A Bayesian Approach for Optimization Problems in Computer Experiments

by

Jhong-Yi Sie
Advisor
Ray-Bing Chen

Institute of Statistics,
National University of Kaohsiung
Kaohsiung, Taiwan 811 R.O.C.
June 2006
Contents

中文摘要 ii
英文摘要 iii

1 Introduction 1

2 Metropolized Matching Pursuit Algorithm 2

3 A Noise Basis-based Response Surface Methodology 4

4 Numerical Experiments 7

5 Some Comparisons with Design and Analysis of Computer Experiments 15
   5.1 DACE on grid .......................................................... 16
   5.2 Comparison .............................................................. 18

6 Conclusions 25

References 27
利用一個貝氏方法來解決電腦模擬實驗的極值問題

指導教授：陳瑞彬 博士
國立高雄大學統計學研究所

學生：謝鐘毅
國立高雄大學統計學研究所

摘要

本篇論文中，我們提出了一個迭代的演算法用來解決電腦模擬實驗 (computer experiments) 上的極值問題 (optimization problem)。在我們的演算法中的一個重點便是利用預先給定的基底 (basis) 的線性組合來逼近未知的真實反應曲面，並採用一個貝氏方法 (Bayesian approach) 來求得相對應的係數。在此我們利用一些模擬以及實際的實驗來呈現我們的新演算法的效能。最後比較我們的演算法及其他統計上求極值的方法。

關鍵字：Metropolized matching pursuit 演算法, 過度完全基底, 反應曲面, 代用法.
A Bayesian Approach for Optimization Problems in Computer Experiments

Advisor: Dr. Ray-Bing Chen
Institute of Statistics
National University of Kaohsiung

Student: Jhong-Yi Sie
Institute of Statistics
National University of Kaohsiung

ABSTRACT

In this thesis, we propose an iterative algorithm for solving the optimization problems in computer experiments. The key point of our algorithm is to approximate the true yet unknown response surface by a linear combination of predefined bases, and a Bayesian approach is applied here for inferring the corresponding coefficients. Several simulations and real experiments are displayed to show the performances of our new algorithm. Finally we compare our algorithm with other statistical optimization methods.

Keywords: Metropolized matching pursuit algorithm, overcomplete basis, response surface, surrogate approach.
1 Introduction

The optimization problem in computer experiment is considered here, and this optimization problem can be represented as

\[
\begin{align*}
\text{maximize} \quad & y = f(x), \\
\text{subject to} \quad & x \in \mathcal{X},
\end{align*}
\]  

where \( y \in \mathbb{R} \) is the response variable, \( f : \mathbb{R}^k \rightarrow \mathbb{R} \) is the response function, \( x = (x_1, \ldots, x_k) \) is the vector of \( k \) factors, and \( \mathcal{X} \subset \mathbb{R}^k \) is the compact experimental region. Typically, here the response function \( f \) is assumed to be unknown or very complicated. An intuitive idea for finding optimums is that given a grid of \( \mathcal{X} \) first, we compute the responses for all grid points, and then the optimum is found by maximizing these response values. However, the charge for computing all grid points is costly, especially when the experimentation is very expensive or time-consuming. There is another possible approach for solving this kind of optimization problem, which is based on the surrogate models. In this surrogate approach, the response function \( f \) is approximated by an inexpensive surrogate \( \hat{f} \), and then we optimize \( \hat{f} \) instead of \( f \).

One popular statistical optimization method with surrogate models is Response Surface Methodology (RSM), proposed by Box and Wilson (1951). In RSM, the surrogate models are the low-order polynomial models, and the central composite designs (CCD) are used to collecting the data for fitting these surrogate models. The first step of RSM is to use the first-order polynomial model for finding the direction of the maximum improvement in the response. If surface curvature exists, then the first-order polynomial model would be modified by adding higher-order terms into the model. At this time, usually the second-order polynomial model is employed to approximate the response to locate the optimum.

Besides RSM, Design and Analysis of Computer Experiment (DACE, Sacks et al., 1989b) is another useful statistical method for optimization by using surrogate approach. In DACE, the surrogate approximation is constructed by an interpolation scheme, kriging method (Matheron, 1963), and the space filling designs, for example, the Latin hypercube designs, are chosen for sampling experimental points. Here kriging method can be understood as linear prediction. Hence the kriging predictor for the unobserved response is a weighted linear combination of all the observed experiment points, and these weight coefficients are estimated by minimizing the Mean Square Error (MSE) of the predictor.

Wang and Chen (2004) and Chen et al. (2006) proposed the Basis-based Response Surface Method (BRSM), which is also an optimization algorithm by using surrogate
models. The key idea of BRSM is to consider response surfaces as images and then the surrogate surfaces are linear combinations of pre-defined bases, i.e. \( \hat{f} = \sum c_i \phi_i(x) \), where \( \phi_i \) is an element of the basis dictionary defined on the experimental region.

Due to the noiseless model of response, \( y = f(x) \), in BRSM, the coefficients of this surrogate surface, \( \hat{f} = \sum c_i \phi_i(x) \), are inferred by a practical image representation technique, matching pursuit algorithm (MP, Mallat and Zhang, 1993, and Bergeaud and Mallat, 1996). Basically, MP is to directly minimize \( \|y - \sum c_i \phi_i(x)\| \). In fact, we can consider the difference between \( f \) and the surrogate response \( \hat{f} \) as a noise \( \varepsilon \). Hence in this thesis, the response \( y \) is modelled as a noise model, \( y = \sum c_i \phi_i + \varepsilon \). Indeed, in image representation (Lewicki and Olshausen, 1999, Lewicki and Sejnowski, 2000 and Wu et al., 2003) and signal processing (Wolfe et al., 2004), the models of the images (or signals) also can be represented as noise model. Thus, the goal of this thesis is to modify BRSM for this noise model, and the key point of this modification is that we need to have a new method for inferring the unknown coefficients instead of MP. Here, based on the noise model, the Bayesian inference approach is considered, and a Markov Chain Monte Carlo algorithm, Metropolized matching pursuit algorithm (Wu et al., 2003), is adopted for inferring the coefficients.

This thesis is organized as follows. In Section 2, Metropolized matching pursuit algorithm is introduced. Then the modification of BRSM for the noise model is proposed in Section 3. In Section 4, the results of experiments for exploring the performances of our new algorithm are presented. In Section 5, our novel algorithm is compared with DACE numerically. Finally, a conclusion is given in Section 6.

## 2 Metropolized Matching Pursuit Algorithm

In image (or signal) representation, the model of an image \( I \) is usually represented as a linear superposition of basis dictionary, i.e. \( I = \sum c_i b_i \), where \( b_i \) is a basis in the dictionary and \( c_i \) is the corresponding basis coefficient. To infer the coefficients \( c_i \), several iterative algorithms have been proposed, and the matching pursuit algorithm is the most prominent one. Basically, the MP algorithm is an iterative algorithm to infer \( c_i \)'s by minimizing the \( \ell_2 \) norm of the residual of the image \( I \) and the weighted bases. However, the difference between the image \( I \) and the approximation \( \sum c_i b_i \) can be considered as a noise. Thus, the image \( I \) can also be modelled by a noise model. In fact, the noise models for images have been applied in image representation and signal processing. For detail, see Lewicki and Olshausen (1999), Lewicki and Sejnowski (2000), Wu et al. (2003), and
As mentioned above, the image $I$ is modelled as a noise model:

$$I = \sum_{(x,y,l,\theta,e)} c_{(x,y,l,\theta,e)} b_{(x,y,l,\theta,e)} + \varepsilon;$$  \hspace{2cm} (2)

where $\varepsilon$ is a vector of white noise with variance $\sigma^2$, the bases $b_{(x,y,l,\theta,e)}$ are selected from an overcomplete base dictionary, and $c_{(x,y,l,\theta,e)}$ is the corresponding base coefficient of $b_{(x,y,l,\theta,e)}$. For each base, $(x, y)$ is the central position of the base on the image domain, and $(l, \theta, e)$ is the type of the base. $l$ is the scale or the length of the base, $\theta$ is the orientation, and $e$ is the indicator for even/odd bases.

Based on this noise model (2), to infer the coefficients $c_i$’s can be accomplished by a Bayesian framework. Specifically, these coefficients $c_i$’s are inferred by sampling a posterior probability,

$$P(c_{(x,y,l,\theta,e)} | I) \propto P(I | c_{(x,y,l,\theta,e)}) P(c_{(x,y,l,\theta,e)}),$$  \hspace{2cm} (3)

where the prior of the coefficients $c_{(x,y,l,\theta,e)}$ is assumed to be independently distributed according to a distribution $p(c)$, which is the mixture of a point mass at 0 and a Gaussian distribution with a large variance, i.e.,

$$c_{(x,y,l,\theta,e)} \sim p(c) = \rho \delta_0 + (1 - \rho) N(0, \tau^2), \hspace{2cm} (4)$$

In this section, a Bayesian approach for inferring $c_i$’s, the Metropolized matching pursuit algorithm, is introduced. For simplicity, we use $i$ and $j$ to index $(x, y, l, \theta, e)$ and we define $z_i = 1$ if $b_j$ is active, i.e., $c_j \neq 0$, and $z_j = 0$ otherwise. Here we also assume that the parameters $(\rho, \tau^2, \sigma^2)$ are known for our model. Then the Metropolized matching pursuit algorithm iterate the following steps:

1. Randomly select a window $W$ on the image domain, let $A$ be the number of active bases within $W$, and let $B$ be the number of inactive bases within $W$. With probability $p_{\text{birth}}$ (a pre-designed number), go to [2]. With probability $p_{\text{death}} = 1 - p_{\text{birth}}$, go to [4].

2. For each inactive base $j = (x, y, l, \theta, e)$ with $(x, y) \in W$ and $z_j = 0$, compute the residual image, $R = I - \sum_{i \neq j} c_i b_i$ and

$$\gamma_j = \frac{p(I | z_j = 1; \{c_i, \forall i \neq j\}, I)}{p(I | z_j = 0; \{c_i, \forall i \neq j\}, I)} = \exp\left\{\frac{r_j^2}{2\sigma^2}\right\} \sqrt{\sigma^2 / \tau^2},$$  \hspace{2cm} (5)

where $r_j = \langle R, b_j \rangle / (1 + \sigma^2 / \tau^2)$, and $\sigma^2 = 1/(1/\sigma^2 + 1/\tau^2)$. Then with probability

$$p_{\text{accept}} = \frac{p_{\text{death}} \sum_{j: z_j = 0; (x,y) \in W} (1 - \rho) \gamma_j}{p_{\text{birth}} \times (A + 1)},$$

go to [3], with probability $1 - p_{\text{accept}}$ go back to [1].
3. Among all the inactive bases $j$ with $(x, y) \in W$ and $z_j = 0$, sample a base $j$ with probability proportional to $\gamma_j$, then let $z_j = 1$ and sample $c_j \sim N(r_j, \sigma^2)$. Go back to [1].

4. If $A > 0$, then randomly select an active base $b_k$ with $z_k = 1$ and $(x, y) \in W$. Then temporarily turn off $b_k$, i.e., set $c_k = 0$, and $A \rightarrow A - 1$ temporarily. Then for all the inactive bases $j$ with $z_j = 0$ and $(x, y) \in W$, including base $k$, do the same computation as [2] (as if $c_k = 0$), and compute $p_{\text{accept}}$ as in [2].

5. With probability $1/p_{\text{accept}}$, accept the proposal of deleting the base $k$, i.e., set $c_k = 0$. Go back to [1]. With probability $1 - 1/p_{\text{accept}}$, reject the proposal of deleting base $k$, i.e., recover the original $c_k$. Go back to [1].

Figure 1 shows the result of natural image with size $70 \times 70$ by Metropolized MP algorithm. In this example, the dictionary is composed of two-dimensional Gabor functions, and the details of Gabor bases will be described in section 4. Here the parameters $(\rho, \tau^2)$ of the prior are assumed to be $(0.5, 1)$, the variance $\sigma^2$ of the noise is assumed to be 1 and the pre-designed number $p_{\text{birth}}$ is assumed to be 0.7.

3 A Noise Basis-based Response Surface Methodology

Here, the noise model of response is considered, and a modification of the Basis-based Response Surface Methodology is proposed for the noise model. This novel algo-
Algorithm is called as Noise Basis-based Response Surface Methodology (NBRSM).

The response surface $y$ is modelled as

$$y = \sum_i c_i \phi_i + \varepsilon. \quad (6)$$

As mentioned in Section 2, the Metropolized matching pursuit algorithm is used here for inferring coefficients in the surrogate model. Just like BRSM, NBRSM is an iterative algorithm for numerical optimization, and the outline of NBRSM is shown in Figure 2.

1. Generate grid over the experimental region
2. Choose initial experiment points
3. Generate basis functions
4. Repeat until the optimal points are found
   4.1 Evaluate the response variables
   4.2 Construct the surrogate surface
   4.3 Predict possible optimal points and choose the next experiment points according to the constructed surrogate surface

Figure 2: Schema of the Noise Basis-based Response Surface Method.

The details of NBRSM are introduced as follows.

**Grid generation.** The first step of NBRSM is to choose a grid in the experimental region. Specifically, the $k$-dimensional experimental region $\mathcal{X}$ is discretized into a grid $\mathcal{P} = (p_1 \times p_2 \times \cdots \times p_N)$. These grid points are denoted as $x_1, \ldots, x_N$, where $N = p_1 p_2 \cdots p_k$. Therefore the response surface on the grid is defined as

$$\mathcal{S}_{\mathcal{P}} = \{(x_i, f(x_i)) | x_i \in \mathcal{P}\}, \quad (7)$$

and the discretized response surface can be treated as an (multi-dimension) image with finite number of pixels.

**Initial design.** Since the response surface is unknown, a set of grid points is chosen as the initial experimental points, $\mathcal{P}_{\text{init}} \subset \mathcal{P}$. Then we compute the corresponding response for the experimental points to construct the surrogate surface. Here, we choose a space filling designs, the uniform design (Fang, 1980, and Fang et al., 2000), to sample the initial experimental points. Note that we denote the set containing all the experiment points as $\mathcal{P}_{\text{exp}}$ and assume $\mathcal{P}_{\text{exp}} = \mathcal{P}_{\text{init}}$ initially.
Surrogate approximation. At the $t^{th}$ iteration, the response values $f(x)$’s are evaluated for all $x \in P_{\text{exp}}$. Thus, we only have the response values over the grid $P$ with respect to $P_{\text{exp}}$, and let $\mathcal{V}_{P_{\text{exp}}}$ be the corresponding response vector, i.e.,

$$
\mathcal{V}_{P_{\text{exp}}} = (f(x_1), \ldots, f(x_N))^T
$$

where

$$
f(x_i) = \begin{cases} 
f(x_i), & \text{if } x_i \in P_{\text{exp}}, \\
0, & \text{if } x_i \in P \backslash P_{\text{exp}}. 
\end{cases}
$$

As mentioned before, the response vector $\mathcal{V}_{P_{\text{exp}}}$ is treated as an image and is modelled as (2), i.e.,

$$
\mathcal{V}_{P_{\text{exp}}} = \sum_{j=1}^{M} c_{tj} \phi_j + \varepsilon,
$$

where $\varepsilon$ is a vector of white noises with variance $\sigma^2$, $\phi_j$’s are elements of basis dictionary over the whole grid $P$, and $c_{tj}$’s are the corresponding coefficients at the $t^{th}$ iteration. In addition, in this thesis, the basis dictionaries are considered as overcomplete dictionaries. That is, the number of the elements of the dictionary, $M$, is larger than the dimensionality of the grid $P$, i.e., $M > N$.

However, as mentioned in Chen et al. (2006), it may circumvent the possible pitfall that force the response variables in $P \backslash P_{\text{exp}}$ to be a constant. Thus, following the suggestion of Chen et al. (2006), we only collect the response values of $\mathcal{V}_{P_{\text{exp}}}$ that have been evaluated on the grid $P$, i.e.,

$$
\tilde{\mathcal{V}}_{P_{\text{exp}}} = (f(x_1), \ldots, f(x_p))^T.
$$

Let $e_{x_i}$ is the $N \times 1$ unit vector whose values are all zero expect the one corresponding to the point $x_i$ in which the value is assigned to be one, and $I_p$ be the $p \times N$ identification matrix that the $i$th row of $I_p$ is $(e_{x_i})^T$. Then $\tilde{\mathcal{V}}_{P_{\text{exp}}}$ can be represented as

$$
\tilde{\mathcal{V}}_{P_{\text{exp}}} = I_p \mathcal{V}_{P_{\text{exp}}},
$$

and from (8), we then have

$$
\tilde{\mathcal{V}}_{P_{\text{exp}}} = \sum_{j=1}^{M} c_{tj} \tilde{\phi}_j + \tilde{\varepsilon},
$$

$$
= \sum_{j=1}^{M} c_{tj} \frac{\tilde{\phi}_j}{\|\tilde{\phi}_j\|} \|\tilde{\phi}_j\| + \tilde{\varepsilon},
$$

$$
= \sum_{j=1}^{M} \tilde{c}_{tj} \frac{\tilde{\phi}_j}{\|\tilde{\phi}_j\|} + \tilde{\varepsilon},
$$

where $\tilde{c}_{tj}$ is the corresponding coefficient at the $t^{th}$ iteration.
where $\tilde{\varepsilon}$ is still a vector of white noises with variance $\sigma^2$, and $\tilde{\phi}_j = \mathcal{I}_p \phi_j$ is an element of basis dictionary corresponding to the experiment points. Here, the coefficients $\tilde{c}_{ij}$ of the current surrogate surface $\sum_{j=1}^{M} \tilde{c}_{ij} \tilde{\phi}_j / \| \tilde{\phi}_j \|$ can be inferred by using the Metropolized matching pursuit algorithm. After obtaining the coefficient $\tilde{c}_{ij}$, we construct the surrogate surface defined over the whole experiment region $P$ as

$$\sum_{j=1}^{M} \frac{\tilde{c}_{ij}}{\| \mathcal{I}_p \phi_j \|_2} \phi_j(x).$$ (13)

From the above discussion, the algorithm of NBRSM is summarized in Figure 3.

1. **Generate** a grid $\mathcal{P}$ containing $N$ points on the experimental region $\mathcal{X}$.
2. **Choose** $\mathcal{P}_{\text{init}}$ initial experiment points by a uniform design and define $\mathcal{P}_{\text{exp}} = \mathcal{P}_{\text{init}}$.
3. **Generate** a basis dictionary $\{\phi_j, j = 1, \ldots, M\}$.
4. **Repeat** until the optimal points are found.
   4.1. **Evaluate** $f(x)$'s for all $x \in \mathcal{P}_{\text{exp}}$.
   4.2. **Construct** $\tilde{\mathcal{V}}_{\mathcal{P}_{\text{exp}}}$ and $\mathcal{I}_p \phi_j$ accordingly.
   
   Estimate the coefficients $\tilde{c}_j$ by the “Metropolized matching pursuit algorithm”.
   
   Construct the surrogate surface $\sum_{j=1}^{M} \frac{\tilde{c}_j}{\| \mathcal{I}_p \phi_j \|_2} \phi_j(x)$.

4.3. **Find** new possible optimal point $x_{\text{new}}$ by minimizing the surrogate surface from $\mathcal{P} \setminus \mathcal{P}_{\text{exp}}$. Check if $f(x_{\text{new}})$ is optimal.

   **Update** $\mathcal{P}_{\text{exp}} = \mathcal{P}_{\text{exp}} \cup x_{\text{new}}$.

**Figure 3**: The Noise Basis-based Response Surface Method Algorithm.

### 4 Numerical Experiments

In this section, several numerical experiments for illustrating the performances of NBRSM on the cases of two factors, i.e. $k = 2$, are described. Before showing these numerical experiments, we introduce the basis dictionary which is used here.

In this thesis, our basis dictionary is an overcomplete dictionary, which is widely used in image representation, because the wider range of generating elements allows more flexibility in the image representation and may avoid some numerical difficulties while approximating an image by limited bases. For more details about the necessary of overcomplete dictionary, please see Donoho et al. (2004). For the overcomplete dictionary,
a popular dictionary is a set of Gabor functions. The two-dimensional Gabor functions are defined by

\[
g(u, v) = \frac{1}{Z} \exp \left[ -\frac{1}{2} (\sigma_u u^2 + \sigma_v v^2) \right] \cos \left[ \frac{2\pi u}{\lambda} \right] + \varphi,
\]

(14)

\[
u = u_0 + x_1 \cos \theta - x_2 \sin \theta
\]

(15)

\[
v = v_0 + x_1 \sin \theta - x_2 \cos \theta
\]

(16)

where \( Z \) is the normalizing constant, \((x_1, x_2)\) are coordinates of \( X \), \( u_0, v_0, \sigma_u \), and \( \sigma_v \) are user chosen parameters of a two-dimensional Gaussian window satisfying relations \( \sigma_v = \sqrt{2} \sigma_u \) and \( \lambda = \sqrt{2\pi} \sigma_u \), \( \lambda \) and \( \varphi \) are parameters of a sinusoidal grating, and \( \theta \) is the angle between the \( x_1 \)-axis of the image and the \( u \)-axis of the Gabor functions. In Figure 4, the surface of a 2D Gabor functions for \( \sigma_u = 1 \), \( \theta = 0 \) and \( \varphi = 0 \), and the 8 orientations of the Gabor dictionary for \( \sigma_u = 1 \) are shown.

To illustrate the performance of NBRSM, five experiments are demonstrated below, and the parameters of Gabor dictionary are: the angle \( \theta = \{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\} \), \( \sigma_u = \frac{10}{t\sqrt{2\pi}} \), for \( t = \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8\} \), and phase \( \varphi = 0 \). Thus, our Gabor dictionary contains \( M = 8 \times 7 \times N \) bases for \( N \) grid points. Meanwhile, the parameters in the Metropolized MP algorithm is set to be \((\phi, \tau^2, \sigma^2) = (0.5, 4, 0.04)\) and \( p_{\text{birth}} = 0.5 \).
Figure 5: The response surface and its corresponding contours of the objective function defined in problem $P_B$.

**Experiment 1.** We first consider a problem which is defined as

$$\text{Problem } P_B : \max f_B(x, y) = \frac{100}{10((x+1)^2-(y+1)^2)+x^2+4},$$

s.t. $-2.5 \leq x \leq 0.5$ and $-1.5 \leq y \leq 1.5$.  

(17)

This response surface is a banana-shaped function with unique maximum, and it has been studied in Balkin and Lin (2000) and Chen et al. (2006). The real response surface of $P_B$ and its corresponding contour are shown in (a) and (b) of Figure 5, respectively. To apply NBRSM to solve the problem, we choose the grid set $P_B = \{(x, y)|x \in \{-2.5, -\frac{14}{6}, -\frac{13}{6}, \ldots, 0.5\} \text{ and } y \in \{-1.5, -\frac{8}{6}, -\frac{7}{6}, \ldots, 1.5\}\}$. In other words, the grid contains 361 (19 × 19) points in the two-dimensional domain $[-2.5, 0.5] \times [-1.5, 1.5]$. Then the initial experimental points are chosen from the uniform design with 19 levels.

Figure 6 shows the evolution of NBRSM for solving problem $P_B$, and these four sub-figures illustrate the surrogate response surface that contains 19, 23, 27, and 35 experimental points, respectively. Note that, in the figures, the extreme point is marked by $\triangle$, the initial uniform design points are marked by $\bullet$, the chosen experimental points are marked by $\circ$, and the next experimental point determined by the corresponding surrogate surface is marked by $\times$.

From Figure 6, it seems that the trend of the true response surface is quickly detected by NBRSM. In fact, at the first iteration, the shape of the true response surface is soon sketched in rough as shown in part (a) and the "hot spot", or the area which contains the extreme point probably is also identified by NBRSM efficiently. Moreover, as shown in part (b)-(d), most of the new experimental points are chosen
in the hot spot area, and at the 17th iteration (35 experimental points), the extreme point of the response surface with 361 grid points is located by NBRSM.

**Experiment 2.** In this experiment, we considered a problem which is called Weber’s problem. Weber’s problem is to locate a central facility so that the total cost associated with distribution to several demand centers is minimized. For more details about Weber’s problem, please see (Francis et al., 1992). The cost function in this problem is

\[
    f_W(x, y) = \sum_i w_i \sqrt{(x - z_{1i})^2 + (y - z_{2i})^2} + \sin(0.0035(x^2 + y^2)) + 5 \sin(0.03((x + 20)^2 + y^2)).
\]

where \( w = (2, -4, 2, 1) \), \( z_1 = (-10, 0, 5, 25) \), and \( z_2 = (-10, 0, 8, 30) \). Here the Weber’s problem is transformed as the following optimization problem

\[
    \text{Problem } P_W : \quad \max f_W(x, y),
\]

s.t. \(-20 \leq x \leq 60\), and \(-20 \leq y \leq 60\). \quad (18)

The real response surface of \( f_W \) and its corresponding contour are shown in (a) and
Figure 7: The response surface and its corresponding contours of the objective function defined in problem $P_W$.

(b) of Figure 7, respectively. To apply NBRSM to solve the problem, we choose the grid set $P_W = \{(x, y) | x \in \{-20, \frac{50}{3}, \frac{40}{3}, \ldots, 60\} \text{ and } y \in \{-20, \frac{50}{3}, \frac{40}{3}, \ldots, 60\}\}$. In other words, the grid contains 625 ($25 \times 25$) points in the two-dimensional domain $[-20, 60] \times [-20, 60]$. Then the initial experimental points are chosen from the uniform design with 25 levels.

Figure 8 shows the evolution of NBRSM for solving problem $P_W$, and these four sub-figures illustrate the surrogate response surface that contains 25, 27, 30, and 33 experimental points, respectively. From Figure 8, NBRSM can also quickly detect the hot spot area which contains the maximum point, and finally finds the maximum point at 8 iterations (33 experimental points).

**Experiment 3.** Here we consider a problem which is defined as

\[
\text{Problem } P_{LI} : \quad \max f_{LI}(x, y) = (e^{-x^2-y^2} - 1) \\
\times \max \{(x + c_1)^2 + (y + c_2)^2, (x + d_1)^2 + (y + d_2)^2\},
\]

\[\text{s.t. } -8.4 \leq x \leq 7, \text{ and } -8.4 \leq y \leq 7, \quad (19)\]

where $(c_1, c_2) = -(d_1, d_2) = (30, 40)$. This problem has been presented in Kolda et al. (2003), and the true response surface $f_{LI}$ is locally Lipschitz and is strictly differentiable at its maximum point $(0, 0)$. The real response surface of $f_{LI}$ and its corresponding contour are shown in (a) and (b) of Figure 9, respectively. In this problem, the grid is set to be $P_{LI} = \{(x, y) | x \in \{-8.4, -7.7, -7, \ldots, 7\} \text{ and } y \in \{-8.4, -7.7, -7, \ldots, 7\}\}$. In other words, the grid contains 529 ($23 \times 23$) points in the two-dimensional domain $[-8.4, 7] \times [-8.4, 7]$. Then the initial experimental points are chosen from the uniform design with 23 levels.
Figure 8: Evolution of the surrogate response surfaces for solving Problem \( P_W \) by NBRSM. The four surrogate response surfaces containing 25, 27, 30, and 33 experimental points are shown in each of the sub-figures, respectively.

Figure 10 demonstrates the evolution of NBRSM for solving problem \( P_{LI} \), and these four sub-figures illustrate the surrogate response surface that contains 23, 24, 27, and 30 experimental points, respectively. Observing the figure, we see that NBRSM quickly arrive the vicinity of the peak and finds the maximum point at \((0,0)\) after 7 iterations. Also NBRSM captures the trend of the true response surface.

**Experiment 4.** In this case, we consider an artificial problem with two symmetric humps, which has been studied in Chen et al. (2006), and apply NBRSM to find the two local extreme points of the response surface. The problem is defined as follows.

Problem \( P_{TH} \): \[
\max f_{TH}(x, y) = \frac{-x^4 + 4.5x^2 + 2}{e^{2y^2}},
\]
\[
\text{s.t. } -2 \leq x \leq 2 \text{ and } -2 \leq y \leq 2.
\] (20)

As shown in Figure 11, there is a saddle point at \((0,0)\) and two maximum points at \((1.5,0)\) and \((-1.5,0)\) in the true response surface. The two maximum values of this are all equal to 113/16. To apply NBRSM to solve problem \( P_{TH} \), we choose the grid set \( \mathcal{P}_{TH} = \{(x, y) | x \in \{-2, -\frac{11}{6}, -\frac{10}{6}, \ldots, 2\} \text{ and } y \in \{-2, -\frac{11}{6}, -\frac{10}{6}, \ldots, 2\}\}. \)
Figure 9: The response surface and its corresponding contours of the objective function defined in problem $P_{LI}$.

That is, the grid contains 625 (25 × 25) points in the domain $[-2, 2] \times [-2, 2]$. Then the initial experimental points are chosen from the uniform design with 25 levels.

The evolution of NBRSM for solving problem $P_{TH}$ is shown in Figure 12, and the surrogate response surface that contains 25, 26, 47, and 52 experimental points is illustrated in part (a)-(d) of Figure 12, respectively. In part (a) of Figure 12, two hot spot areas are identified at the first iteration. Then, NBRSM successfully locates one maximum at $(1.5, 0)$ after 2 iterations as shown in part (b). In part (c)-(d), NBRSM keeps examining the area for a while and switches to the opposite area that contains another maximum point at $(-1.5, 0)$. Finally, the second maximum is located by NBRSM at 28th iteration (52 experimental points). Observing the searching process, we know that NBRSM not only finds both the optimums, NBRSM also detects the shape of the true response surface.

**Experiment 5.** In this case, we show the performance of NBRSM for solving an oscillatory response surface, and the problem is also adopted in Wang and Chen (2004) and Chen et al. (2006). Here in this problem, a dynamical system modelling an absorptive bistable laser diodes with an electronic-controlled external drive is considered, and in this dynamical system, the rate equations are

\[
\eta \frac{dN_{e1}}{dT} = S_{p1} + m_e \sin(2\pi \cdot m_f \cdot T) - \alpha_1 N_{e1}^2 - (\alpha_2 N_{e1}^2 + N_{e1} + \alpha_3)N_p - N_{e1}, \tag{21}
\]

\[
\eta \frac{dN_{e2}}{dT} = S_{p2} - \alpha_1 N_{e2}^2 - (\alpha_2 N_{e2}^2 + N_{e2} + \alpha_3)N_p - \alpha_4 N_{e2}, \tag{22}
\]

\[
\frac{dN_p}{dT} = N_p[\gamma_1(\alpha_2 N_{e1}^2 + N_{e1} + \alpha_3) + \gamma_2(\alpha_2 N_{e2}^2 + N_{e2} + \alpha_3)] - N_p
\]
Figure 10: Evolution of the surrogate response surfaces for solving Problem $P_{LI}$ by NBRSM. The four surrogate response surfaces containing 23, 24, 27, and 30 experimental points are shown in each sub-figures, respectively.

$$+\varepsilon(\gamma_1 N_{e1}^2 + \gamma_2 N_{e2}).$$

(23)

One essential indicator in characterizing a dynamical system is Lyapunov exponents. A positive Lyapunov exponent implies that the system is chaotic for the corresponding parameter setting. More details regarding the definition and computation of Lyapunov exponents can be found in Parker and Chua (1989).

Here we intend to find at least one set of the parameters $S_{p1}$ and $m_c$ in Eq. (21), such that the associated Lyapunov exponent is positive. However, there are two difficulties for finding suitable these parameters. First, it is extremely time consuming for computing Lyapunov exponents. Secondly, the relationship between the adjustable parameters and the resulting Lyapunov exponents are exceeding complicated. Therefore, an efficient numerical scheme for solving the problem is needed.

The problem for searching positive Lyapunov exponents is considered as an optimization problem, and as mentioned in Chen et al. (2006), the goal is the find at least one effective point such that the corresponding Lyapunov exponent is positive. Now the problem is shown as follows.

Problem $P_L : \max f_L(S_{p1}, m_c),$
where \( f_L(S_{p1}, m_c) \) is the value of Lyapunov exponent at \((S_{p1}, m_c)\). To apply NBRSM to solve problem \( P_L \), we first discretize the experimental region into the grid \( \mathcal{P}_L = \{(S_{p1}, m_c) | S_{p1} \in \{20, 20.5, 21, \ldots, 30\} \text{ and } m_c \in \{5, 5.5, 6, \ldots, 15\}\}, \) which contains 441 (\(21 \times 21\)) points in the domain \([20, 30] \times [5, 15]\). Then the initial experimental points are chosen from the uniform design with 21 levels. Figure 13 shows the true response surface and its projection over the \(21 \times 21\) grid of problem \( P_L \).

Figure 14 shows the surrogate response surfaces containing 21, 24, 59, and 101 experimental points. Observing the searching process, we see that NBRSM successfully found all 21 positive Lyapunov exponents. To be precise, expect the three positive Lyapunov are found by the initial points, NBRSM identifies the other 18 experimental points by using 24, 30, 33, 36, 38, 42, 43, 47, 49, 59, 60, 75, 76, 77, 97, 99, 100, and 101 experimental points. Besides, NBRSM also detects the trend of the true response surface, and keeps searching in the right quarter of the domain which contains most positive Lyapunov exponents possibly.

5 Some Comparisons with Design and Analysis of Computer Experiments

In this section, first we briefly introduce a popular statistical optimization method, Design and Analysis of Computer Experiments (DACE), and then we compare the performance between DACE and NBRSM by using above five problems.
Figure 12: Evolution of the surrogate response surfaces for solving Problem $P_{TH}$ by NBRSM. The four surrogate response surfaces containing 25, 26, 47, and 52 experimental points are shown in each sub-figures, respectively.

5.1 DACE on grid

In DACE, the response model is typically expressed as

$$y(x) = \beta^T F(x) + Z(x),$$

(25)

where $F(x)$ is a $q \times 1$ vector of the $q$ chosen regression functions, $\beta$ is a $q \times 1$ vector of the regression coefficients, and $Z(x)$ is a Gaussian stochastic process with zero mean and covariance function of the form $c(s, t) = \sigma^2 r_\theta(s, t)$. Note that $\sigma^2 > 0$ is unknown and $r_\theta$ is an correlation function, for example, Gauss correlation model, i.e.,

$$r_\theta(s, t) = \exp(-\theta \|s - t\|^2).$$

Further, in DACE, a space filling designs, for example, the Latin hypercube designs, is chosen for sampling a set of experimental points $\{x_i| i = 1, \cdots , p\}$. Let $Y_s = (y_1, \cdots , y_p)^T$ be the corresponding response vector, and $F_s = (F(x_1), \cdots , F(x_p))$, where $y_i$ is the response for $x_i$. Here we also define $R$ as the matrix of stochastic process correlations between the experimental points, i.e., $R = [r_\theta(x_{si}, x_{sj})]$, and define $r(x)$ as the vector of correlations between experimental points and an untried point $x$, i.e., $r(x) = [r_\theta(x_{si}, x_u)]$, for $n = 1, \cdots , k$. Then the surrogate response surfaces is constructed using the kriging
method. Here the unknown parameters \((\beta, \sigma^2, \theta)\) is estimated by the method of maximum likelihood, and the surrogate response surface is constructed by computing

\[
\hat{y}(x) = \hat{\beta}^T F(x) + r(x) R^{-1} (Y_s^T - F_s^T \hat{\beta}).
\]  

Hence, the kriging method can be considered as an exact interpolation. That is, the estimated values at experimental points are exactly equal to the response values.

Schonlau et al. (1998) noted that the general framework is proceeded as follows:

1. Choose a small initial experimental design spread over the entire input space. Run the computer code at these points.
2. Use all previous function evaluations to fit a statistical model for the objective function.
3. Based on the fitted model, find the “most promising” point in the input space for the next run.
4. Compute a stopping criterion. If it is met, then stop.
5. Run the computer code at the selected point in the input space. Go to Step 2.

According to above framework, an optimization algorithm on grid is proposed by applying DACE for surrogate surface construction, and this modified algorithm is called G-DACE. The first step of G-DACE is to discretize the experimental region as a grid. A uniform design is chosen for drawing the initial experimental points, and then the corresponding responses are evaluated. Based on these responses, DACE could be used for constructing the surrogate surfaces. Therefore, exclude the model assumption, the difference between Figure 13: The three-dimensional true response surfaces and two-dimensional projection over the \(21 \times 21\) grid of the objective functions defined in problem \(P_L\) are shown in each sub-figure, respectively.
Figure 14: Evolution of the surrogate response surfaces for solving Problem $P_L$ by NBRSM. The four surrogate response surfaces containing 21, 24, 59, and 101 experimental points are shown in each sub-figures, respectively.

G-DACE and NBRSM is the surrogate surface construction. The algorithm of G-DACE for optimization problem is shown in Figure 15.

1. **Generate** a grid $\mathcal{P}$ containing $N$ points on the experimental region $\mathcal{X}$.
2. **Choose** $N_{\text{init}}$ initial experiment points by a uniform design and define $\mathcal{P}_{\text{exp}} = \mathcal{P}_{\text{init}}$.
3. **Repeat** until the optimal points are found.
   3.1. **Evaluate** $f(x)$'s for all $x \in \mathcal{P}_{\text{exp}}$.
   3.2. **Construct** the surrogate surface $\hat{V}_{\mathcal{P}_{\text{exp}}}$ by the kriging method.
   3.3. **Find** new possible optimal point $x_{\text{new}}$ by minimizing the surrogate surface from $\mathcal{P} \setminus \mathcal{P}_{\text{exp}}$. Check if $f(x_{\text{new}})$ is optimal.
      
      **Update** $\mathcal{P}_{\text{exp}} = \mathcal{P}_{\text{exp}} \cup x_{\text{new}}$.

Figure 15: The G-DACE Algorithm.

5.2 Comparison

In this subsection, we would show the results of G-DACE for solving the five problems in Section 4, and then compare these results with those of NBRSM. Here in
G-DACE, the function of the kriging method is from a Matlab toolbox which is available at the web-site \url{http://www2.imm.dtu.dk/~hbn/dace}. For more details about this Matlab toolbox, please see Lophaven et al. (2002).

In our experiments, the regression function $F(\cdot)$ is chosen as the second-order polynomial model, and the correlation function $r(\cdot, \cdot)$ is Gaussian function. The results of these five problems are shown in the following:

**Problem $P_B$:** Figure 16 shows the evolution of DACE for solving problem $P_B$, and these four sub-figures illustrate the surrogate surface that contains 19, 20, 22 and 23 experimental points, respectively. In Figure 16, G-DACE finds the possible area which contains optimal point and searches around the area. After 5 iterations (23 experimental points), G-DACE successfully identifies the optimal point at $(0, 0)$. By comparing Figure 6 and 16, it seems that the trend of surface detects by NBRSM are is more similar to true surface than G-DACE does.

**Problem $P_W$:** The evolution of G-DACE for solving problem $P_W$ is shown in Figure 17, and the surrogate surface that contains 25, 54, 60, and 63 experimental points is shown in each sub-figure, respectively. From Figure 17, though G-DACE detects
the trend of the response surface roughly, G-DACE keeps searching in the left side of the surface. After 39 iterations, G-DACE finds the optimal point.

Problem $P_{LI}$: In Figure 18, the searching process of G-DACE for solving problem $P_{LI}$ shows that G-DACE detects the trend quickly, and uses only 26 experimental points to identify the optimal point at $(0, 0)$. The surrogate surface that contains 23, 24, 25 and 26 experimental points is illustrated in part (a)-(d) of Figure 18, respectively.

Problem $P_{TH}$: From Figure 19, we see that the trend of true response surface is detected by G-DACE at the first iteration, and then one optimal point at $(1.5, 0)$ is successfully identified after 2 iterations. Then, G-DACE keeps searching around the first optimal point, and G-DACE finally finds the second optimal point at $(-1.5, 0)$ after 11 iterations.

Problem $P_L$: Figure 20 shows the evolution of G-DACE for solving problem $P_L$, and these four sub-figures illustrate the surrogate surface that contains 21, 50, 150, and 439 experimental points, respectively. In Figure 20, G-DACE first searches in the right quarter of the surface, but after 30 iterations, G-DACE identifies 11 optimal points and starts to search in the median of the surface as shown in part (b) and
Figure 18: Evolution of the surrogate response surfaces for solving Problem \( P_{LI} \) by G-DACE. The four surrogate response surfaces containing 23, 24, 25, and 26 experimental points are shown in each sub-figures, respectively.

(c). In part (d), we see that almost all the grid points are evaluated, and then the last optimal point is just identified after 419 iterations. To be precise, besides three points are found by 21 initial points, G-DACE identified the other 18 positive Lyapunov exponents when 26, 28, 29, 32, 34, 40, 42, 47, 60, 61, 261, 340, 341, 355, 413, 436, 437, and 440 experimental points are used.

By observing above results, except the problem \( P_L \), it seems that G-DACE also have well performances for solving these problems. Here we compare the performances of NBRSM and G-DACE from these five problems. Since NBRSM is a Bayesian approach, we repeat 1000 times for each problem, and let \( S \) be a \( 1 \times 1000 \) vector of the number of experimental points for identifying the optimal point(s). Then from \( S \), we obtain the sample mean, sample variance, and sample standard deviation. Beside, we compute the 2.5\(^{th}\) and 97.5\(^{th}\) sample percentile for NBRSM. Here the \((100p)^{th}\) sample percentile of \( n \) samples is the \((np)^{th}\) order statistic. In addition, we define the efficient probability of G-DACE via NBRSM as

\[
p = P(X \leq c),
\]

where \( X \) is the number of experimental points for identifying the optimal point(s) by
Figure 19: Evolution of the surrogate response surfaces for solving Problem $P_{TH}$ by G-DACE. The four surrogate response surfaces containing 25, 26, 34, and 35 experimental points are shown in each sub-figures, respectively.

NBRSM, $c$ is the number of experimental points for identifying the optimal point(s) by G-DACE. Here, the efficient probability $p$ of G-DACE via NBRSM can be considered as that the NBRSM explores less points to identify the extremes than those used in G-DACE with probability $p$. The performances for these problems are summarized in the following tables.

In Table 1, besides the performances of NBRSM and G-DACE, the BRSM algorithm in Chen et al. (2006) is also applied here to solve these problems, but for problem $P_{LI}$, we add a step of basis selection into BRSM algorithm for getting the better performance. In NBRSM, since the number of explored experimental points for problem $P_{LI}$ is always 31, there is no sample variance in this problem. Hence, we denote the sample variance, standard deviation, and sample percentile for $P_{I}$ to be N/A. Moreover, for $P_{TH}$, there are two extreme points in the response surface. Thus we denote $P_{TH}$: 1 Max. to be the problem for identifying one optimal point, and $P_{TH}$: 2 Max. to be the problem for identifying two optimal points in problem $P_{TH}$. From Table 1, we see that all these three methods can efficiently find the optimal point for each problem. For NBRSM and BRSM, we find that the results of these two algorithms are similar. The reason might be that the difference of these two algorithms is only the method for inferring coefficients in the
Figure 20: Evolution of the surrogate response surfaces for solving Problem $P_L$ by G-DACE. The four surrogate response surfaces containing 21, 50, 150, and 439 experimental points are shown in each sub-figures, respectively.

<table>
<thead>
<tr>
<th>Problem</th>
<th>$P_B$</th>
<th>$P_W$</th>
<th>$P_{LI}$</th>
<th>$P_{TH}$: 1 Max.</th>
<th>$P_{TH}$: 2 Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid points</td>
<td>361</td>
<td>625</td>
<td>529</td>
<td>625</td>
<td>625</td>
</tr>
<tr>
<td>sample mean</td>
<td>30.0820</td>
<td>42.1820</td>
<td>31</td>
<td>26.158</td>
<td>52.561</td>
</tr>
<tr>
<td>NBRSN sample standard deviation</td>
<td>6.4078</td>
<td>7.3851</td>
<td>N/A</td>
<td>0.8437</td>
<td>4.3863</td>
</tr>
<tr>
<td>$2.5^{th}$ sample percentile</td>
<td>22</td>
<td>31</td>
<td>N/A</td>
<td>26</td>
<td>43</td>
</tr>
<tr>
<td>$97.5^{th}$ sample percentile</td>
<td>44</td>
<td>60</td>
<td>N/A</td>
<td>29</td>
<td>60</td>
</tr>
<tr>
<td>G-DACE explored point</td>
<td>24</td>
<td>64</td>
<td>27</td>
<td>27</td>
<td>36</td>
</tr>
<tr>
<td>efficient probability</td>
<td>0.2</td>
<td>0.982</td>
<td>N/A</td>
<td>0.96</td>
<td>0.006</td>
</tr>
<tr>
<td>BRSM explored point</td>
<td>30</td>
<td>39</td>
<td>29</td>
<td>29</td>
<td>46</td>
</tr>
<tr>
<td>efficient probability</td>
<td>0.64</td>
<td>0.344</td>
<td>N/A</td>
<td>0.98</td>
<td>0.068</td>
</tr>
</tbody>
</table>
surrogate models. Furthermore, according to efficient probabilities, except the problems $P_W$ and $P_{TH}$: 1 Max., it seems that G-DACE is quicker than NBRSM and BRSM to identify the optimal point. For example, in problem $P_B$, the efficient probability is equal to 0.2, and it means that G-DACE use less points to find the optimal point than those of NBRSM with probability 0.8. However, in the problem $P_W$, we see that the efficient probability is equal to 0.982, and it can be considered as that NBRSM use less points to identify the optimal point than G-DACE with probability 0.982. On the other hand, by observing the sample mean of NBRSM and the explored point of G-DACE in problem $P_B$, we find that the efficient probability of G-DACE is smaller than NBRSM, but the difference of explored point between NBRSM and G-DACE is equal to 6. It is because that in the 1000 repeated experiments, the results of explored points may contain 29 or 30 mostly. However, in problem $P_W$, the sample mean of NBRSM is only 42.1820, but the explored point of G-DACE is 64, and the difference is about 22. Thus, it seems that the performances of NBRSM is better than G-DACE. Here we show the response surface of $f_B$ and $f_W$ in Figure 21. From this figure, it seems that $f_W$ is an oscillatory surface, and $f_B$ is more smoother than $f_W$. From Table 1, we think that the reason for such kinds of results is caused by the surrogate construction and the smoothness of response surfaces. Since the kriging method is an exact interpolation, then the estimated values at experimental points are exactly equal to the response values. Thus, while the response surfaces are oscillatory or not smooth and the experimental points are fewer, the interpolation may be improper.

In the real problem $P_L$, the corresponding response surface is not smooth. In our experiments of G-DACE in Section 5, we find that G-DACE uses a lot of experimental points to search the median region and explores almost all grid points to find all the optimal points. Table 2 compares the performance of NBRSM and G-DACE for problem $P_L$. In this table, we see that the efficient probabilities of G-DACE via NBRSM are small,

![Figure 21: The response surfaces of $f_B$ and $f_W$.](image)
when we only need to identify few positive Lyapunov exponents. It means at this time, G-DACE explores less points than NBRSM with higher probability. However, in order to find more than 14 positive Lyapunov exponents, NBRSM seems have better performance than G-DACE. Thus, we believe that NBRSM is enable to deal the cases of unsmooth surface.

6 Conclusions

In this thesis, we propose a Bayesian algorithm, Noise Basis-based Response Surface Methodology (NBRSM), for optimization problems in computer experiments. Basically, NBRSM is the modification of Basis-based Response Surface Methodology (BRSM) for noise model. Like BRSM, the key point of NBRSM is to consider the response surface as an image, and to approximate the true yet unknown response surface by a linear combination of overcomplete Gabor bases. To infer the coefficients in the surrogate model, we apply a Bayesian approach, Metropolized matching pursuit algorithm. Then based on the is surrogate surface, we find the possible optimal points.

Five optimization problems are illustrated here for showing the performances of NBRSM. For the smooth surfaces, we see that NBRSM quickly sketches the trend of the unknown response surfaces, and efficiently identifies the hot spot areas, which may contains the possible optimal point. For the unsmooth surface, NBRSM also quickly finds the hot spots, and then NBRSM still successfully identify all extremes. For the real problem $P_L$, NBRSM explores 101 points of 441 grid points to find the all the positive Lyapunov exponents. In Section 5, we also show the results of these five problems with G-DACE. It seems that G-DACE does not work well on unsmooth surfaces. According

<table>
<thead>
<tr>
<th>positive L.E.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBRSM</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>25</td>
<td>29.5264</td>
<td>32.336</td>
<td>40.7615</td>
<td>46.0046</td>
<td>49.6468</td>
<td>54.29</td>
<td>93</td>
</tr>
<tr>
<td>G-DACE</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>26</td>
<td>28</td>
<td>29</td>
<td>32</td>
<td>34</td>
<td>40</td>
<td>42</td>
<td>47</td>
</tr>
<tr>
<td>efficient prob.</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.2924</td>
<td>0.2844</td>
<td>0.0034</td>
<td>0</td>
<td>0.0378</td>
<td>0.0333</td>
<td>0.0528</td>
<td></td>
</tr>
<tr>
<td>BRSM</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>31</td>
<td>36</td>
<td>37</td>
<td>42</td>
<td>46</td>
<td>49</td>
<td>56</td>
</tr>
<tr>
<td>efficient prob.</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.9920</td>
<td>0.3326</td>
<td>0.0034</td>
<td>0</td>
<td>0.0378</td>
<td>0.0333</td>
<td>0.0528</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>positive L.E.</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBRSM</td>
<td>62.0505</td>
<td>66.0642</td>
<td>70.4381</td>
<td>75.1663</td>
<td>80.0252</td>
<td>86.2741</td>
<td>91.9839</td>
<td>99.2317</td>
<td>108.914</td>
<td>116.9633</td>
</tr>
<tr>
<td>G-DACE</td>
<td>60</td>
<td>61</td>
<td>261</td>
<td>340</td>
<td>341</td>
<td>355</td>
<td>413</td>
<td>436</td>
<td>437</td>
<td>440</td>
</tr>
<tr>
<td>efficient prob.</td>
<td>0.4232</td>
<td>0.3073</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BRSM</td>
<td>65</td>
<td>69</td>
<td>70</td>
<td>76</td>
<td>84</td>
<td>93</td>
<td>100</td>
<td>120</td>
<td>122</td>
<td>139</td>
</tr>
<tr>
<td>efficient prob.</td>
<td>0.6560</td>
<td>0.6468</td>
<td>0.5103</td>
<td>0.5723</td>
<td>0.7076</td>
<td>0.8119</td>
<td>0.8291</td>
<td>0.9702</td>
<td>0.8578</td>
<td>0.9461</td>
</tr>
</tbody>
</table>

Table 2: Performance summary of NBRSM, G-DACE and BRSM for $P_L$. 
to these experiments, we believe that NBRSM is enable efficiently to find the multiple extremes, no matter the true surface is smooth or not.

A weakness of NBRSM is that the optimal points identified by NBRSM may not be the “exact” optimal point, because NBRSM need to discretize the whole experimental region first. But due to the results of our experiments, it seems that NBRSM can quickly identify the trend of the unknown response surface and the hot spot areas. Thus, in order to search the exact optimal point, we would suggest to run our NBRSM first for identifying the hot spots for possible extremes, and then the iterative optimization method, for example, Response Surface Methodology (RSM) or Pattern Search (Torczon, 1997), can be applied to search the exact optimal point.
References


