with DoCE where generating the output values is expensive. In our situation, we work with a given dataset with known output values and can thus use the output values at no additional computational costs.

We use the output values in the following way. First, we apply Maximin LHD Selection (see Section 3.1) using only the input values to determine an initial training set. Then we fit a Kriging model to the training set and calculate the prediction error at the non-selected points. The idea is now to add non-selected points with a large error to the training set. We should, however, take into account that if the Kriging model is inaccurate in a certain region, all points in that region have a large error. Consequently, simply selecting, for instance, n_1 points with the highest error might result in adding points that are clustered together in one or a couple of regions.

To reduce this problem, we use two methods. The first method is to add one point at a time. This means that the point with the largest error is added to the training set and then the Kriging model is refitted. This method completely solves the above problem, but is unfortunately quite time consuming as we fit a new Kriging model after every added point. This is the reason that we developed a second method. This method determines the $n_2 > n_1$ worst points and then uses the greedy MAXMIN heuristic described in Section 3.3 to select a uniform subset of n_1 points. These n_1 points are then added to the training set. In this paper, we use $n_1 = 10$ and $n_2 = 40$ to test this method. We have not tested the influence of the choice of n_1 and n_2 on the quality of the resulting subset. Determining this possible influence and a suitable method for selecting these values thus remain topics for further research. To determine whether the above problem really occurs and results in worse models, we also test the method where we simply add the $n_1 = 10$ worst points.

For each method, the selection and addition step are repeated until the desired number of points are selected. Notice, that because we use Maximin LHD Selection to determine the initial training set, the initial training set might not always be equal to the size of the LHD. We take this into account when determining the number of selection and addition steps. As for two

methods we add 10 points per step, it is not always possible to determine this such that we exactly get the desired number of points. In these cases, we select the largest attainable number of points smaller than the desired number of points.

As it is a sequential method, we can decide to use another stopping criterion to determine how often we repeat the steps. An example would be to stop when the maximum error in the non-selected points is below a certain predefined value. In this paper, we choose to predefine the number of selected points for easier comparison between the different methods.

Finally, we want to make a remark on the implementation of this method. As mentioned, the method requires that a new Kriging model is fitted in every iteration. As this is the most time consuming part of the method, we tried to speed up the optimization of the Kriging parameters. We managed to do this by using the fact that subsets in consecutive iterations are quite similar as one is obtained by adding points to the other. This fact makes it quite plausible that the optimal parameter settings of the Kriging models fitted to both subsets are also quite similar. The optimal parameter setting of the Kriging model in the previous iteration therefore seems a good starting solution for fitting the Kriging model in the current iteration. Some tests show that using the optimal parameters of the previous iteration instead of a fixed starting solution, can indeed reduce the time needed to select a subset by 15 percent. Therefore, we use this implementation for the Sequential Selection method in the remainder of the paper.

4 Computational results

4.1 Tested subset selection methods

The methods that we will compare are the following:

- Random Selection (RS).
- Orthogonal Array Selection (OAS).
- Fast Exchange Algorithm (FEX).
- Greedy MAXMIN Selection (MAXMIN).
- Greedy DELETION Algorithm (DELETION).
- Sequential Selection adding one point at a time (SS1).
- Sequential Selection adding ten worst points at a time (SS10).
- Sequential Selection adding ten uniform points from the 40 worst points at a time (SS1040).

The first method randomly selects the required number of points without taking into account any of their

properties. The performance of this method is used as a reference for the performance of the other methods.

To test the effect of the training set size on the resulting Kriging model, we select subsets of 250, 350 and 500 points from the datasets described in Section 4.3. As the SS1 method is quite time consuming, we will only generate subsets of 250 and 350 points using this method. Furthermore for Orthogonal Array Selection, we cannot determine the exact subset size in advance. By using orthogonal arrays of 250 and 686 points, we get subsets of at most these amounts of points. As mentioned in Section 3.1, we generate subsets using ten different orthogonal arrays to test the effect of the choice of orthogonal array on the resulting subset. For each performance measure, we therefore report the mean, minimum, and maximum over these ten subsets.

4.2 Performance measures

The different motivations for selecting a subset require that we use several performance measures to determine how good a certain subset is. We describe the used performance measures and our motivation for choosing them.

4.2.1 RMSE and maximum error

Our first motivation is the selection of a subset that results in an accurate model. We thus need to measure the accuracy of the resulting Kriging model. Common measures are the average error, root mean squared error (RMSE), and maximum error. We cannot measure these on the training data because the Kriging model will interpolate through the training data. Instead, we need a validation set which could either be the remaining dataset or a separately generated set.

In the case of a real-life dataset, we only have the first option. If the original dataset is non-uniform, the remaining dataset might not be particularly suited to measure the overall accuracy of the model as the accuracy in densely populated regions weighs heavier than accuracy in sparsely populated regions. This problem could be reduced by taking the weighted average error or weighted RMSE. The weights should then be determined such that errors in points in sparse regions get more weight than errors in points in dense regions. It is, however, unclear how exactly the weights should be determined to achieve this effect. Therefore, we will use the usual RMSE in this paper, although we are aware of its deficiency. Finding a method for determining the weights remains a worthwhile subject for further research. Besides the RMSE, we will also

use the Maximum Error to compare the accuracy of different Kriging models.

For artificial datasets, the best option is to separately generate a uniform validation set. This way we avoid any problems with densely and sparsely populated regions. In this paper, a uniform grid is therefore used as the validation set.

4.2.2 Time model fitting

To determine the reduction in time necessary to fit the Kriging model, we simply use the amount of required CPU-time. For fitting the Kriging model, we used the Matlab toolbox DACE provided by Lophaven et al. (2002). All of the calculations were performed on a PC with a 2.4-GHz Pentium 4 processor. The results in this paper are all reported in minutes.

4.2.3 Time subset selection

Besides fitting the Kriging model, the time necessary for constructing the subset is also measured. This is needed to determine if the time gained for model fitting is not outweighed by the additional time needed to select a good subset. Note that all subset selection methods are coded by the author in Matlab. By using the same program for each method, we aim to make a fair comparison between the different methods. The results are again reported in minutes.

4.2.4 Condition number

We also want to avoid ill-conditioned correlation matrices by selecting a subset. To determine if a correlation matrix is ill-conditioned, we can use its condition number. The condition number κ of a square matrix C is defined as (Golub and van Loan 1996):

$$\kappa(C) = \frac{\sigma_{\max}(C)}{\sigma_{\min}(C)},$$

where $\sigma_{\max}(C)$ and $\sigma_{\min}(C)$ are the largest and smallest singular values of C . The condition number is a measure of the worst case loss in precision when solving a linear system. A matrix with a large condition number is called ill-conditioned and is thus susceptible to numerical inaccuracies.

4.2.5 Average and maximum robustness

Finally, we want to measure the influence of errors in the output data on the resulting Kriging model. Depending on the kind of data we are dealing with, these errors can have different causes. If, for instance,

the data is obtained using a simulation model, the error in the output can result from model errors or numerical errors. If a Kriging model is robust, a small inaccuracy in the output data does not cause the Kriging model to deviate much from the Kriging model based on the correct data.

To measure the robustness of a Kriging model with respect to errors in the output data, Siem and Den Hertog (2007) suggest to use $\|c(x)\|$ as a measure of robustness at the point x , where $c(x)$ is the vector of Kriging weights and $\|\cdot\|$ is the Euclidean norm (see Appendix 5 for a basic description of the Kriging method). The general idea behind this measure is the following. Assume, we have a training dataset $x^1, \dots, x^n$ for which the real output values are $y^1, \dots, y^n$, but the measured output values are $y^1 + \varepsilon^1, \dots, y^n + \varepsilon^n$. An upper bound for the deviation from the correct Kriging model at point x is then approximated by $\|\varepsilon\| \|c(x)\|$ where $\varepsilon = [\varepsilon^1 \dots \varepsilon^n]^T$. The measure $\|c(x)\|$ thus gives an indication of the factor by which errors in the output values of the dataset are multiplied when calculating the output value of the Kriging model at point x .

This robustness-criterion only determines the robustness at a certain point x . To measure the overall robustness, we use the maximal and average value of the robustness values at a set of points. As we do not need to know the real output values at these points, we will select them on a uniform grid even in the case of a real-life dataset.

4.3 Datasets

To test the different selection methods, we use two types of datasets: artificial datasets with known underlying function and real-life datasets for which the underlying function is unknown. For real-life datasets, it is clearly more difficult to judge the accuracy of the resulting model, especially as these datasets are often unstructured and non-uniform. This also holds for the real-life dataset used in this paper. The dataset was originally used in the design of the High Speed Civil Transport (HSCT) aircraft and contains 2,490 points in 25-dimensional space (Padula et al. 1996). As this dataset is used by Srivastava et al. (2004), we use it to make a comparison with their results. Before using it, we first removed the duplicate points which left us with a dataset of 2,487 unique points.

Artificial datasets are generally constructed by drawing points from the design space and calculating their value for a known function. In this paper, the six-variable Hartman-6 function (Dixon and Szegö 1978)

is used to generate the output data. This function is defined as follows:

$$f(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^6 \alpha_{ij} (x_j - p_{ij})^2 \right)$$

with $x_j \in [0, 1]$, $j = 1, \dots, 6$, and where the parameters are given in the following table:

i	$\alpha_{ij}, j = 1, \dots, 6$					
1	10	3	17	3.5	1.7	8
2	0.05	10	17	0.7	8	14
3	3	3.5	1.7	10	17	8
4	17	8	0.05	10	0.1	14

i	c_i	$p_{ij}, j = 1, \dots, 6$					
1	1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

The output values of the Hartman-6 function range between approximately -3.23 and 0 . The Hartman-6 function is chosen because it is a widely used test problem with a relatively large number of dimensions (Wang et al. 2001; Jin et al. 2002). As most real-life datasets are high dimensional, we prefer this test problem over other widely used but lower dimensional test problems.

To draw the points from the design space, the following methods are used:

1. Draw the values of all coefficients from a uniform distribution on the range of each variable.
2. Divide the design space into m^d equally sized cells, where d is the dimension. Determine m^d numbers that sum up to the total size of the dataset. Randomly assign these numbers to the cells. For each cell use a uniform distribution to sample the assigned number of points within the cell. By adjusting the distribution of the number of points, we can vary the uniformity of the dataset.

We refer to datasets constructed with the first method as uniform datasets. We use the second method to construct non-uniform datasets of 2,000, 5,000, and 10,000 points. For all sizes, m is set equal to 3 which means that we have 729 cells. The distributions of the number of points per cell are given in Table 2.

For 2,000 points, it is still possible to fit a Kriging model and a RBF-model to the complete dataset.

Table 2 Distributions of the number of points per cell used to construct non-uniform datasets of 2,000, 5,000, and 10,000 points

Number of cells	30	30	74	238	431	595
Points per cell 2,000 points	20	10		2	1	
Points per cell 5,000 points	50	25	5			4
Points per cell 10,000 points	100	50	10			8

Therefore, we use these datasets to test if taking a subset is really better than using the complete set. For the datasets of 5,000 points, we were no longer able to fit a Kriging model to the complete dataset with the DACE toolbox. A RBF-model could, however, still be fitted to the complete dataset. Hence, we use these datasets to test how a Kriging model fitted to a subset compares to a RBF-model fitted to the complete set. When using 10,000 points, we could no longer fit a RBF-model to the complete dataset. We thus use these datasets to compare the results of the different subset selection methods for Kriging only.

For each size and type of artificial dataset, we randomly generated 50 datasets. In the comparisons, we use the average, minimum, and maximum performance over these datasets for all performance measures.

4.4 Results for artificial datasets of 2,000 points

For datasets of 2,000 points, it is still possible to fit a Kriging and a RBF-model to the complete dataset. By taking a subset, we aim to improve the different performance measures. In Table 3 in Appendix 1, we have an overview of the results of the performance measures.

When looking at the RMSE and the Maximum Error, the Kriging model fitted to the complete dataset is the most accurate model for both uniform and non-uniform datasets. Unfortunately, it requires by far the most time to fit. Fitting a RBF-model is much faster than Kriging and also faster than most subset selection methods if we take into account the times necessary to select the subsets. However, some of the Kriging models fitted to a subset are more accurate than the RBF-model. For subsets of 200 points, the RMSE of most Kriging models is still worse, but SS1, SS10 and SS1040 already produce almost equally good or even slightly better Kriging models. For 350 points SS1, SS10 and SS1040 result in Kriging models with a really lower RMSE than the RBF-model. When using 500 points, these methods result in Kriging models with even lower RMSE values, as expected. For non uniform datasets, also the best subsets obtained with OAS result in Kriging models with a lower RMSE. The Maximum Error shows even more cases where Kriging models fitted to subsets are better than the RBF-model.

When we consider the subset selection times, two methods that require hardly any time are RAND and OAS. Nevertheless, using RAND is not a very good choice, as the resulting Kriging models are quite inaccurate. The OAS method results in considerably more accurate Kriging models. Subset methods which result in even more accurate Kriging models are SS1, SS10 and SS1040. When we compare these methods for each subset size separately, SS1 results in the lowest RMSE and SS1040 in the lowest Maximum Error. When we, however, take into account the time to select the subset, SS1 is the least favorable of these methods. Selecting a subset of 350 points even requires more time than fitting a Kriging model to the complete dataset. The methods SS10 and SS1040 are also relatively time-consuming compared to RAND, MAXMIN, FEX and OAS, but they are more accurate and have still a considerable time reduction compared to Kriging fitted to the complete dataset.

Considering the condition numbers, SS1, SS1040, MAXMIN, DELETION and OAS perform very well. The good performance of the latter two methods might be explained by the fact that they only focus on optimizing the uniformity of the input data which influences the condition number of the correlation matrix. In contrast, it is remarkable that the FEX method, which has the same focus, performs considerably worse. This seems to indicate that the FEX does not succeed as well in selecting a uniform subset as SS1, SS1040, MAXMIN, DELETION and OAS do. Furthermore, it is nice to see that SS1040 gives better result than SS10 as expected. Taking 10 uniformly selected points from the 40 worst instead of just taking the 10 worst, thus seems to result in a more uniform subset. Most important, however, is that all subset selection methods result in a correlation matrix with a considerably lower condition number than that of the correlation matrix of the complete dataset. Therefore, taking a dataset clearly reduces the chance of numerical inaccuracies.

For robustness, the same conclusion can be drawn that taking a subset considerably improves the performance measure. When we look at the maximum robustness, SS1, SS1040, MAXMIN, DELETION and OAS perform best. For the average robustness also SS10 performs well.

Besides these performance measures, we present the real sizes of the subsets. We do this because for OAS the number of points in the subset is not known exactly in advance. The results show that for 250 points, the number of selected points is quite close to the size of the orthogonal array. For the orthogonal arrays of 686 points, the difference is much larger. As mentioned in Section 3.5, SS10 and SS1040 can result in slightly

different subset sizes as Maximin LHD Selection is used for the initial training set. Even so, the results show that this does not occur for the tested datasets.

When we make a comparison between uniform and non-uniform datasets, we see almost the same results for most performance measures. The ranking of the different subset selection methods is generally the same. The main differences are between the condition number and robustness of the Kriging models fitted on the complete dataset. On these aspects, the Kriging models fitted on the uniform datasets performs better than the Kriging models fitted on the non-uniform datasets. For the Kriging models fitted on the subsets, the differences on these aspects are much smaller. This seems to indicate that the difference in uniformity between the subsets is smaller than the difference in uniformity between the two types of complete sets.

A factor which also influences the performance measures is the subset size. When we compare the performance measures, we see two opposite effects. The RMSE and Maximum Error improve when larger subsets are used and the other performance measures worsen. To determine the 'best' subset size, we thus have to make a trade-off between accuracy and robustness, required time and numerical accuracy. Which trade-off is best depends on the importance of these aspects and the estimated accuracy of the dataset.

Finally, we take a look at the effect of the choice of orthogonal array on the results of the OAS method. We do this by comparing the mean, minimum, and maximum of the performance measures over ten subsets obtained with ten different orthogonal arrays. The differences in these results show that the choice of orthogonal array has a clear effect on the quality of the resulting subset and Kriging model. When using the OAS method, it is therefore better not to use just one random orthogonal array. It would be better to carefully choose a suitable orthogonal array or to create multiple subsets using different (random) orthogonal arrays and then to select the best subset. Which option works best and how exactly we should implement them is not immediately clear and is thus an interesting subject for further research.

4.5 Results for artificial datasets of 5,000 and 10,000 points

For datasets of 5,000 points, we could no longer fit a Kriging model to the complete dataset. A RBF-model could, on the other hand, still be fitted. As most results are quite similar to the results of the datasets containing 2,000 points, we mainly discuss the differences. Table 4 containing the results can be found in Appendix 2.

When comparing the RMSE, we see that now only SS10 and SS1040 with 500 points result in Kriging models with approximately equal RMSE values as the RBF-model. The Maximum Error is, however, considerably lower for all subsets of SS1, SS10 and SS1040 and for the large subsets obtained with OAS. Combining these two measures, the Kriging models based on subsets of 500 points obtained with SS10 and SS1040 seem to be the most accurate. The time required to select these subsets and to fit the model is unfortunately considerably more than for the RBF-model. We thus have a trade-off between accuracy and the required time. For the condition number and robustness, we see the same result as for the datasets of 2,000 points. Considering the subset sizes, the only difference is that the subsets obtained with OAS have become larger.

The above results also apply to the datasets of 10,000 points. The only difference is that for 10,000 points, it was not possible for us to fit a RBF-model. A comparison with a model fitted to the complete dataset could therefore no longer be made. All the results can be found in Table 5 in Appendix 3.

4.6 Results for HSCT dataset of 2,487 points

To make a comparison with the results of Srivastava et al. (2004) on the HSCT dataset, we have to use some different performance measures. In Srivastava et al. (2004), the root mean squared percentage error (RMSPE), the Average Percentage Error and the Maximum Percentage Error are used to measure the accuracy. We therefore report these accuracy measures for the different subsets instead of the Maximum Error and the RMSE.

Another difference is in the subset sizes as Srivastava et al. (2004) find subsets of 126, 283 and 372 points. To make a fair comparison, we compare our subsets of 250 points with the subset of 283 points and the subsets of 350 points with the subset of 372 points. The subsets of 500 points are thus only used for the comparison between the different methods in this paper.

When comparing the results of Srivastava with our results, one of the surprising results is the difference in subset sizes when using the OAS method. Srivastava reports that using random orthogonal arrays of 250 and 686 points results in subsets of 126 and 283 points respectively. When we, however, use orthogonal arrays of these sizes, we obtain subsets of approximately 225 and 500 points. We tried to determine the cause of this large difference, but were not able to find a satisfying explanation. Use of different orthogonal arrays could be one explanation, but it seems unlikely that this alone causes such a large difference.

Looking at the other performance measures for the subsets of 250 and 283 points, we see that the RM-SPE and the Average Percentage Error obtained by Srivastava are equal to the results obtained with RAND. This is surprising as RAND is a very naive method. Furthermore, SS1, SS10 and SS1040 perform better on these two performance measures. For the Maximum Percentage Error, besides these methods also MAXMIN and OAS give better results. Although OAS is the same method as applied by Srivastava, these results are difficult to compare because of the difference in the resulting subset sizes.

For the subsets of 350 and 372 points, Srivastava obtained the best results for the RMSPE and Average Percentage Error. Of the other methods, SS1 and SS1040 also perform well on these measures. The Maximum Percentage Error is lowest for the SS1 method, followed by Srivastava's results and SS1040. Notice, that SS10 performs quite bad on the Maximum Percentage Error and in a lesser extend on the RMSPE. This could be a sign that indeed a cluster of points is selected which negatively affects the overall accuracy.

As the other performance measures are not calculated by Srivastava, we can only compare them over the different methods we implemented ourselves. For the condition number and average robustness, SS1, SS1040, MAXMIN, DELETION and OAS give good results. The high values for the condition number and the average robustness measure for the SS10 method are again a sign of possible clustering.

The subset selection times are again lowest for RAND and OAS, followed by MAXMIN and DELETION. The time needed to fit the model is lowest for SS1, SS10 and SS1040. This is the result of using the optimal parameters settings of previous iterations of the algorithm. If we would not have used this information, but instead used a standard starting vector, the time to fit the model would be comparable to those of the RAND method. All in all, the results are similar to our results for the three artificial datasets. Although we used some different performance measures for this datasets, the ranking of the different methods for most aspects is the same. This increases our confidence that results obtained by the artificial datasets can also be obtained for real life datasets.

5 Conclusions and further research

In this paper, we show that fitting a Kriging model to a smaller but more uniform dataset can result in better Kriging models. Especially for large non-uniform datasets using a uniform subset instead of the complete

dataset can have several advantages. Reducing the time necessary to fit the model, avoiding numerical inaccuracies and improving the robustness with respect to errors in the output data are some of the aspects which can be improved by using a more uniform subset.

To select a uniform subset several new and current methods are described. All these methods are tested on artificial subsets of three different sizes and two levels of uniformity. Furthermore, tests are performed on the HSCT dataset, which was also used by Srivastava et al. (2004). The tests show that by using uniform subsets, we can indeed find accurate Kriging models in less time. Furthermore, these Kriging models are more robust and less susceptible to numerical inaccuracies. When comparing the different methods for finding subsets, there is no overall winner. SS1040 generally performs well on accuracy, robustness and numerical accuracy. Subsets obtained with SS1040 even results in Kriging models which are more accurate than RBF-models fitted on the complete dataset. Compared to the other methods, SS1040 is, however, relatively time consuming, but still considerably faster than fitting a Kriging model to the complete dataset. The OAS method is much faster, but has a lower accuracy, robustness and numerical accuracy. Deciding which method is best for a practical application, thus depends on how the different aspects are valued. The comparison made in this paper can be used to facilitate the user in making a good choice.

Further research on this problem could be targeted at finding new methods for selecting a subset. These new methods could for instance try to find even better subsets by using a different objective to obtain a uniform subset or by taking into account other properties of the dataset. Another option would be to develop methods which dynamically determine a suitable subset size. Furthermore, it would be interesting to determine whether the results found in this paper also apply to other modeling methods. There could be more modeling methods which benefit from using a small uniform dataset instead of a large non-uniform one.

Acknowledgements The author wishes to thank Edwin van Dam and Dick den Hertog for the many inspiring conversations on the topic of this paper, and Peter Stehouwer and Erwin Stinstra for some useful ideas. The author also wishes to thank Kemper Lewis, Christina L. Bloebaum and Timothy W. Simpson for providing the HSCT dataset. Furthermore, the author wishes to thank the anonymous referees and the associate editor for their valuable comments on a previous version of this paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Table 3 Average, minimum, and maximum of all performance measures over 50 artificial datasets of 2,000 points

Springer

Table 3 (continued)

Uniform datasets				Non-uniform datasets															
Average over datasets				Maximum over datasets				Minimum over datasets				Maximum over datasets				Minimum over datasets			
Time subset selection																			
Subset size		250	350	500	250	350	500	250	350	500	250	350	500	250	350	500			
RAND		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			
SS1		9.56	24.15		10.18	24.91		9.39	23.67		9.52	24.05		10.12	24.89				
SS10		0.92	2.33	6.82	0.95	2.38	7.38	0.90	2.28	6.69	0.92	2.32	6.81	0.96	2.39	7.40			
SS1040		0.92	2.32	6.79	0.95	2.77	7.24	0.86	2.26	6.64	0.92	2.32	6.80	0.95	2.37	7.39			
MAXMIN		0.02	0.02	0.02	0.02	0.02	0.03	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.01	0.01			
DELETION		0.07	0.07	0.06	0.08	0.07	0.07	0.06	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.06			
FEX		0.21	0.28	0.33	0.37	0.47	0.70	0.08	0.09	0.11	0.24	0.30	0.39	0.35	0.42	0.57			
OAS mean		0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01			
OAS min		0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01			
OAS max		0.00		0.01	0.01		0.01	0.00		0.01	0.00		0.01	0.00		0.01			
Time model fitting																			
Kriging		15.09		15.81		14.87		15.02		15.40		14.87		15.40		14.87			
RBF		0.25		0.26		0.25		0.25		0.25		0.25		0.26		0.25			
Subset size		250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	500		
RAND		0.10	0.21	0.51	0.11	0.23	0.53	0.09	0.20	0.48	0.10	0.21	0.51	0.11	0.23	0.53	0.48		
SS1		0.08	0.16		0.09	0.17		0.07	0.14		0.08	0.16		0.15	0.17		0.15		
SS10		0.08	0.16	0.40	0.09	0.19	0.44	0.07	0.14	0.38	0.08	0.16	0.40	0.10	0.19	0.44	0.37		
SS1040		0.08	0.16	0.40	0.09	0.17	0.44	0.07	0.15	0.37	0.08	0.16	0.41	0.10	0.17	0.71	0.36		
MAXMIN		0.05	0.10	0.22	0.05	0.11	0.23	0.05	0.09	0.21	0.05	0.10	0.22	0.05	0.11	0.24	0.21		
DELETION		0.05	0.10	0.22	0.05	0.10	0.23	0.05	0.09	0.21	0.05	0.10	0.22	0.05	0.11	0.23	0.21		
FEX		0.10	0.21	0.51	0.11	0.25	0.53	0.10	0.21	0.48	0.10	0.21	0.51	0.11	0.23	0.54	0.48		
OAS mean		0.08		0.62	0.08		0.70	0.08		0.59	0.08		0.59	0.08		0.62	0.57		
OAS min		0.07		0.50	0.08		0.60	0.06		0.47	0.07		0.47	0.07		0.53	0.42		
OAS max		0.08		0.71	0.09		0.85	0.08		0.66	0.08		0.68	0.10		0.72	0.64		
Condition number																			
Kriging		237,859		935,201		66,964		420,661		1,409,590		95,175		95,175		2,384			
Subset size		250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	500		
RAND		1,826	5,367	8,592	19,835	47,339	47,927	355	495	651	1,986	4,194	11,183	9,879	17,801	38,230	2,384		
SS1		288	579		679	1,024		137	333		349	811		983	2,539	188	403		
SS10		625	1,119	1,999	1,869	3,292	4,792	259	387	778	748	1,461	3,012	2,596	8,534	16,395	1,390		
SS1040		266	493	1,100	1,361	1,014	1,956	157	314	728	286	633	1,645	667	1,275	3,744	977		
MAXMIN		150	355	904	731	2,062	3,750	31	102	188	145	332	925	475	916	3,062	237		
DELETION		187	430	1,167	457	1,418	4,019	42	101	172	235	405	998	656	1,023	3,514	214		
FEX		919	2,267	4,898	2,390	12,665	14,522	280	242	1,199	922	2,634	5,159	3,717	8,739	27,249	892		
OAS mean		473	3,991	3,991	1,090		6,902	273		2,524	505		4,222	958		9,199	2,072		

OAS min	148	1,351	271	2,531	74	514	172	1,419	299	2,526	79	586
OAS max	1,232	9,668	4,787	28,866	434	4,320	1,275	9,628	3,344	28,256	342	3,972
Maximum robustness												
Kriging												
Subset size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	7.27	10.12	12.60	21.34	29.26	56.40	3.53	4.03	4.01	24.40	26.62	42.52
SS1	1.75	1.95		2.33	2.54		1.41	1.50		2.76	3.44	
SS10	2.38	2.61	2.74	4.52	4.77	4.63	1.53	1.95	2.02	4.15	10.50	11.57
SS1040	1.72	1.93	2.20	2.49	2.67	2.86	1.38	1.65	1.68	2.76	3.29	3.52
MAXMIN	1.35	1.45	1.64	1.78	1.89	2.22	1.10	1.21	1.30	1.66	1.83	2.24
DELETION	1.43	1.56	1.76	1.84	2.00	2.34	1.12	1.25	1.37	1.96	1.98	2.28
FEX	5.25	6.47	8.59	13.72	13.19	16.36	2.51	3.24	4.53	14.61	30.67	24.34
OAS mean	2.18		3.13	2.59		3.68	1.92		2.64	2.85	3.98	1.91
OAS min	1.64		2.33	2.02		2.87	1.41		1.82	2.06	2.91	1.43
OAS max	2.96		4.37	4.01		7.11	2.23		2.98	5.38	8.11	2.25
Average robustness												
Kriging												
Subset size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1.11	1.57	1.94	2.29	5.01	3.97	0.60	0.73	0.78	1.19	1.55	2.19
SS1	0.67	0.76		0.74	0.81		0.61	0.70		0.69	0.78	
SS10	0.74	0.81	0.93	0.82	0.89	1.00	0.64	0.73	0.87	0.76	0.85	0.97
SS1040	0.71	0.79	0.90	0.84	0.86	0.94	0.64	0.71	0.83	0.73	0.80	0.92
MAXMIN	0.82	0.89	1.01	1.04	1.11	1.25	0.66	0.76	0.81	0.89	1.00	1.03
DELETION	0.84	0.93	1.05	1.00	1.12	1.28	0.64	0.73	0.78	0.87	0.92	1.03
FEX	0.98	1.28	1.70	1.47	2.94	2.53	0.60	0.56	1.00	0.98	1.39	1.65
OAS mean	0.94		1.27	1.05		1.38	0.88		1.17	1.03	1.27	1.44
OAS min	0.79		1.08	0.88		1.21	0.67		0.93	0.87	1.08	1.22
OAS max	1.12		1.48	1.49		1.78	0.98		1.31	1.11	1.47	1.69
Real subset size												
Subset size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	250	350	500	250	350	500	250	350	500	250	350	500
SS1	250	350	500	250	350	500	250	350	500	250	350	500
SS10	250	350	500	250	350	500	250	350	500	250	350	500
SS1040	250	350	500	250	350	500	250	350	500	250	350	500
MAXMIN	250	350	500	250	350	500	250	350	500	250	350	500
DELETION	250	350	500	250	350	500	250	350	500	250	350	500
FEX	250	350	500	250	350	500	250	350	500	250	350	500
OAS mean	237		578	239		583	235		571	236	238	572
OAS min	227		533	233		552	219		517	225	231	542
OAS max	244		610	248		624	241		596	243	248	616

Appendix 2: Results for artificial datasets of 5,000 points

Table 4 Average, minimum, and maximum of all performance measures over 50 artificial datasets of 5,000 points

RMSE	Subset size	Uniform datasets				Non-uniform datasets			
		Average over datasets		Maximum over datasets		Minimum over datasets		Maximum over datasets	
RBF	Subst size	250	350	500	250	350	500	250	350
		0.17	0.14	0.12	0.19	0.17	0.15	0.14	0.12
		0.10	0.08	0.07	0.11	0.08	0.07	0.10	0.07
		0.11	0.09	0.07	0.14	0.09	0.07	0.14	0.07
		0.10	0.08	0.07	0.19	0.09	0.07	0.19	0.07
		0.17	0.14	0.11	0.23	0.19	0.14	0.21	0.17
		0.17	0.14	0.12	0.19	0.17	0.15	0.14	0.12
		0.17	0.15	0.12	0.20	0.18	0.15	0.23	0.18
		0.17	0.15	0.12	0.20	0.18	0.15	0.23	0.18
		0.17	0.15	0.12	0.20	0.18	0.15	0.23	0.18
Maximum error	Subst size	250	350	500	250	350	500	250	350
		0.17	0.14	0.12	0.19	0.17	0.15	0.14	0.12
		0.10	0.08	0.07	0.11	0.08	0.07	0.10	0.07
		0.11	0.09	0.07	0.14	0.09	0.07	0.14	0.07
		0.10	0.08	0.07	0.19	0.09	0.07	0.19	0.07
		0.17	0.14	0.11	0.23	0.19	0.14	0.21	0.17
		0.17	0.14	0.12	0.19	0.17	0.15	0.14	0.12
		0.17	0.15	0.12	0.20	0.18	0.15	0.23	0.18
		0.17	0.15	0.12	0.20	0.18	0.15	0.23	0.18
		0.17	0.15	0.12	0.20	0.18	0.15	0.23	0.18
RBF	Subst size	250	350	500	250	350	500	250	350
		1.59	1.45	1.33	1.96	1.90	1.79	1.63	1.48
		0.76	0.74	0.69	1.04	1.09	0.97	0.81	0.77
		0.77	0.72	0.69	1.02	1.01	0.97	0.80	0.76
		0.74	0.70	0.68	1.31	0.97	0.90	0.78	0.75
		1.58	1.39	1.15	2.29	2.08	1.39	1.47	1.28
		1.56	1.41	1.17	2.21	2.19	1.84	1.59	1.40
		1.61	1.50	1.29	1.95	2.01	1.76	1.59	1.47
		1.59	1.15	1.15	1.75	1.75	1.36	1.58	1.58
		1.30	0.88	0.88	1.51	1.51	1.07	0.68	1.25
RBF	Subst size	250	350	500	250	350	500	250	350
		1.59	1.45	1.33	1.96	1.90	1.79	1.63	1.48
		0.76	0.74	0.69	1.04	1.09	0.97	0.81	0.77
		0.77	0.72	0.69	1.02	1.01	0.97	0.80	0.76
		0.74	0.70	0.68	1.31	0.97	0.90	0.78	0.75
		1.58	1.39	1.15	2.29	2.08	1.39	1.47	1.28
		1.56	1.41	1.17	2.21	2.19	1.84	1.59	1.40
		1.61	1.50	1.29	1.95	2.01	1.76	1.59	1.47
		1.59	1.15	1.15	1.75	1.75	1.36	1.58	1.58
		1.30	0.88	0.88	1.51	1.51	1.07	0.68	1.25

 Springer

Table 4 (continued)

Uniform datasets												
Non-uniform datasets												
Average over datasets												
Maximum over datasets												
Minimum over datasets												
 												

Appendix 3: Results for artificial datasets of 10,000 points

Table 5 Average, minimum, and maximum of all performance measures over 50 artificial datasets of 10,000 points

RMSE	Subset size	Uniform datasets						Non-uniform datasets					
		Average over datasets			Maximum over datasets			Average over datasets			Maximum over datasets		
		250	350	500	250	350	500	250	350	500	250	350	500
Maximum error	RAND	0.17	0.15	0.13	0.21	0.18	0.15	0.17	0.15	0.13	0.22	0.20	0.16
	SS1	0.10	0.08	0.07	0.11	0.08	0.09	0.10	0.08	0.07	0.21	0.09	0.10
	SS10	0.12	0.09	0.07	0.15	0.10	0.08	0.12	0.09	0.07	0.13	0.10	0.09
	SS1040	0.10	0.08	0.06	0.21	0.10	0.07	0.10	0.08	0.06	0.11	0.09	0.07
	MAXMIN	0.17	0.14	0.11	0.20	0.17	0.15	0.17	0.14	0.11	0.21	0.18	0.14
	DELETION	0.17	0.14	0.12	0.20	0.17	0.15	0.17	0.15	0.12	0.20	0.17	0.14
	FEX	0.17	0.14	0.13	0.20	0.16	0.14	0.17	0.15	0.13	0.22	0.19	0.15
	OAS mean	0.17	0.14	0.10	0.18	0.16	0.11	0.17	0.15	0.10	0.18	0.16	0.11
	OAS min	0.15		0.09	0.17		0.10	0.15		0.10	0.16		0.10
	OAS max	0.19	0.12	0.12	0.21	0.21	0.13	0.19	0.12	0.12	0.21	0.13	0.17
Maximum error	Subset size	250	350	500	250	350	500	250	350	500	250	350	500
	RAND	1.56	1.48	1.35	2.12	2.01	1.95	1.55	1.52	1.32	1.92	2.17	1.78
	SS1	0.59	0.54		0.90	0.88		0.69	0.62		1.46	0.84	
	SS10	0.64	0.56	0.50	0.87	0.84	0.73	0.69	0.62	0.58	0.97	0.98	0.94
	SS1040	0.63	0.54	0.50	1.48	0.78	0.71	0.65	0.62	0.57	0.99	0.99	0.85
	MAXMIN	1.61	1.39	1.11	2.37	1.92	1.79	1.58	1.34	1.11	1.95	1.84	1.50
	DELETION	1.59	1.42	1.23	2.04	1.84	1.60	1.58	1.39	1.10	2.03	1.99	1.47
	FEX	1.60	1.40	1.32	1.99	1.87	1.86	1.64	1.45	1.38	1.92	1.86	1.74
	OAS mean	1.60		1.11	1.75		1.25	1.58		1.13	1.75		1.24
	OAS min	1.29		0.85	1.54		1.08	1.27		0.85	1.48		1.10
	OAS max	1.93	1.42	1.42	2.33	2.33	1.70	1.89	1.89	1.42	2.15	1.63	1.72

Table 5 (continued)

Time subset selection	Uniform datasets						Non-uniform datasets					
	Average over datasets			Maximum over datasets			Average over datasets			Maximum over datasets		
	250	350	500	250	350	500	250	350	500	250	350	500
Subst size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SS1	24.17	52.39	11.67	27.94	55.15	12.33	24.21	52.33	11.69	25.68	55.15	11.24
SS10	2.39	5.17	11.57	2.74	5.51	12.33	2.38	5.16	11.69	2.78	5.76	4.89
SS1040	2.36	5.10	11.57	2.73	5.53	12.01	2.37	5.15	11.60	2.73	5.66	4.82
MAXMIN	0.78	0.79	0.82	0.87	0.89	0.92	0.79	0.81	0.84	1.16	1.18	0.70
DELETION	3.55	3.53	3.49	3.68	3.65	3.60	3.62	3.59	3.55	4.02	3.99	3.20
FEX	0.83	0.95	1.18	1.16	1.40	1.92	1.10	1.41	1.93	1.36	1.77	1.61
OAS mean	0.02	0.05	0.05	0.02	0.05	0.05	0.02	0.05	0.05	0.02	0.05	0.05
OAS min	0.02	0.05	0.05	0.02	0.05	0.05	0.02	0.05	0.05	0.02	0.05	0.04
OAS max	0.02	0.05	0.05	0.03	0.07	0.07	0.02	0.05	0.05	0.03	0.06	0.05
Time model fitting												
Subst size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.10	0.20	0.50	0.10	0.22	0.52	0.10	0.20	0.50	0.11	0.22	0.52
SS1	0.08	0.17	0.40	0.09	0.18	0.46	0.08	0.17	0.40	0.09	0.18	0.36
SS10	0.08	0.17	0.41	0.09	0.19	0.45	0.08	0.17	0.41	0.09	0.19	0.37
SS1040	0.08	0.17	0.41	0.09	0.18	0.45	0.08	0.17	0.41	0.09	0.19	0.37
MAXMIN	0.10	0.21	0.50	0.11	0.22	0.52	0.10	0.21	0.51	0.11	0.22	0.48
DELETION	0.10	0.21	0.50	0.11	0.23	0.53	0.10	0.21	0.50	0.11	0.23	0.47
FEX	0.10	0.21	0.50	0.10	0.26	0.55	0.10	0.21	0.50	0.11	0.22	0.48
OAS mean	0.09	0.17	0.48	0.10	0.26	1.00	0.09	0.21	0.96	0.10	0.22	0.94
OAS min	0.09	0.17	0.48	0.10	0.26	1.00	0.09	0.21	0.96	0.10	0.22	0.94
OAS max	0.10	0.21	0.50	0.11	0.26	1.09	0.10	0.21	0.99	0.10	0.22	0.84
Condition number												
Subst size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1,553	2,824	6,239	6,462	12,486	31,343	2,083	5,611	11,945	8,990	61,456	52,456
SS1	217	432	762	612	925	2,031	341	590	1,194	4,518	4,283	2,194
SS10	1,153	1,546	2,509	4,013	3,682	9,362	1,605	2,139	3,153	6,111	9,442	11,228
SS1040	214	463	762	568	5,698	6,318	212	423	849	314	798	1,883
MAXMIN	109	221	685	302	855	2,031	121	273	607	283	940	2,142
DELETION	183	356	813	693	1,378	2,289	186	359	961	638	1,624	4,593
FEX	1,012	1,886	5,069	5,271	5,548	23,544	1,209	2,937	4,149	8,944	14,051	18,497
OAS mean	448	891	4,296	891	5,548	6,921	447	2,937	4,149	909	8,439	2,831
OAS min	141	232	1,435	232	2,653	2,653	145	145	1,581	322	2,517	599
OAS max	1,226	10,107	33,058	5,348	33,058	408	1,172	11,156	44,761	3,812	44,761	4,509

Maximum robustness												
Subset size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	7.39	9.09	11.91	22.77	22.48	24.26	3.66	4.25	4.12	9.88	14.76	19.66
SS1	1.64	1.77		2.54	2.56		1.31	1.50		1.70	1.93	
SS10	3.05	3.24	3.41	7.21	6.96	6.60	1.91	1.99	2.10	3.39	3.37	3.50
SS1040	1.67	1.79	1.90	2.28	3.43	3.50	1.33	1.50	1.61	1.67	1.85	1.99
MAXMIN	1.35	1.36	1.53	1.77	1.62	1.92	1.08	1.19	1.30	1.32	1.40	1.50
DELETION	1.41	1.53	1.68	1.83	1.80	1.97	1.12	1.27	1.36	1.45	1.53	1.71
FEX	5.23	5.65	8.26	17.93	10.25	19.53	2.69	3.58	3.49	5.31	7.48	8.71
OAS mean	2.15		3.19	2.75		3.79	1.80		2.77	2.14		3.25
OAS min	1.61		2.39	1.91		2.77	1.37		1.97	1.64		2.39
OAS max	3.14		4.53	6.02		8.86	2.01		3.42	3.06		4.72
Average robustness												
Subset size	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1.13	1.37	1.76	2.61	2.23	3.01	0.69	0.60	1.03	1.25	1.72	2.24
SS1	0.60	0.66		0.65	0.69		0.57	0.64		0.61	0.67	
SS10	0.76	0.81	0.88	1.01	0.94	1.01	0.67	0.72	0.82	0.78	0.83	0.91
SS1040	0.64	0.69	0.76	0.73	0.84	0.83	0.50	0.65	0.72	0.64	0.69	0.77
MAXMIN	0.80	0.86	0.96	0.92	1.00	1.10	0.68	0.77	0.84	0.80	0.88	0.95
DELETION	0.84	0.91	1.01	1.04	1.06	1.17	0.63	0.78	0.86	0.84	0.89	1.02
FEX	1.02	1.25	1.67	2.05	2.07	3.49	0.59	0.82	0.90	1.06	1.42	1.63
OAS mean	0.92		1.29	0.99		1.35	0.84		1.23	0.93		1.29
OAS min	0.78		1.11	0.87		1.23	0.67		1.00	0.79		1.11
OAS max	1.08		1.46	1.31		1.68	0.98		1.34	1.09		1.49
Average robustness												
Subset size	250	350	500	250	350	500	250	350	500	250	350	500
SS1	250	350		250	350		250	350		250	350	
SS10	250	350	500	250	350	500	250	350	500	250	350	500
SS1040	250	350	500	250	350	500	250	350	500	250	350	500
RAND	250	350	500	250	350	500	250	350	500	250	350	500
MAXMIN	250	350	500	250	350	500	250	350	500	250	350	500
DELETION	250	350	500	250	350	500	250	350	500	250	350	500
FEX	250	350		250	350		250	350		250	350	
OAS mean	248		666	249		669	247		663	247		660
OAS min	245		650	247		660	242		636	244		640
OAS max	250		678	250		683	249		670	250		675

Appendix 4: Results for HSCT dataset of 2,487 points

Table 6 Results for HSCT-dataset of 2,487 points

Results Srivastava			
Subset size	283	372	
RMSPE	1.00	0.24	
Avg. Perc. Err.	0.59	0.17	
Max. Perc. Err.	6.12	2.02	

RMSPE			
Subset size	250	350	500
RAND	1.00	0.93	0.86
OAS mean	1.30		1.11
OAS min	1.05		0.86
OAS max	1.50		1.38
FEX	1.72	1.73	1.73
MAXMIN	1.31	1.14	1.22
DELETION	1.70	1.83	1.46
SS1	0.66	0.65	
SS10	0.60	1.28	0.68
SS1040	0.58	0.44	0.36

Condition number			
Subset size	250	350	500
RAND	712,731,778	341,942,761	3,072,209,097
OAS mean	112,944,986		1,047,914,507
OAS min	2,213		3,344
OAS max	447,474,975		2,377,979,099
FEX	2,035,776,984	2,328,684,961	1,600,933,318
MAXMIN	68,976	69,945	55,685
DELETION	29,772	22,407	69,130
SS1	69,777	197,848	
SS10	1,568,232,546	7,678,597,938	9,457,605,211
SS1040	162,191	183,002	108,242

Average percentage error			
Subset size	250	350	500
RAND	0.59	0.52	0.52
OAS mean	0.85		0.69
OAS min	0.73		0.57
OAS max	0.95		0.81
FEX	1.19	1.17	1.21
MAXMIN	0.92	0.81	0.91
DELETION	1.22	1.33	1.08
SS1	0.53	0.51	
SS10	0.46	0.76	0.53
SS1040	0.44	0.35	0.27

Maximum percentage error			
Subset size	250	350	500
RAND	9.37	8.96	7.65
OAS mean	5.77		5.67
OAS min	4.75		4.97
OAS max	7.11		6.82
FEX	8.59	9.19	8.45
MAXMIN	5.60	5.44	5.32
DELETION	7.55	7.76	6.51
SS1	2.02	2.34	
SS10	2.37	13.73	2.13
SS1040	2.36	1.76	2.50

Time subset selection			
Subset size	250	350	500
RAND	0.00	0.00	0.00
OAS mean	0.01		0.04
OAS min	0.01		0.04
OAS max	0.01		0.05
FEX	7.29	7.81	7.97
MAXMIN	0.12	0.12	0.13
DELETION	0.47	0.45	0.42
SS1	21.30	57.15	
SS10	2.64	6.08	16.72
SS1040	2.31	5.74	17.12

Average robustness			
Subset size	250	350	500
RAND	43.08	15.11	27.80
OAS mean	19.06		63.38
OAS min	0.87		0.75
OAS max	62.81		117.07
FEX	177.70	182.17	142.40
MAXMIN	0.84	0.85	0.77
DELETION	1.61	1.66	1.06
SS1	1.50	1.35	
SS10	126.76	85.95	106.68
SS1040	1.17	1.03	0.91

Real subset size			
Subset size	250	350	500
RAND	250	350	500
OAS mean	226		505
OAS min	224		494
OAS max	230		515
FEX	250	350	500
MAXMIN	250	350	500
DELETION	250	350	500
SS1	250	350	
SS10	250	350	500
SS1040	250	350	500

Time model fitting			
Subset size	250	350	500
RAND	0.41	0.84	1.92
OAS mean	0.42		2.40
OAS min	0.23		2.23
OAS max	0.78		2.55
FEX	0.41	0.85	1.93
MAXMIN	0.38	0.56	1.21
DELETION	0.34	0.72	1.63
SS1	0.27	0.45	
SS10	0.23	0.43	0.97
SS1040	0.23	0.41	0.98

Appendix 5: Kriging model

As we focus on Kriging, we summarize some theory according to Sacks et al. (1989). In Kriging, the output data $y(x)$ is treated as a realization of a random function $Y(x)$. This random function is divided into a regression part and a stochastic part:

$$Y(x) = \sum_{j=0}^k \beta_j f_j(x) + Z(x)$$

where $k+1$ is the number of regression functions including $f_0(x) = 1$. In this paper for the regression part, we use the linear functions $f_j(x) = x_j$ for $j = 1, \dots, d$ where d is the number of dimensions. The stochastic part $Z(x)$ is assumed to have zero mean. Furthermore, the covariance between $Z(x)$ and $Z(w)$ is assumed to be of the form:

$$V(w, x) = \sigma^2 R(w, x)$$

where σ^2 is the constant process variance and $R(w, x)$ is the correlation between $Z(x)$ and $Z(w)$. To fit the Kriging model, we use a dataset with input data $X = [x^1, \dots, x^n]$ and corresponding output data $y_X = [y(x^1), \dots, y(x^n)]$. In Kriging, the vector y_X is assumed to be a realization of the stochastic vector $[Y(x^1), \dots, Y(x^n)]$.

To predict the output value at a new point x , Kriging uses the Best Linear Unbiased Predictor (BLUP). This means that $\hat{y}(x)$, the predicted output value at point x , is given by:

$$\hat{y}(x) = c^T(x) y_X$$

where the Kriging weights $c(x)$ are determined such that they minimize:

$$MSE(\hat{y}(x)) = E(c^T(x) Y_X - Y(x))^2 \quad (1)$$

under the unbiasedness constraint:

$$E(c^T(x) Y_X) = E(Y(x)) \quad (2)$$

Let us now introduce the following notation:

$$f(x) = [f_0(x), f_1(x), \dots, f_k(x)] = [1, x_1, \dots, x_d]^T$$

$$F = \begin{bmatrix} f^T(x^1) \\ \vdots \\ f^T(x^n) \end{bmatrix}$$

$$r(x) = [R(x, x^1), \dots, R(x, x^n)]^T$$

$$R = \begin{bmatrix} r^T(x^1) \\ \vdots \\ r^T(x^n) \end{bmatrix} = \begin{bmatrix} R(x^1, x^1) & R(x^1, x^2) & \dots & R(x^1, x^n) \\ R(x^2, x^1) & R(x^2, x^2) & \dots & R(x^2, x^n) \\ \vdots & \vdots & \ddots & \vdots \\ R(x^n, x^1) & R(x^n, x^2) & \dots & R(x^n, x^n) \end{bmatrix}$$

The matrix R is thus the correlation matrix containing the correlations between $Z(x^i)$ and $Z(x^j)$ for all $i, j \in \{1, \dots, n\}$. By definition of the spatial correlation function $R(\cdot)$, this matrix R is positive semi-definite and symmetric with ones on the diagonal.

Classical Kriging assumes that the Kriging weights $c(x)$ are independent of the output data. Therefore, we can rewrite the MSE in (1) as (Santner et al. 2003)

$$MSE(\hat{y}(x)) = \sigma^2 (1 + c^T(x) R c(x) - 2c^T(x) r(x)) \quad (3)$$

and the constraint in (2) as:

$$F^T c(x) = f(x)$$

Using Lagrange multipliers $\lambda(x)$, the MSE in (3) can be minimized by solving the following system of equations:

$$\begin{bmatrix} 0 & F^T \\ F & R \end{bmatrix} \begin{bmatrix} \lambda(x) \\ c(x) \end{bmatrix} = \begin{bmatrix} f(x) \\ r(x) \end{bmatrix}$$

Solving this system gives the following expressions for $\lambda(x)$ and $c(x)$:

$$\lambda(x) = (F^T R^{-1} F)^{-1} (F^T R^{-1} r(x) - f(x))$$

$$c(x) = R^{-1} (r(x) - F \lambda(x))$$

We see that determining the prediction of the output at point x requires solving a linear system containing the $n \times n$ matrix R . If the size of the dataset, n , becomes very large this can be quite time consuming and we can thus save time by using less data.

Thus far, we have not specified the form of the correlation function $R(x, w)$. For the correlation function, there are a number of alternatives. We choose to use the Gaussian correlation function:

$$R^\theta(w, x) = \prod_{j=1}^d \exp(-\theta_j |w_j - x_j|^2)$$

as this is the most frequently used correlation function for Kriging (Jin et al. 2001). Furthermore, we assume that $Z(x)$ is a Gaussian process.

We now still have to determine β , σ and θ such that the Kriging model interpolates through the training data. To do this, we use the maximum likelihood estimator (MLE). For β , this gives the generalized least-squares estimate:

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y_X.$$

The MLE of σ^2 is given by:

$$\sigma^2 = \frac{1}{n} (y_X - F\hat{\beta})^T R^{-1} (y_X - F\hat{\beta}).$$

To determine the MLE of θ , we must solve the following minimization problem (Sacks et al. 1989):

$$\min_{\theta} |R|^{1/n} \hat{\sigma}^2$$

Unfortunately, we do not have an analytic expression for the $\hat{\theta}$ that solves this problem. We thus need some numerical optimization procedure to determine $\hat{\theta}$. As mentioned in Section 4, we use the DACE toolbox of Lophaven et al. (2002). Notice that this minimization problem also contains R^{-1} as it is part of the expression for $\hat{\theta}$. The numerical optimization procedure has to determine R for multiple values of θ in order to find the solution to the minimization problem. This can become very time-consuming if R is very large. As R is a $n \times n$ matrix, the size of the training set n directly affects the time-consumption of this step.

Appendix 6: Radial basis functions

Besides Kriging models, we can also use radial basis function (RBF) models to construct a model. RBF-models have been developed by Hardy (1971) to interpolate through a training set of multi-dimensional data. To approximate the output value at a point x , the distances between this point and the training points are used. Each distance value is used as the input of a radially symmetric function. A linear combination of the output values of this function forms the approximation of the output value in x . The simple RBF-model used in this paper to approximate y is:

$$\hat{y} = \sum_i \beta_i ||x - x_i||$$

where $||x - x_i||$ denotes the Euclidean distance between the training point x_i and the approximation point x . When we fill in all training points (x_i, y_i) for $i = 1, \dots, n$, we obtain a set of linear equations. By solving this system, we can determine the coefficients β_i .

The advantage of RBF-models is that they have shown good fits to both stochastic and deterministic functions (Powell 1987) and that they require significantly less time to be fitted than Kriging models (Jin et al. 2001). Therefore, they are more suitable for larger datasets than Kriging. However, by combining subset selection and Kriging, we aim to make Kriging equally, or even better, suitable for large datasets. We therefore compare Kriging models fitted on a subset with radial basis functions fitted on the complete dataset. As some performance measures cannot be calculated for radial basis functions, we will only make a comparison of the accuracy and the time required to fit the model. For the Kriging models, this time also includes the time necessary to select the subset.

References

- Agca S, Eksioglu B, Ghosh JB (2000) Lagrangian solution of maximum dispersion problems. *Nav Res Logist* 47(2):97–114
- Banzhaf W, Francone FD, Keller RE, Nordin P (1998) Genetic programming: an introduction on the automatic evolution of computer programs and its applications. Morgan Kaufmann, San Francisco, CA, USA
- Booker AJ, Dennis JE, Frank, PD, Serafini, DB, Torczon V, Trosset MW (1999) A rigorous framework for optimization of expensive functions by surrogates. *Struct Multidisc Optim* 17(1):1–13
- Cherkassky V, Mulier F (1998) Learning from data: concepts, theory, and methods. Wiley, New York, NY, USA
- Davis GJ, Morris MD (1997) Six factors which affect the condition number of matrices associated with Kriging. *Math Geol* 29:669–683
- Dixon LCW, Szegö GP (1978) The global optimization problem: an introduction. In: Dixon LCW, Szegö GP (eds) *Toward global optimization*, vol 2. North-Holland, pp 1–15
- Erkut E (1990) The discrete p -dispersion problem. *Eur J Oper Res* 46:48–60
- Erkut E, Neuman S (1989) Analytical models for locating undesirable facilities. *Eur J Oper Res* 50:275–291
- Ghosh JB (1996) Computational aspects of the maximum diversity problem. *Oper Res Lett* 19:175–181
- Golbraikh A, Shen M, Xiao Z, Xiao YD, Lee KH, Tropsha A (2003) Rational selection of training and test sets for the development of validated QSAR models. *J Comput-Aided Mol Des* 17(2):241–253
- Golbraikh A, Tropsha A (2002) Predictive QSAR modeling based on diversity sampling of experimental datasets for the training and test set selection. *J Comput-Aided Mol Des* 16(5–6):357–369
- Golub GH, van Loan CF (1996) *Matrix computations*, 3rd edn. Johns Hopkins University Press, Baltimore
- Hansen P, Moon IJ (1994) Dispersing facilities on a network. *Cahiers du CERO* 36:221–234
- Hardy, RL (1971) Multiquadratic equations of topography and other irregular surfaces. *J Geophys Res* 76(8):1905–1915
- Hedayat AS, Sloane NAJ, Stufken J (1999) *Orthogonal arrays: theory and applications*. Springer, New York

- Husslage BGM, van Dam ER, den Hertog D, Stehouwer HP, Stinstra E (2003) Coordination of coupled black box simulations in the construction of metamodels. *Concurr Eng* 11(4):267–278
- Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodeling techniques under multiple modelling criteria. *Struct Multidisc Optim* 23:1–13
- Jin R, Chen W, Sudjianto A (2002) On sequential sampling for global metamodeling in engineering design. In: DETC-DAC34092, 2002 ASME design automation conference, pp 1–10
- Jin R, Chen W, Sudjianto A (2005) An efficient algorithm for constructing optimal design of computer experiments. *J Stat Plan Inference* 134(1):268–287
- Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *J Glob Optim* 21(4):345–383
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13(4):455–492
- Koehler JR, Owen AB (1996) Computer experiments. *Handb Stat* 13:261–308
- Kordon A (2006) Evolutionary computation at dow chemical. *SIGEVolution* 1(3):4–9
- Koza JR (1992) Genetic programming: on the programming of computers by natural selection. MIT Press, Cambridge, MA, USA
- Krige DG (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. *J Chem Metall Min Soc S Afr* 52(6):119–139
- Kuo CC, Glover F, Dhir KS (1993) Analyzing and modeling the maximum diversity problem by zero-one programming. *Decis Sci* 24:1171–1185
- Lam RLH, Welch WJ, Young SS (2002) Uniform coverage designs for molecule selection. *Technometrics* 44:99–109
- Lophaven SN, Nielsen HB, Sondergaard J (2002) DACE: a Matlab Kriging toolbox version 2.0. Technical Report IMM-TR-2002-12. Technical University of Denmark, Copenhagen
- Matheron G (1963) Principles of geostatistics. *Econ Geol* 58(8):1246–1266
- McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Owen AB (1992) Orthogonal arrays for computer experiments, integration and visualization. *Stat Sin* 2:439–452
- Padula SL, Alexandrov N, Green LL (1996) MDO test suite at NASA Langley research center. In: AIAA, NASA, and ISSMO, symposium on multidisciplinary analysis and optimization, vol 6. Bellevue, WA, pp 410–420
- Pisinger D (2006) Upper bounds and exact algorithms for p-dispersion problems. *Comput Oper Res* 33(5):1380–1398
- Powell MJD (1987) Radial basis functions for multivariable interpolation: a review. In: Clarendon press institute of mathematics and its applications conference series, pp 143–167
- Ravi SS, Rosenkrantz DJ, Tayi GK (1991) Facility dispersion problems: heuristics and special cases (extended abstract). In: Algorithms and data structures, 2nd workshop WADS '91, 14–16 August, Ottawa, Canada, pp 355–366
- Ravi SS, Rosenkrantz DJ, Tayi GK (1994) Heuristic and special case algorithms for dispersion problems. *Oper Res* 42:299–310
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4:409–435
- Santner ThJ, Williams BJ, Notz WI (2003) The design and analysis of computer experiments. Springer series in statistics. Springer, New York
- Siem, AYD, den Hertog D (2007) Kriging models that are robust with respect to simulation errors. Center Discussion Paper 2007-68. Tilburg University
- Simpson, TW, Peplinski J, Koch PN, Allen JK (2001) Metamodels for computer-based engineering design: survey and recommendations. *Eng Comput* 17:129–150
- Srivastava A, Hacker K, Lewis K, Simpson TW (2004) A method for using legacy data for metamodel-based design of large-scale systems. *Struct Multidisc Optim* 28:145–155
- Stehouwer HP, den Hertog D (1999) Simulation-based design optimization: methodology and applications. In: Proceedings of the first ASMO UK / ISSMO conference on engineering design optimization. Ilkley, UK
- Stein M (1987) Large sample properties of simulations using Latin hypercube sampling. *Technometrics* 29(2):143–151
- Steuer RE (1986) Multiple criteria optimization: theory and application. John Wiley, New York
- Tang B (1993) Orthogonal array-based Latin hypercubes. *J Am Stat Assoc* 88:1392–1397
- van Dam ER, Husslage BGM, den Hertog D, Melissen JBM (2007) Maximin Latin hypercube designs in two dimensions. *Oper Res* 55(1):158–169
- Wang G, Dong Z, Aitchison P (2001) Adaptive response surface method – a global optimization scheme for approximation-based design problems. *J Eng Optim* 33:707–734