MINIMUM ENERGY DESIGNS: EXTENSIONS, ALGORITHMS, AND APPLICATIONS

A Thesis
Presented to
The Academic Faculty

by

Li Gu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2016

MINIMUM ENERGY DESIGNS: EXTENSIONS, ALGORITHMS, AND APPLICATIONS

Approved by:

Dr. Roshan Vengazhiyil, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. C. F. Jeff Wu, Co-advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Benjamin Haaland
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Brani Vidakovic
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. William Myers
Quantitative Sciences
The Procter & Gamble Company

Date Approved: 2 June 2016

To my parents,

for their continuous love and support.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Roshan Joseph Vengazhiyil. Without his tremendous guidance, immense support, and genuine passion for research, I would never be able to accomplish the work.

I am extremely thankful to my co-advisor, Professor C. F. Jeff Wu, for his guidance, assistance, and inspiration. He took care of me both academically and personally. He has not only been my academic advisor, but also an unforgettable mentor for my life.

I would like to thank Professor Godfried Augenbroe and Professor Ying Hung for having extended their support to my research and having guided and inspired me. I am also thankful to Professor Benjamin Haaland, Professor Brani Vidakovic and Dr. William Myers for serving on my dissertation committee and for their valuable comments and suggestions.

All my lab mates, Dr. Matthias Tan, Dr. Yijie Wang, Dr. Heng Su, Dr. Yuan Wang, Dr. Dianpeng Wang, Simon Mak, Chih-Li Sung, and Yuanshuo Zhao, and all my friends and colleagues, who shared time and knowledge with me at Georgia Tech, made me a wonderful and memorable graduate study. I will forever be grateful to you all.

Last but not least, my heartfelt appreciation and gratitude goes to my parents, for their continuous love and support. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Minimum Energy Design (MED) is a recently proposed technique for generating deterministic samples from any arbitrary probability distribution. The idea originated from space-filling designs in computer experiments. Most space-filling designs look for uniformity in the region of interest. In MED, some weights are assigned in the optimal design criterion so that some areas are preferred over the other areas. With a proper choice of the weights, the MED can asymptotically represent the target distribution.

In this dissertation, we improve and extend MED in three different aspects. The dissertation consists of three chapters. In Chapter 1, we propose an efficient approach that uses MED to construct proposals for an independence sampler in a Monte Carlo Markov chain, which integrates MED with Monte Carlo techniques. The MED criterion is generalized and a fast algorithm for constructing MEDs is developed in Chapter 2. Finally, in Chapter 3, we propose a new type of MEDs and a new modeling method for robust parameter design in computer experiments.

Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) methods have found wide application in studying and analyzing complex systems, among which Metropolis-Hastings algorithm is commonly used. Traditional Metropolis-Hastings proposals, which move locally, are not efficient to sample from complex distributions with multiple modes. Existing tempering methods generate multiple chains at different temperatures, but how to efficiently transfer the mixing information from high to low temperature chains is unknown and is a challenging problem. In the first

chapter, we propose a new approach to construct proposals for independence sampler using the idea of MED. Between two adjacent temperatures, MED points are selected to keep and transfer the mixing information. Final samples are generated by independence sampler with the exploratory proposals constructed by the selected MED points. Simulations and a real data example show that the proposed approach is more stable and efficient than existing tempering methods, which can save a large number of function evaluations.

When evaluations on the posterior distribution become expensive, traditional MC/MCMC methods are infeasible because of the requirement of large samples. MED is a good way to overcome this problem. It can be viewed as a "deterministic" sampling method that avoids repeated sampling in the same places, which dramatically decreases the number of required samples. However, MED has two limitations, which are improved in this chapter. One is its efficiency in integration. The integration error rate using MED points is low and can be worse than MC in high dimensional cases. In Chapter 2, we define a generalized distance and use it to generalize the MED criterion. With a proper choice of the tuning parameter, the efficiency of the generalized MEDs is greatly improved. The other limitation is the construction algorithm. An MED is constructed by a one-point-at-a-time greedy algorithm, where a global optimization is required in each iteration. The function evaluations are too many to make MED competitive to MC/MCMC methods. In Chapter 2, we develop a fast algorithm for constructing MEDs with much less function evaluations. In each iteration, the algorithm constructs simplexes to search the optimal MED point while keeping all the evaluated points as a candidate list for finding good starting points in the next iteration. The proposed algorithm is shown to have better performance with much less function evaluations.

Space-filling designs, commonly used in computer experiments try to spread out points uniformly in the experimental region. However, in robust parameter design,

when the objective is to achieve robustness against noise factors, uniformity is no longer needed in the space of noise factors. This is because noise factors usually follow non-uniform distributions such as normal distribution. It makes more sense to place points in the high probability regions where more "actions" take place. In Chapter 3, we develop new design and modeling methods for robust parameter design experiments. In the design part, a new design based on the generalized MED criterion is proposed, where different tuning parameters are used for control and noise factors. Since the design points are not equally-spaced, stationary covariance functions can lead to numerical instability in computation and tend to perform poorly in prediction. In the modeling part, we propose a simple but efficient nonstationary Gaussian process that takes into account of the experimental design structure to solve this potentially difficult problem. Both the proposed design and model are demonstrated to improve the performance over conventional methods using simulated examples and a real example on Procter and Gamble packaging process.

CHAPTER I

EXPLORATORY PROPOSALS FOR INDEPENDENCE SAMPLER

1.1 Introduction

Monte Carlo (MC) methods have found wide application in studying and analyzing complex systems. They simulate probability distributions and use the random samples to make statistical inference numerically. Markov Chain Monte Carlo (MCMC) methods, constructing Markov chains with the equilibrium distribution being the target distribution that we want to sample from, are solid and efficient tools to sample from complex distributions. Popular MCMC methods include Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970), Gibbs sampler (Geman and Geman, 1984; Gelfand and Smith, 1990), and their numerous extensions. See Brooks et al. (2011) for a review.

Among MCMC methods, MH-type algorithms play a fundamental role. They make use of a transition proposal function P. Consider to sample from a target distribution f. Given the current sample $\boldsymbol{x}$, a new sample $\boldsymbol{y}$ is drawn from $P(\boldsymbol{x}, \boldsymbol{y})$, and is accepted with the probability

$$\min\left\{1, \frac{f(\boldsymbol{y})P(\boldsymbol{y}, \boldsymbol{x})}{f(\boldsymbol{x})P(\boldsymbol{x}, \boldsymbol{y})}\right\}. \tag{1.1.1}$$

A common choice of the proposal is $P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{y} - \boldsymbol{x})$, that is, $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a random variable with mean zero. It is called Random Walk Metropolis (RWM) algorithm . Since the scale of the proposal in MH-type algorithms is in general small compared to that of the whole distribution, in every local region, the distribution becomes simple and can be well simulated by the proposal. Most of MH proposals are considered to be local moves.

Because of the local mechanism, MH proposals are easily trapped into a local mode. When the target distribution has multiple modes and there are apparent gaps between these modes, it is not easy for MH proposals to jump freely from one mode to the others. Tempering, which was first proposed in Parallel Tempering (PT) by Geyer (1991), is a popular idea for simulating multimodal distributions. In tempering methods, a decreasing series of temperatures is first defined. Multiple chains are generated at all the temperatures. The initial temperature is high enough so that the target distribution is flattened, and the proposals can move freely and provide better mixing between the modes. Note that what we want is only the samples at temperature one. One chanllenge in tempering is how to efficiently transfer the mixing information from high to low temperature chains, thus connecting multiple chains. It can be handled in different ways. PT draws multiple chains simultaneously and swaps two samples in a neighbor pair of chains occasionally. Evolutionary Monte Carlo (Liang and Wong, 2001) extended PT by adding mutation and crossover operators. Kou et al. (2006) proposed Equi-Energy (EE) sampler that generates the current chain partially by selecting the existing samples from the previous chain at the higher temperature, and partially by drawing from the proposal. Other tempering methods include Marinari and Parisi (1992), Geyer and Thompson (1995), and so on.

Apart from local proposals in MH-type algorithms, there is an exception where the proposal is not local. In independence sampler, the proposal is $P(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{y})$, that is, a new sample is drawn independently with the current sample. A good $P(\boldsymbol{y})$ should be able to approximate f well. See more discussions in Tierney (1994) and Liu (1996). The proposal for independence sampler is global rather than local. However, it is always difficult to choose an appropriate proposal before sampling, which prevents the application of independence sampler.

In this chapter, we propose a new approach for independence sampler to construct a proposal that is able to explore the target distribution before final sampling, which

is called exploratory proposal. Independence Sampler with the Exploratory Proposal (ISEP) works for all distributions, and especially for multimodal distributions. The idea of exploratory proposals stems from Minimum Energy Designs (MEDs), which was recently proposed by Joseph, Dasgupta, Tuo and Wu (2015), for generating deterministic design points from any arbitrary distributions. MED points are obtained by optimizing the energy criterion, which is quite different from existing MC and MCMC methods in which samples are generated by random sampling. Based on the deterministic viewpoint on sampling, MED points can be used in tempering to transfer the mixing information from high to low temperature chains efficiently. The mixing information of the current samples is stored in the MED points, and the subsequent chain is generated based on it.

The rest of the chapter is organized as follows. MEDs are briefly reviewed in Section 1.2. An exchange algorithm is presented in Section 1.3 for selecting MED points. In Section 1.4, we propose the main algorithm for constructing exploratory proposals. Several examples are given in Section 1.5 to illustrate ISEP.

1.2 Minimum Energy Designs

The idea of MEDs is to analogy the electric field with charged particles. Consider n design points $\boldsymbol{D} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$. Let $q(\boldsymbol{z}_i)$ be the positive charge of the particle at the ith design point $\boldsymbol{z}_i$. Given that the electric potential energy is proportional to the charges of the particles and inversely proportional to the distance of the particles, the energy of a pair of two particles $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is defined by

$$\frac{q(\boldsymbol{z}_i)q(\boldsymbol{z}_j)}{d(\boldsymbol{z}_i, \boldsymbol{z}_j)}, \tag{1.2.1}$$

where $d(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is the Euclidean distance between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. An MED is a design that minimizes the maximum energy

$$\min_{\boldsymbol{D}} \max_{i \neq j} \frac{q(\boldsymbol{z}_i)q(\boldsymbol{z}_j)}{d(\boldsymbol{z}_i, \boldsymbol{z}_j)}. \tag{1.2.2}$$

The key feature of MEDs is that if we take $q(\boldsymbol{z}) = f(\boldsymbol{z})^{1/(2p)}$, where f is the target distribution, the limiting distribution of the MED points is f. A theoretical proof for uniform distributions was given in their paper. Tuo and Lv (2016) proved that the limiting distribution holds for arbitrary distributions.

Note that the optimization on (1.2.2) can be done no matter whether the normalizing constant is known or not. MEDs clearly show some potential on sampling from target distributions. However, the potential use can be restricted by the optimization algorithm. A greedy algorithm was used for sequentially generating MED points in Joseph, Dasgupta, Tuo and Wu (2015). It is easy to implement, but is still time-consuming to run n times of global optimization in the p-dimensional space. Compared to MCMC methods that can generate hundreds of thousands samples in seconds, the computation time of generating an MED increases approximately at the rate of $p^{1.5}N^{2.25}$. For instance, 50 MED points in two dimensions can take ten seconds. On the other hand, it is sensitive to the choice of the starting point.

Wang et al. (2016) generalized the MED criterion using the following generalized distance

$$d_s(\boldsymbol{u}, \boldsymbol{v}) = \left(\frac{1}{p} \sum_{l=1}^{p} |u_l - v_l|^s \right)^{1/s}, \tag{1.2.3}$$

where $s \in (0, 2]$. When $s \to 0$, the generalized distance becomes

$$d_0(\boldsymbol{u}, \boldsymbol{v}) = \prod_{l=1}^{p} |u_l - v_l|^{1/p}. \tag{1.2.4}$$

They demonstrated that the best performance in terms of the limiting distribution can be obtained when $s \to 0$. In addition, the proof in Tuo and Lv (2016) holds for any $s \in (0, 2]$. As they suggested, we use the generalized MED criterion given by

$$\min_{\boldsymbol{D}} \max_{i \neq j} \frac{q(\boldsymbol{z}_i) q(\boldsymbol{z}_j)}{d_0(\boldsymbol{z}_i, \boldsymbol{z}_j)} \tag{1.2.5}$$

which is equivalent to

$$\max_{\boldsymbol{D}} \min_{i \neq j} \frac{1}{2} \log f(\boldsymbol{z}_i) + \frac{1}{2} \log f(\boldsymbol{z}_j) + \sum_{l=1}^{p} \log |z_{il} - z_{jl}|. \tag{1.2.6}$$

The main advantage of this criterion is that taking the logarithm of f and the distance can improve the numerical stability, which is very important since f can be very small in cases of high dimensions.

1.3 Exchange Algorithm

Instead of directly optimizing the MED criterion, we need an algorithm for selecting MED points from finite candidate points.

In the field of computer experiments, several stochastic optimization algorithms have been proposed for constructing optimal Latin Hypercube Designs (LHDs), such as local search (Li and Wu, 1997; Ye, 1998), simulated annealing (Morris and Mitchell, 1995; Joseph and Hung, 2008), and stochastic evolutionary (Jin et al., 2005) algorithms. See Fang et al. (2006) for a review. Note that the number of candidate points, np, are finite in LHDs, while searching MED points requires continuous optimization. They cannot be directly used for finding optimal MEDs, but inspired us to select MED points from finite candidate points.

We apply the simulated annealing algorithm to select MED points from finite candidate points, which will be used in the construction of exploratory proposals. The basic idea is that in each iteration, the worst MED point in the current design is replaced, with a probability, by another possibly better point in the candidate set. We called it exchange algorithm because two points are exchanged in each iteration. The details are described below.

Define the energy matrix of $\boldsymbol{D}$ by $\boldsymbol{E}$, which is an $n \times n$ matrix with entry

$$\{\boldsymbol{E}\}_{ij} = \frac{1}{2}\log f(\boldsymbol{z}_i) + \frac{1}{2}\log f(\boldsymbol{z}_j) + \sum_{l=1}^{p} \log|z_{il} - z_{jl}|. \tag{1.3.1}$$

Define a function ϕ of maximum energy on both $\boldsymbol{D}$ and $\boldsymbol{z}_j$ as follows: $\phi(\boldsymbol{D}) = \min_{i,j,i\neq j}\{\boldsymbol{E}\}_{ij}$ and $\phi(\boldsymbol{z}_j) = \min_{i,i\neq j}\{\boldsymbol{E}\}_{ij}$. Note that for computing ϕ on either $\boldsymbol{D}$ or $\boldsymbol{z}_i$, $\boldsymbol{E}$ needs to be computed first, and then the smallest one can be chosen.

Denote the given N candidate points by $\boldsymbol{C} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$.

1. Randomly pick n distinct points $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ from $\boldsymbol{C}$ as the initial $\boldsymbol{D}$.

2. For each point $\boldsymbol{z}_i \in \boldsymbol{D}$, compute the maximum energy $\phi(\boldsymbol{z}_i)$. Find $\boldsymbol{z}^*$ such that

$$\boldsymbol{z}^* = \arg\max_{\boldsymbol{z}_i \in \boldsymbol{D}} \phi(\boldsymbol{z}_i). \tag{1.3.2}$$

   It follows that $\boldsymbol{z}^*$ is the worst MED point in $\boldsymbol{D}$. Note that there is a pair of points that share the same value of ϕ, and it does not matter which one to be chosen.

3. Randomly pick another point $\boldsymbol{z}_{new}$ in $\boldsymbol{C}$. Compute $\phi(\boldsymbol{D}_{new})$, where $\boldsymbol{D}_{new} = \boldsymbol{D} \setminus \boldsymbol{z}^* \cup \boldsymbol{z}_{new}$.

4. With the probability of $\pi = \min\{\exp(\phi(\boldsymbol{D}_{new}) - \phi(\boldsymbol{D})/t), 1\}$, $\boldsymbol{D} = \boldsymbol{D}_{new}$, where t is a temperature parameter and gradually decreases to zero as the procedure goes; with the probability of $(1 - \pi)$, keep $\boldsymbol{D}$.

5. Go to Step 3 until it converges.

We have some remarks for the exchange algorithm. First, for computing $\phi(\boldsymbol{D}_{new})$ in Step 3, $\boldsymbol{E}_{new}$, which is the energy matrix of $\boldsymbol{D}_{new}$, needs to be re-evaluated in each iteration, which leads to slow computation. Instead of evaluating all the $(n-1)^2/2$ entries in $\boldsymbol{E}_{new}$, it can be simplified as follows. For $\boldsymbol{z}^*$ being the i^*th point in $\boldsymbol{D}$, we can compute the energy between $\boldsymbol{z}_{new}$ and all the points in $\boldsymbol{D}$ expect $\boldsymbol{z}^*$, and update the i^*th row and i^*th column of $\boldsymbol{E}$, where the number of evaluations is only $(n-1)$.

Second, in Step 4, based on the comparison between $\phi(\boldsymbol{D}_{new})$ and $\phi(\boldsymbol{D})$, a rule is set to decide if $\boldsymbol{D}$ is updated by $\boldsymbol{D}_{new}$. The rule can vary from different stochastic optimization algorithms. Here, we adopt the simulated annealing rule in the algorithm. The choice of the temperature parameters is referred to Fang et al. (2006). Other rules can be applied similarly.

1.4 Exploratory Proposals

We split the construction of exploratory proposals into two steps: exploration and selection. Multiple MCMC chains are run to explore the target distribution at the current temperature. Then, representative points based on the MED criterion are selected to keep the mixing information at the next temperature. The two steps iterate several times as the temperatures decrease to one. An exploratory proposal is finally constructed based on the MED points at temperature one. A flow chart for constructing exploratory proposals is shown in Figure 1.

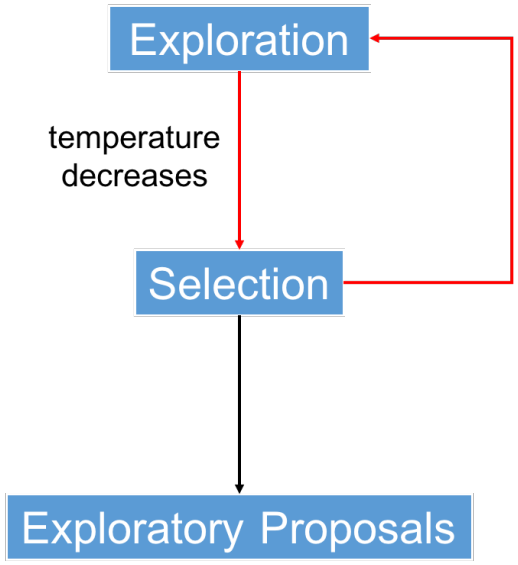


Figure 1: Flow chart for constructing exploratory proposals.

1.4.1 Construction Algorithm

The exploration step begins with n given points, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$. Considering each $\boldsymbol{z}_i$ to be a starting point, we generate an m-sample chain by an MCMC method. In total, we have n chains with $N = nm$ samples. Denote them by $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$.

Next step is selection. Let the candidate set $\boldsymbol{C}$ be $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. Apply the exchange algorithm to select n MED points from $\boldsymbol{C}$. Denote them by $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$.

In the exploration step, all the MCMC chains run freely and independently with distinct starting points, which enables the exploration step to visit every local region of the target distribution. The performance, of course, highly depends on the choice of the starting points. We obtain $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ in the following ways. An initial design with good space-filling properties is reasonable for the first iteration, since the knowledge of the target distribution is completely zero at this stage. We generate n_0 points from low discrepancy sequences (*e.g.*, Sobol sequences) or space-filling design points (*e.g.*, maximin LHDs). Sobol sequences are adopted throughout the chapter. The points that have nearly zero values of the distribution are screened out. So n_0 should be large enough so that at least n points can be left after the screening.

For the following iterations, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ are selected in the previous selection step. They are representatives of the target distribution at the current temperature and become the starting points for the next exploration step. As a comparison, Gelman and Rubin (1992) used multiple chains to monitor the convergence and to make better inference, where the modes were considered to be the starting points. Since MED points mimic the whole distribution, besides the modes, the selection step will also provide a few points proportionally to the distribution, for guaranteeing the performance on other local regions.

The two steps iterate with tempering. Define a decreasing series of temperatures $T_0 \geq \cdots \geq T_L = 1$. Instead of sampling directly from the target distribution f, in iteration l, the exploration step draws samples from $f^{1/T_{l-1}}$, and the subsequent selection step works on f^{1/T_l}.

The high temperature in the first iteration flattens the distribution, so that the proposals can move freely from one local region to another, and mix in all the regions. As the temperature gradually decreases, the distribution is cooling and becomes spiky. The MED points selected in the following iterations will shrink and concentrate around each local region, from which multiple chains are able to explore

every local region. Because MED points are representative, all the mixing information in $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ at T_{l-1} is stored in $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ at T_l. Note that f^{1/T_L} returns to be f since $T_L = 1$.

In final sampling, given the n selected MED points $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ at temperature $T_L = 1$, an independence sampler is employed with the exploratory proposal

$$\frac{1}{n}\sum_{i=1}^{n} N(\boldsymbol{z}_i, s\boldsymbol{\Sigma}_i), \tag{1.4.1}$$

where s is a scale parameter and $\boldsymbol{\Sigma}_i$, depending on the local information around $\boldsymbol{z}_i$, differs from each other. The details of choosing the parameters will be given in Section 1.4.3.3. We call the sampling procedure Independence Sampler with the Exploratory Proposal (ISEP).

The scheme of ISEP is summarized in the Table 1.

Table 1: Scheme of ISEP

Independence Sampler with the Exploratory Proposal
Set $T = T_0$
Generate n initial points $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$
for $l = 1, 2, \ldots, L$
Draw n m-sample chains $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, starting from each $\boldsymbol{z}_i$
Set $T = T_l$
Select n MED points $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ from $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
endfor
Draw N samples with proposal (1.4.1)

1.4.2 Remarks

In inference, tempering methods are sometimes considered inefficient for not utilizing samples at temperatures $\{T_2, \ldots, T_L\}$. One way to improve the efficiency is in estimating expectations, combining all samples at all temperatures based on importance sampling weights. Optimal weights were discussed in Gramacy et al. (2010). However, such a framework does not apply to the samples generated in exploration steps for constructing exploratory proposals. Because of the usage of multiple chains,

we cannot claim that samples at each temperature can shortly converge to the true distribution, which is used in the computation of weights. Nevertheless, this is in fact flexibility of exploratory proposals. The convergence of samples from multiple chains at each temperature is not required at all. The samples at the current temperature are used only for exploring the target distribution and for selecting MED points at the next temperature to store the mixing information. As long as the samples can well spread out and explore the distribution, the performance on MED points is guaranteed. Finally, ISEP can correctly generate samples. This gives exploratory proposals flexibility that much less samples at each temperature are required in each exploration step, which improves the efficiency as well.

It is worth mentioning that for complex but unimodal distributions, tempering is not required. We can set $L = 1$ and $T_0 = T_1 = 1$, so that the exploration step and the selection step will iterate only once where the temperature keeps one. Thanks to one iteration of exploration and selection, ISEP can still improve on these distributions, compared to MH-type algorithms. See Section 1.5.1.

1.4.3 Implementation Details

Some details in practical implementation are given below.

1.4.3.1 Exploration Step

The number of starting points and the number of MED points in each iteration are not necessarily identical. One can decrease the number of MED points gradually as we know more and more about the distribution to make computation faster. For simplicity, we take the same n. The choice of n depends on the complexity of the target distribution. Based on prior knowledge, if each local region is simple, say spherical, $c_1 p$ points that are supposed to be assigned for each one local region are enough, where $c_1 \in [1, 3]$. If some local regions are more complex, more points will be needed. Since the points with nearly zero values will be screened out, for the initial

design in the first exploration step, $n_0 = c_2 n$ is appropriate, where $c_2 \in [5, 10]$.

Both low discrepancy sequences and space-filling design points have a pre-specific range (usually a hypercube $[0, 1]^p$). They need to be transformed to the range of the target distribution, which brings up the question on how to define the range of the target distribution. A rough estimation of the range can be obtained from prior knowledge. Then, an adjustment may be needed based on the evaluations on these points. We can expand the range if the points near the boundary have high values, or narrow it if those have near zero values. Since the points serve as the starting points of multiple chains, which can move freely within each of the local regions, the range does not need to be exactly accurate, as long as it overlaps all the local regions.

The choice of the MCMC method for multiple chains is flexible. It should be able to explore the local regions. It can be RWM algorithm for simplicity, Gibbs sampler when the dimension is high, or other sampling methods that are suitable for the target distribution at the current local region. The only criterion is that the acceptance rate of the sampler can be slightly lower than the typical one, because the objective in the exploration step is to widely explore the target distribution, rather than getting more samples for inference.

For each chain, the length m is not necessarily the same. But except that we have already known some local regions are much more complex than the others, there is no particular benefit for choosing a different m. A good choice is simply $m = N/n$.

1.4.3.2 Tempering

The initial temperature T_0 should be large enough so that all the chains can move freely across all the local regions. It can be found after some preliminary trials. We should conservatively choose a large T_0 in order to guarantee all the local regions are connected. The series of temperatures is decided by a rough guideline that $\log T_i$ are equally spaced (Kou et al., 2006). The decreasing temperatures reflect the shrinkage

rate of the target distribution. Because of the deterministic way to choose representative points, the shrinkage rate can be larger than that of existing tempering methods, which is an advantage of ISEP.

The number of temperatures L depends on the complexity of the target distribution. We need more temperatures if the target distribution has higher dimensions or more modes. L can start from the number of dimensions and is increased if some clear discrepancies are found between two adjacent temperatures.

1.4.3.3 Final sampling

The parameters in the exploratory proposal (1.4.1) are decided based on the principle that the proposal for the independence sampler should be close to the target distribution. An optimization on minimizing the error between the exploratory proposal and the target distribution over all the parameters is appropriate.

Note that $\mathbf{\Sigma}_i$ has $p \times p$ unknown parameters, and optimizing n of $\mathbf{\Sigma}_i$ can be troublesome. However, since $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ are MED points that already represent the target distribution, it is easy to define a good $\mathbf{\Sigma}_i$ based on the local information around $\boldsymbol{z}_i$. A straightforward idea is that $\mathbf{\Sigma}_i = \mathrm{diag}(\min_j(z_{i1} - z_{j1})^2, \ldots, \min_j(z_{ip} - z_{jp})^2)$, but it does not perform stably.

For a more stable result, we use the average of the $2p$ closest points as follows.

For any $i = 1, \ldots, n$, $l = 1, \ldots, p$, rank $(z_{il} - z_{jl})^2$ in ascending order for all $j = 1, \ldots, n$, $j \neq i$, so we have that $(z_{il} - [z_{1l}])^2 \leq (z_{il} - [z_{2l}])^2 \leq \cdots \leq (z_{il} - [z_{(n-1),l}])^2$. The diagonal terms of $\mathbf{\Sigma}_i$ are defined by

$$(\mathbf{\Sigma}_i)_{ll} = \frac{1}{2p} \sum_{j=1}^{2p} (z_{il} - [z_{jl}])^2. \tag{1.4.2}$$

Note that these dimension-wise distances are necessary since different dimensions may have very different scales.

After $\mathbf{\Sigma}_i$ is defined, s can be obtained by optimization. Consider least squares of

the error

$$\min_{c,s} \sum_{j=1}^{N} \left\{ f(\boldsymbol{x}_j) - \frac{c}{n} \sum_{i=1}^{n} N(\boldsymbol{x}_j; \boldsymbol{z}_i, s\boldsymbol{\Sigma}_i) \right\}^2, \tag{1.4.3}$$

where c is a normalizing constant. It is easy to see that in order to minimize (1.4.3), c can be written as a function of s given by

$$\widehat{c}(s) = \frac{\sum_{j=1}^{N} \frac{1}{n} \sum_{i=1}^{n} N(\boldsymbol{x}_j; \boldsymbol{z}_i, s\boldsymbol{\Sigma}_i) f(\boldsymbol{x}_j)}{\sum_{j=1}^{N} \left(\frac{1}{n} \sum_{i=1}^{n} N(\boldsymbol{x}_j; \boldsymbol{z}_i, s\boldsymbol{\Sigma}_i) \right)^2}. \tag{1.4.4}$$

Then, the objective function becomes

$$\min_{s} \sum_{j=1}^{N} \left\{ f(\boldsymbol{x}_j) - \frac{\widehat{c}(s)}{n} \sum_{i=1}^{n} N(\boldsymbol{x}_j; \boldsymbol{z}_i, s\boldsymbol{\Sigma}_i) \right\}^2, \tag{1.4.5}$$

which is one dimensional w.r.t. s.

1.5 Examples

1.5.1 Mixture of Bivariate Normal Distributions

We first check the performance of ISEP without tempering. Consider a mixture of three bivariate normal distributions (Gilks et al., 1998). The target distribution is

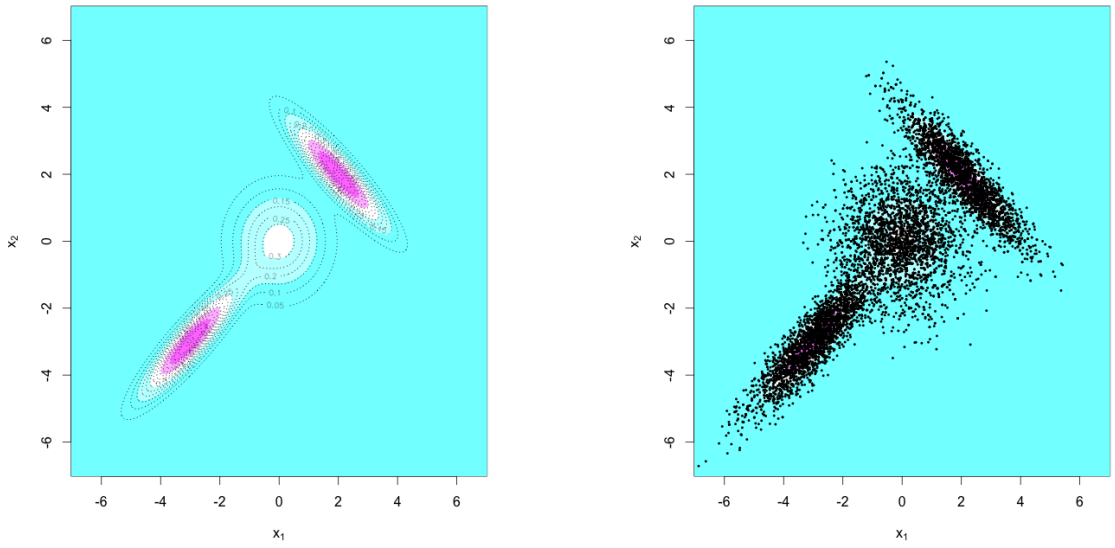
$$f(\boldsymbol{x}) = 0.34N(\boldsymbol{x}, (0,0)^T, \boldsymbol{\Sigma}_1) + 0.33N(\boldsymbol{x}, (-3,-3)^T, \boldsymbol{\Sigma}_2) + 0.33N(\boldsymbol{x}, (2,2)^T, \boldsymbol{\Sigma}_3), \tag{1.5.1}$$

where

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}, \text{ and } \boldsymbol{\Sigma}_3 = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}.$$

In this example, although the three modes are connected and not far away from each other, since the covariance matrices are very different, sampling is still a difficult task for MH-type algorithms. See Figure 2a for an illustration of the target distribution.

We compared ISEP to RWM algorithm and Adaptive Metropolis (AM) algorithm (Haario et al., 2001). AM algorithm is an efficient method to adaptively choose appropriate proposals, where the current proposal is updated based on the information



(a) Target distribution

(b) Samples of ISEP

Figure 2: Mixture of bivariate normal distributions.

of the previous samples. The total number of samples was $N = 10,000$ after 2,000 samples burn-in. The proposal of RWM algorithm was a scaled standard normal distribution for the 23.4% optimal acceptance rate. For AM algorithm, R package "MHadaptive" (Chivers, 2012) with default parameters was used. For ISEP, $n_0 = 300$ initial points were generated from the Sobol sequence. The number of MED points was $n = 50$. A RWM algorithm was used for multiple chains in the exploration step. Because the target distribution is relatively simple, we chose only one temperture $T_0 = 1$. Because of the use of independence sampler, there is no need to burn-in. The result of one simulation of ISEP is shown in Figure 2b.

Sample mean and covariance matrix were compared. The Mean Square Errors (MSEs) with 100 replications are summarized in the Box plots shown in Figure 3. We can see that ISEP always give the most accurate estimations for all the three quantities. Moreover, the variance of ISEP is significantly lower than the other two methods, which is due to the deterministic way to choose representative points.

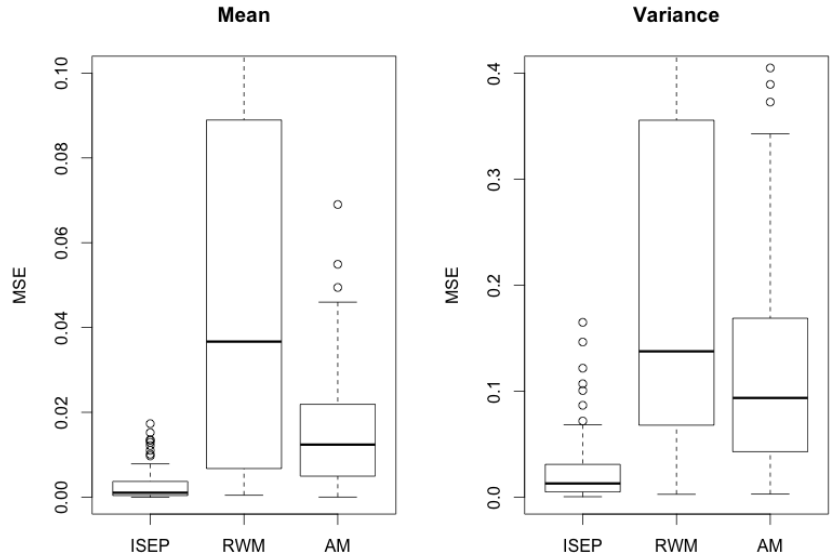


Figure 3: MSEs for the mixture normal distribution.

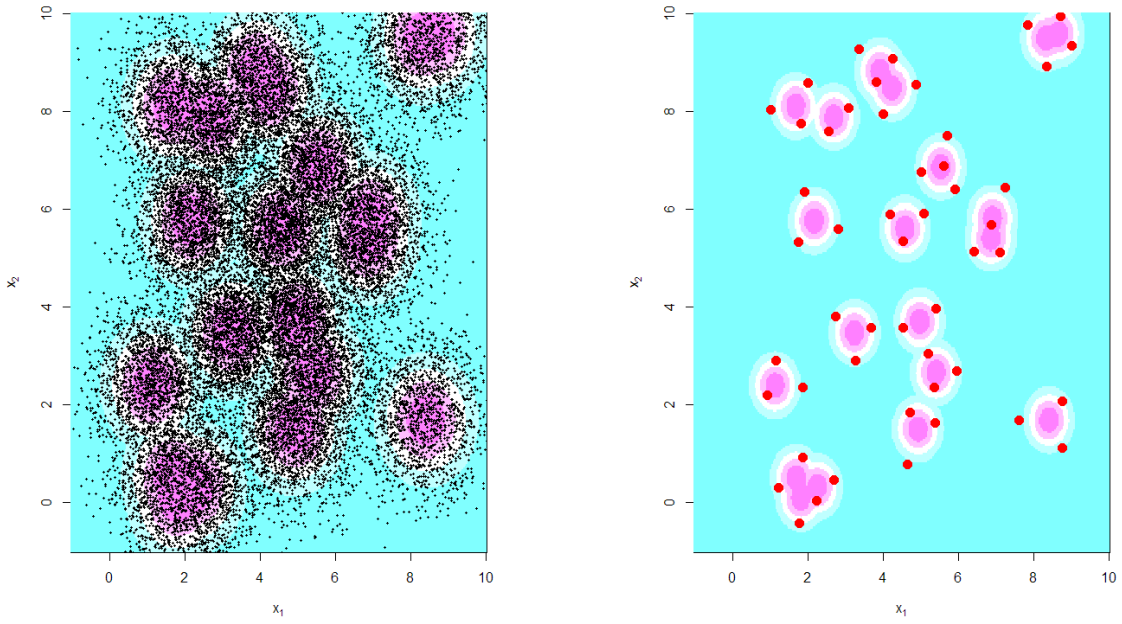
1.5.2 Multimodal Distribution

A multimodal example was presented in Liang and Wong (2001). It was also studied in Kou et al. (2006). The target distribution is

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{i=1}^{20} \omega_i \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T(\boldsymbol{x}-\boldsymbol{\mu}_i)\right\}, \tag{1.5.2}$$

where $\sigma = 0.1$ and $\omega_1 = \cdots = \omega_{20} = 0.05$ and μ_i are given in Table 2. In the target distribution, 20 normal distributions with the same spherical covariance are mixed with the same weight. Some of the modes are far from others, which make the proposal difficult to jump. Tempering is widely used in this example. We compared ISEP to PT and EE sampler. Note that different from the other two methods, ISEP draws the current chain based on the multiple proposals, of which the centers are the MED points selected from the previous higher temperature chain.

The same settings of PT and EE sampler in Kou et al. (2006) were applied. The total number of samples was $N = 50,000$ after burn-in. The temperature parameters were $T = \{60, 21.6, 7.7, 2.8, 1\}$. So the results can be directly compared. For ISEP,



(a) Exploration

(b) Selection after temperature decreasing

Figure 4: Multimodal distribution. Black dots denote the samples and Red dots the MED points.

Table 2: Means of the 20 modes of the multimodal distribution.

i	μ_{i1}	μ_{i2}	i	μ_{i1}	μ_{i2}	i	μ_{i1}	μ_{i2}	i	μ_{i1}	μ_{i2}
1	2.18	5.76	6	3.25	3.47	11	5.41	2.65	16	4.93	1.50
2	8.67	9.59	7	1.70	0.50	12	2.70	7.88	17	1.83	0.09
3	4.24	8.48	8	4.59	5.60	13	4.98	3.70	18	2.26	0.31
4	8.41	1.68	9	6.91	5.81	14	1.14	2.39	19	5.54	6.86
5	3.93	8.82	10	6.87	5.40	15	8.33	9.50	20	1.69	8.11

the number of initial points were $n_0 = 200$ generated from the Sobol sequence, and $n = 50$ MED points were selected at each temperature. A RWM algorithm was used in the exploration step. The temperature parameters were $T = \{60, 15.3, 3.9, 1\}$. As an illustration, we can see from Figure 4 that multiple chains have explored the distribution under the current temperature $T_0 = 60$, and MED are good representatives for the distribution under the next lower temperature $T_1 = 15.3$. The two steps are iterated two more times until the temperature reduces to one.

Table 3: MSEs for the multimodal distribution.

MSE	$E\boldsymbol{x}_1$	$E\boldsymbol{x}_2$	$E\boldsymbol{x}_1^2$	$E\boldsymbol{x}_2^2$
PT	0.03244	0.080765	3.318025	8.324277
EE	0.01202	0.020834	1.307429	2.194599
ISEP	0.00092	0.001155	0.095324	0.117447

First and second moments on each dimension were compared. The MSEs with 20 replications are given in Table 3. We can see that ISEP improves the performance a lot over the two existing methods on all the moments. Besides, PT and EE sampler used five temperatures, while we used four in ISEP. For the computation time, the evaluation times on the target distribution were $250,000$ in PT and EE sampler, and $200,200$ in ISEP.

1.5.3 Galaxy Data

The galaxy example was first presented in Postman et al. (1986), and has been studied by several statisticians (Chib, 1995; Neal, 1999; Liang and Wong, 2001). The galaxy dataset comprises the velocities of 82 galaxies from six well-separated conic sections

of the corona borealis region. Denote them by $\boldsymbol{y} = (y_1, \ldots, y_{n_y})$, where $n_y = 82$. The objective is to find a Gaussian mixture model that can fit the data well. Consider a Gaussian mixture model of d components. The likelihood function is

$$L(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n_y} \sum_{j=1}^{d} \omega_j \phi(y_i|\mu_j, \sigma_j^2), \tag{1.5.3}$$

where $\phi(y_i|\mu_j, \sigma_j^2)$ is the probability density function of the normal distribution with mean μ_j and variance σ_j^2, ω_j is the mixing proportion. Denote all the parameters by $\boldsymbol{x} = (\omega_1, \ldots, \omega_{d-1}, \mu_1, \ldots, \mu_d, \sigma_1^2, \ldots, \sigma_d^2)$. All the components are mutually independent and with the prior distributions

$$\mu_j \sim N(\mu_0, \sigma_0^2); \tag{1.5.4}$$

$$\sigma_j^2 \sim \text{IG}\left(\nu_0/2, \delta_0/2\right); \tag{1.5.5}$$

$$(\omega_1, \ldots, \omega_d) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_d), \tag{1.5.6}$$

where $\mu_0 = 20, \sigma_0^2 = 100, \nu_0 = 6, \delta_0 = 40, \alpha_1 = \cdots = \alpha_d = 1$, which follows the same setting in Chib (1995). Denote the prior distribution by $\pi(\boldsymbol{x})$.

The quantity of interest is the marginal likelihood function

$$m(\boldsymbol{y}) = \int L(\boldsymbol{y}|\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}, \tag{1.5.7}$$

which has no analytical form. From Liang and Wong (2001), it can be evaluated using bridge sampling (Meng and Wong, 1996). Consider two distributions that are known up to normalizing constants, that is, $f_1(\boldsymbol{x}) = g_1(\boldsymbol{x})/c_1$ and $f_2(\boldsymbol{x}) = g_2(\boldsymbol{x})/c_2$, where g_1 and g_2 are known. In this example, let $g_1 = L(\boldsymbol{y}|\boldsymbol{x})\pi(\boldsymbol{x})$ and $g_2 = \pi(\boldsymbol{x})$. Bridge sampling can be used to iteratively estimate the ratio $r = c_1/c_2$ by

$$\widehat{r}^{(t+1)} = \frac{\frac{1}{n_2}\sum_{j=1}^{n_2} \frac{l_{2j}}{s_1 l_{2j} + s_2 \widehat{r}^{(t)}}}{\frac{1}{n_1}\sum_{j=1}^{n_1} \frac{l_{1j}}{s_1 l_{1j} + s_2 \widehat{r}^{(t)}}}, \tag{1.5.8}$$

where for $i = 1, 2$ and $j = 1, \ldots, n_i$, $s_i = n_i/(n_1 + n_2)$, $l_{ij} = g_1(\boldsymbol{x}_{ij})/g_2(\boldsymbol{x}_{ij})$, $\{\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{in_i}\}$ are the samples drawn from g_i, and $\widehat{r}^{(t)}$ is the estimated value of r

in the iteration t. $\widehat{r}$ can be iteratively computed starting from any initial value larger than zero. Note that throughout all the iterations, sampling is required only once. Then, in the galaxy example, since $c_1 = \int L(\boldsymbol{y}|\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}$ and $c_2 = \int \pi(\boldsymbol{x})d\boldsymbol{x} = 1$, the marginal likelihood $m(\boldsymbol{y})$ can be estimated by $\widehat{r}$.

Following the bridge sampling, samples are required from both g_1 and g_2. g_2 is easy to directly sample from. The challenge is on g_1, which is a complex multimodal distribution because the components in the Gaussian mixture model can be randomly permuted. We apply ISEP to draw samples from g_1. The target distribution is

$$f_i(\boldsymbol{x}) \propto \{L(\boldsymbol{y}|\boldsymbol{x})\}^{T_i} \pi(\boldsymbol{x}), \tag{1.5.9}$$

where T_i is the temperature.

We ran simulations on the mixture models with two to five components. So the dimension is up to 14. For each model, 20 simulations were replicated. In each one simulation, the total number of samples was $N = 25,000$. The temperature parameters were chosen as $T = \{20, 14.4, 10.3, 7.4, 5.3, 3.8, 2.7, 1.9, 1.4, 1\}$, 10 temperatures in total. The number of initial points was $n_0 = 200p$, and the number of MED points was $n = 20p$. A RWM algorithm within Gibbs sampling was used in the exploration step. It is easy to find that the conditional distribution on each parameter is proportional to the likelihood times its prior.

Table 4: Marginal log-likelihoods for the galaxy data.

Model	Chib	Neal	EMC	ISEP
2E	-240.464 (.006)	-239.764 (.005)	-239.744 (.015)	-240.143 (.001)
3E	-228.620 (.008)	-226.803 (.040)	-226.828 (.061)	-226.796 (.017)
3UE	-224.138 (.086)	-226.791 (.089)	-226.780 (.058)	-226.788 (.019)
4UE			-226.629 (.061)	-226.684 (.020)
5UE			-226.394 (.062)	-226.503 (.027)

Figure 5 shows the samples on $(\mu_1, \ldots, \mu_d)$ in one simulation of the Gaussian mixture model with three components, where we can see that all the multiple modes have been visited. The results on the marginal log-likelihood are summarized in Table

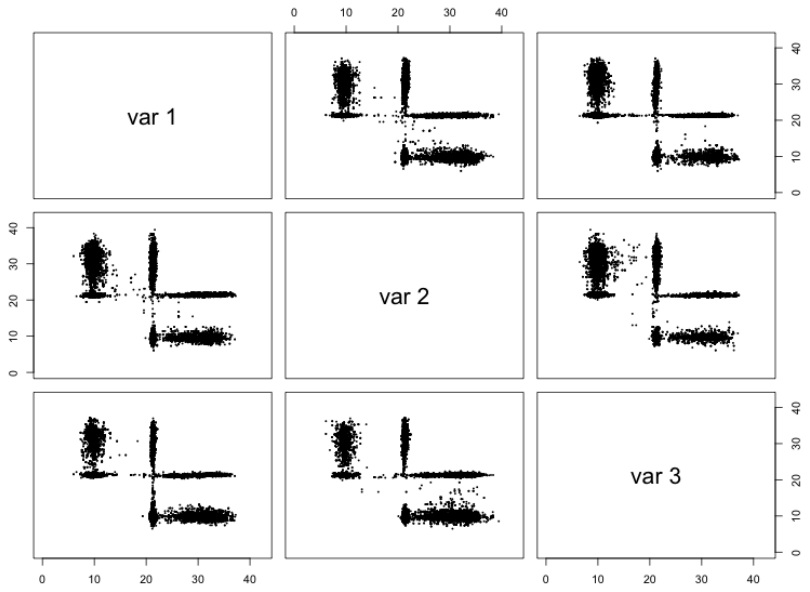


Figure 5: Samples on the means of the Gaussian mixture model for the galaxy data.

4. In the column of "Model", the number denotes the number of components, "E" equal variance, and "UE" unequal variances. The results in ISEP are comparable with those in the other three methods with similar settings, which indicates that the mixture model with three components fits the data best, and those with four or five components have similar marginal log-likelihoods but overfit the data.

The computation time is an advantage of ISEP, which is approximately proportional to the evaluation times of the likelihood function. EMC used 20 temperatures, where 12.5 evaluations were conducted in each iteration at each temperature. So in total, it took 6.25×10^6 evaluations. For ISEP, we need to evaluate $25,000$ times at each temperature. Since only ten temperatures were used, the total number of evaluations was 2.5×10^5.

1.6 Conclusions

In this chapter, we proposed a new approach to construct proposals for independent sampler. By incorporating MEDs into tempering, the mixing information is

transferred from high to low temperature chains much more efficiently than existing tempering methods. The proposed ISEP works well for both complex distributions (without tempering) and multimodal distributions (with tempering).

MED points are obtained by optimization for representing the target distribution. Compared to other stochastic tempering methods, MED points can store more mixing information with less points, which means less temperatures are needed to obtain a comparable performance. Note that samples of length N is needed for one temperature. This dramatically decreases the number of function evaluations.

CHAPTER II

BAYESIAN COMPUTATION USING MINIMUM ENERGY DESIGNS

2.1 Introduction

The main challenge in Bayesian computation is the efficient evaluation of high dimensional integrals arising in Bayesian models. Monte Carlo (MC) and Markov chain Mote Carlo (MCMC) methods are commonly used for this purpose. They work by drawing samples from the posterior and then approximating the integrals using sample averages. Efficient methods for MC/MCMC sampling are proposed in the literature, see Brooks et al. (2011) for a review. However, these methods can be costly in terms of the number of evaluations made on the posterior distribution. This cost is often neglected, especially when the posterior is easy to evaluate. But when the posterior is complex and expensive to evaluate, the cost becomes appreciable. It is not uncommon for the researchers to wait several hours or even days for the MCMC chain to converge and produce final results. This becomes frustrating for the researcher when he/she has to go back and run the chains all over again when minor tweaks are made in the models.

We can overcome the aforementioned problem if we can devise a method that requires only few evaluations of the posterior. We propose to do this by replacing the "random" sampling with "deterministic" sampling. To explain the concepts, let us introduce some notations. Let $f(\boldsymbol{x})$ be the posterior density of the parameters $\boldsymbol{x}$ given the data. Consider two points in the parameter space $\boldsymbol{a}$ and $\boldsymbol{b}$. If $f(\boldsymbol{b})/f(\boldsymbol{a}) = 10$, then MC/MCMC methods would require 10 times more samples in the neighborhood of $\boldsymbol{b}$ than those in the neighborhood of $\boldsymbol{a}$. This is clearly unnecessary because we

know everything about $f(\boldsymbol{x})$ in those two neighborhoods with just two evaluations of it at $\boldsymbol{a}$ and $\boldsymbol{b}$ (assuming $f(\cdot)$ to be sufficiently smooth). Deterministic sampling methods will try to achieve exactly this by avoiding repeated sampling in the same places and making the samples as apart as possible. This is not a new concept because the quadrature method does exactly the same thing, but of course, they do not work in high dimensions. Quasi-Monte Carlo (QMC) techniques try to overcome some of the limitations of the quadrature methods in high dimensions, but they are also not as popular as MC/MCMC methods for a good reason. QMC methods are mainly developed for sampling from hypercubes. Unfortunately, the posterior distributions can be highly correlated and nonlinear making them occupy very little space in a hypercube. Thus, most of the samples from QMC can get wasted. The QMC samples can be saved if they can be pulled towards the high probability regions of the distribution using inverse probability transforms. But this can be done only when the distribution function is known, which is rarely the case in Bayesian problems.

The difficulty with the QMC can be avoided if we can deterministically and directly sample the points from the posterior distribution. One such method was proposed by Joseph, Dasgupta, Tuo and Wu (2015) known as Minimum Energy Design (MED). This method draws ideas from experimental designs in computer experiments. Most experimental designs look for uniformity in the region of interest. The idea behind MED is to assign some weights in the optimal design criterion so that some areas are preferred over the other areas. Joseph, Dasgupta, Tuo and Wu (2015) showed that by judiciously choosing the weights, the design points can be made to mimic the target distribution. Unfortunately, this idea comes with a price. Choosing the weights and finding the optimal experimental design require numerous evaluations of the posterior distribution and tedious global optimizations making MED noncompetitive to the random sampling-based MC/MCMC methods for most Bayesian problems. This chapter tries to overcome this serious deficiency of MEDs by proposing an efficient

procedure for generating them. Moreover, a generalization of the MED criterion is proposed, which is crucial for improving its performance in high dimensions.

There is another approach to overcome the computational problems with expensive posteriors. One can approximate the unnormalized posterior with an easy-to-evaluate model and then work on the approximate model instead of the exact posterior. This is the approach taken by many: Rasmussen et al. (2003), Bliznyuk et al. (2008), Fielding et al. (2011), Bornkamp (2011), and Joseph (2012, 2013). However, the modeling-based methods are severely limited by the curse of dimensionality. That is, tuning the modeling becomes extremely difficult in high dimensions leading to poor approximations. At this moment it is not clear if the deterministic sampling method proposed in this chapter can overcome this problem, but it is clearly a promising alternative.

This chapter is organized as follows. In Section 2.2, we review MED and provide a generalized version of MED to improve the efficiency. A fast algorithm for constructing MEDs with much less function evaluations is developed in Section 2.3. Section 2.4 provides a method of local approximation to further save function evaluations of the proposed algorithm. Examples are given to illustrate the proposed algorithm in Section 2.5. Section 2.6 concludes this chapter with some remarks.

2.2 Minimum Energy Designs

Let $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be the set of deterministic points from the posterior distribution, where each $\boldsymbol{x}_i$ is a p-dimensional vector in $\mathbb{R}^p$. It is called a minimum energy design (MED) if it minimizes

$$E(\boldsymbol{D}) = \max_{i \neq j} \frac{q(\boldsymbol{x}_i)q(\boldsymbol{x}_j)}{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}, \tag{2.2.1}$$

where $q(\boldsymbol{x})$ is called a charge function and $d(\boldsymbol{u}, \boldsymbol{v})$ is the Euclidean distance between the points $\boldsymbol{u}$ and $\boldsymbol{v}$. Joseph, Dasgupta, Tuo and Wu (2015) showed that if $q(\boldsymbol{x}) = 1/f^{1/(2p)}(\boldsymbol{x})$ and if the MED has the smallest index, then the empirical distribution of

the design points will converge to $f(\boldsymbol{x})$ as $n \to \infty$. Here, the index of a design refers to the number of pairs of points with the maximum energy $E(\boldsymbol{D})$. For the rest of the chapter, we will ignore the index because our numerical algorithms rarely finds two pairs with the same energy, especially for nonuniform distributions. Joseph et al.'s proof for the limiting distribution of MED was only heuristic. Recently, Tuo and Lv (2016) was able to give a rigorous proof for this important result.

Thus, our objective is to find a design that minimizes

$$\max_{i \neq j} \frac{1}{f^{1/(2p)}(\boldsymbol{x}_i) f^{1/(2p)}(\boldsymbol{x}_j) d(\boldsymbol{x}_i, \boldsymbol{x}_j)},$$

or equivalently, a design that maximizes

$$\psi(\boldsymbol{D}) = \min_{i \neq j} f^{1/(2p)}(\boldsymbol{x}_i) f^{1/(2p)}(\boldsymbol{x}_j) d(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{2.2.2}$$

An important property that makes this method suitable for Bayesian problems is that we only need to know $f(\cdot)$ up to a constant of proportionality because the proportionality constant does not affect the optimization. So in most Bayesian problems, we take $f(\cdot)$ to be the unnormalized posterior. Clearly, an MED will try to place points as apart as possible and in regions where the density is high. Moreover, for finite n, the empirical distribution of MED can be considered as an approximation to the target distribution. Thus, MED has all the qualities of a "deterministic" sample that we are looking for.

2.2.1 Limitations

Maximizing $\psi(\boldsymbol{D})$ in (2.2.2) to find an MED is not an easy problem. Joseph, Dasgupta, Tuo and Wu (2015) proposed a one-point-at-a-time greedy algorithm. The idea is to start with a point $\boldsymbol{x}_1$ and generate $\boldsymbol{x}_2, \boldsymbol{x}_3, \ldots$ sequentially. The $(n+1)$th design point is obtained by

$$\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x}} \min_{j=1:n} f^{1/(2p)}(\boldsymbol{x}) f^{1/(2p)}(\boldsymbol{x}_j) d(\boldsymbol{x}, \boldsymbol{x}_j). \tag{2.2.3}$$

Extensive simulations conducted by Joseph, Dasgupta, Tuo and Wu (2015) showed that this greedy algorithm works well as long as $\boldsymbol{x}_1$ is a "good" point of the posterior distribution such as posterior mode. However, each step of the algorithm requires a global optimization and numerous evaluations of the density $f(\cdot)$, which somewhat defeats the original motivation for this deterministic sampling method. In the next section, we propose an efficient algorithm to generate an MED that overcomes this major limitation.

There is another serious limitation of MED, which can be explained using an example. Figure 6(a) shows a 25-point MED for a uniform distribution in $[0,1]^2$. This is a full factorial design with five levels for each factor. This structure of the design is expected because the MED reduces to a maximin distance design when $f(\cdot)$ is uniform. A factorial-type design is not good in high dimensions because the number of projected points in each dimension from an n-run design reduces to $n^{1/p}$. Therefore, even if we use a quadrature method that converges at the rate of $O(1/n^3)$ such as Simpson's rule, the effective rate in p dimensions reduces to $O(1/n^{(3/p)})$. This can quickly become worse than an MC sample error rate of $O(1/n^{1/2})$ in high dimensions (when $p > 6$ in this case). We propose an idea to overcome this limitation of MED.

2.2.2 Generalization

Define a generalized distance

$$d_s(\boldsymbol{u}, \boldsymbol{v}) = \left(\frac{1}{p} \sum_{l=1}^{p} |u_l - v_l|^s \right)^{1/s}, \tag{2.2.4}$$

where $s \in (0, 2]$. For $s < 1$, $d_s(\cdot, \cdot)$ is not a metric, but as we show below that it has the desirable properties that are needed to achieve our objectives. Using this generalized distance, the MED criterion becomes

$$\max_{\boldsymbol{D}} \psi(\boldsymbol{D}) = \min_{i \neq j} f^{1/(2p)}(\boldsymbol{x}_i) f^{1/(2p)}(\boldsymbol{x}_j) d_s(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{2.2.5}$$

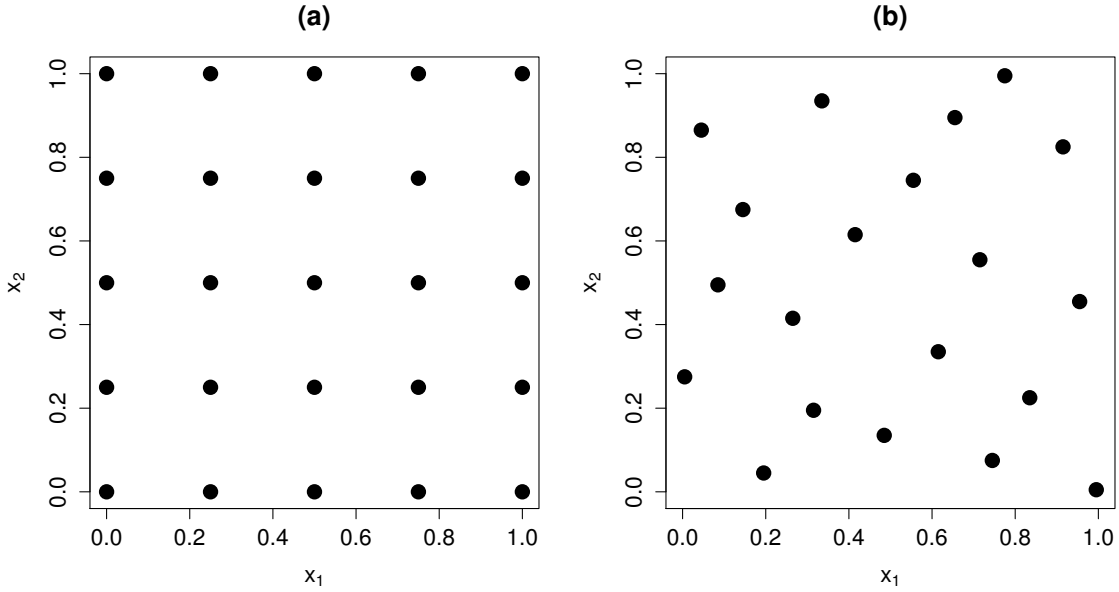


Figure 6: 25-run MED for the uniform distribution (a) $s = 2$ (b) $s = 0$.

Based on our proposal, Tuo and Lv (2016) was able to show that the limiting distribution of this design is $f(\boldsymbol{x})$ irrespective of the value of s. Now, for $s \to 0$, the criterion becomes

$$\max_{\boldsymbol{D}} \psi(\boldsymbol{D}) = \min_{i \neq j} f^{1/(2p)}(\boldsymbol{x}_i) f^{1/(2p)}(\boldsymbol{x}_j) \prod_{l=1}^{p} |x_{il} - x_{jl}|^{1/p}. \tag{2.2.6}$$

Now for $f(\boldsymbol{x}) = 1$, the criterion is to maximize $\prod_{l=1}^{p} |x_{il} - x_{jl}|^{1/p}$. The product measure ensures that no two points can have the same coordinate. Thus, the design will project onto n different points in each dimension, a property shared by the popular Latin hypercube designs. In fact, the criterion in (2.2.6) for $f(\boldsymbol{x}) = 1$ is a limiting case of the MaxPro design criterion proposed by Joseph, Gul and Ba (2015). The Latin hypercube and MaxPro designs have much better centered L_2 discrepancy measures (Fang et al., 2006) than factorial-type designs and thus, are expected to perform much better in high dimensions. We have not established any convergence rate for the integration errors of these new designs, but intuitively it should be comparable to $O((\log n)^{p-1}/n)$ rate of a QMC sample. Figure 6(b) shows the 25-point MED based

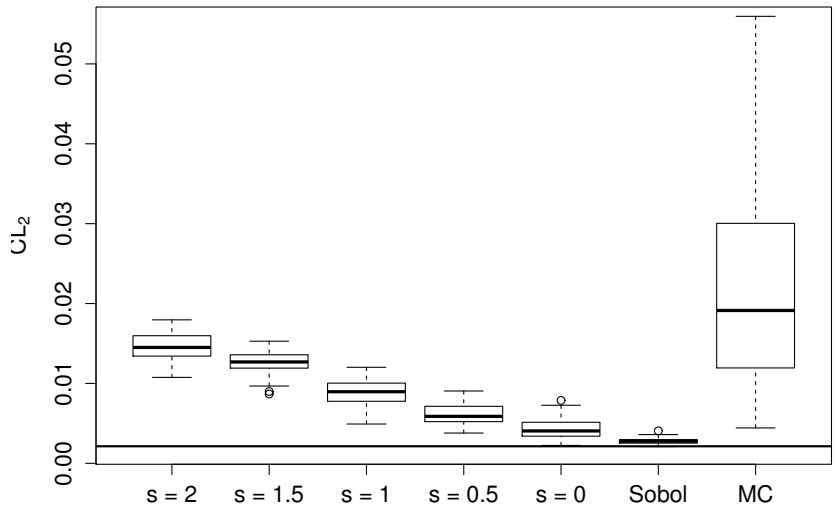


Figure 7: Centered L_2 discrepancies for MEDs with different s for $n = 20$.

on (2.2.6) with $s = 0$, which clearly has better projections than the original MED.

To further study the choice of s, we generated designs by the generalized MED criterion (2.2.5) with $s = 2, 1.5, 1, 0.5$, and criterion (2.2.6) ($s = 0$) using the greedy algorithm. Two settings were considered: $n = 20$, $p = 2$, and $n = 100$, $p = 2$. The centered L_2 discrepancies were computed, and the results with 100 replications randomized by the starting point are shown in Figures 7 and 8. The solid lines are the centered L_2 discrepancy for the uniform design. These designs are compared with the scrambled Sobol sequences (Owen, 1998), Monte Carlo (MC) random sampling, and the uniform designs generated using the software JMP. For both settings, we can clearly see that all the generalized MEDs are significantly better than random sampling. CL_2 decreases as s decreases. Thus, the generalized MED criterion with $s = 0$ can improve the integration performance compared to $s = 2$. For $n = 20$, the MED with $s = 0$ is worse than the scrambled Sobol sequences and the uniform design, but is still acceptable. However, when $n = 100$, the MED with $s = 0$ becomes competitive, which is better than the scrambled Sobol sequences and almost identical

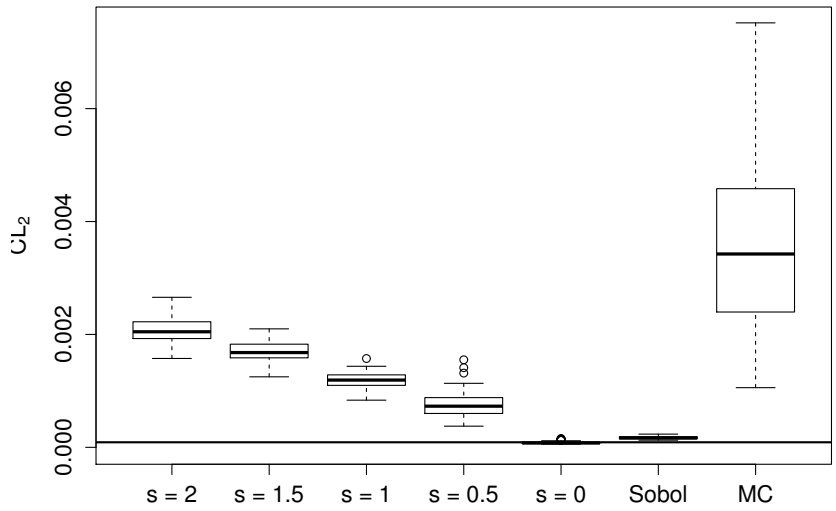


Figure 8: Centered L_2 discrepancies for MEDs with different s for $n = 100$.

to the uniform design. For the rest of the chapter, we have fixed $s = 0$.

2.2.3 Interpretation

Before developing an efficient construction algorithm for MED, we will give an intuition behind MED. The MED criterion with the Euclidean distance in (2.2.2) can also be written as

$$\max_{\boldsymbol{D}} \min_{i \neq j} \sqrt{f(\boldsymbol{x}_i) f(\boldsymbol{x}_j)} V_S(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

where $V_S(\boldsymbol{x}_i, \boldsymbol{x}_j) = \pi^{p/2}/\Gamma(p/2+1)\{d(\boldsymbol{x}_i, \boldsymbol{x}_j)/2\}^p$ is the volume of the sphere with center at $(\boldsymbol{x}_i + \boldsymbol{x}_j)/2$ and passing through the two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. See Figure 2 for an illustration. The term $\sqrt{f(\boldsymbol{x}_i) f(\boldsymbol{x}_j)}$ is the geometric mean of the density values at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Thus, $P_{ij}(\boldsymbol{D}) = \sqrt{f(\boldsymbol{x}_i) f(\boldsymbol{x}_j)} V_S(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is approximately the probability of $\boldsymbol{X}$ falling in the sphere. Let $i^* = arg\min_{j \neq i} P_{ij}(\boldsymbol{D})$. Then, the MED criterion can be written as

$$\max_{\boldsymbol{D}} \min_{i=1:n} P_{ii^*}(\boldsymbol{D}).$$

Now maximizing the minimum probability will tend to make all the probabilities $P_{ii^*}(\boldsymbol{D})$ for $i = 1, \ldots, n$ equal. Thus, roughly speaking, a MED tries to balance the probabilities among adjacent points of the design. This has similarities to the MCMC algorithms, which try to balance the transition probabilities.

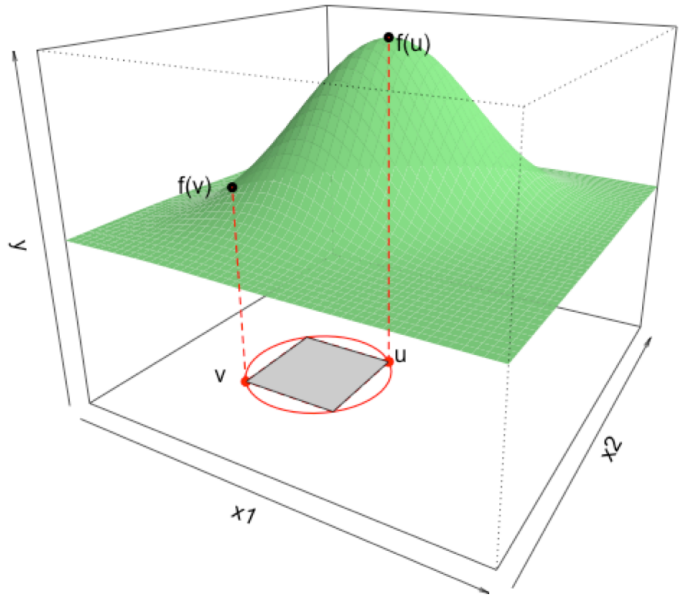


Figure 9: Probability-balancing interpretation of MED.

The MED criterion in (2.2.6) can also be given a similar interpretation. It can be written as

$$\max_{\boldsymbol{D}} \min_{i \neq j} \sqrt{f(\boldsymbol{x}_i) f(\boldsymbol{x}_j)} V_R(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

where $V_R(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{l=1}^{p} |x_{il} - x_{jl}|$ is the volume of the hyper-rectangle, which has $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ at the two opposite corners. See Figure 9 for an illustration. Thus, the same probability-balancing interpretation holds for this criterion as well, which can be obtained by replacing the hyper-sphere volume element with the hyper-rectangle volume element.

2.3 Simplex Construction Algorithm

In this section, we propose a construction algorithm for generating MEDs using few function evaluations compared to the greedy algorithm. Similar to Joseph, Dasgupta, Tuo and Wu (2015), a one-point-at-a-time fashion is used in the new algorithm. We consider the generalized MED criterion

$$\max_{\boldsymbol{x}} \min_{\boldsymbol{x}_i \in \boldsymbol{D}} f^{1/(2p)}(\boldsymbol{x}) f^{1/(2p)}(\boldsymbol{x}_i) \prod_{l=1}^{p} |x_l - x_{il}|^{1/p} \tag{2.3.1}$$

as the objective function, which is equivalent to

$$\max_{\boldsymbol{x}} \min_{\boldsymbol{x}_i \in \boldsymbol{D}} \frac{1}{2} \log f(\boldsymbol{x}) + \frac{1}{2} \log f(\boldsymbol{x}_i) + \sum_{l=1}^{p} \log |x_l - x_{il}|. \tag{2.3.2}$$

The main advantage of this criterion is that taking the logarithm of f and product measures can improve the numerical stability, which is very important since f can have very small values in high dimensional Bayesian problems.

The first step is to find all the modes of f by simplex search (Nelder and Mead, 1965). Then, MED points are generated sequentially also by simplex search. We call the proposed algorithm Minimum Energy Simplex Algorithm (MESA). The key feature of MESA is that all the points generated by simplex search and their evaluations on f are stored in a list. These points are considered to be candidate points for following iterations, which saves function evaluations and improves the performance of optimization.

2.3.1 Simplex Search

We first introduce how to construct and update simplexes that are used in MESA. Nelder-Mead method (Nelder and Mead, 1965) is a widely used nonlinear optimization method where the derivatives of the objective function are not known. In many Bayesian problems, the derivatives of the posterior distribution are not available. Among many derivative-free optimization methods, Nelder-Mead method is simple and efficient without any extra computation. The method constructs simplexes for

searching a local optimum. A p-dimensional simplex is a p-dimensional polytope, which is the convex hull of its $p+1$ vertices. With center $\boldsymbol{x}$ and radius r, throughout the chapter, simplex $S(\boldsymbol{x}, r)$ is constructed as follows. The $p+1$ vertices of the simplex are chosen as $\{\boldsymbol{x}, \boldsymbol{x} + r\boldsymbol{e}_1, \boldsymbol{x} + r\boldsymbol{e}_2, \cdots, \boldsymbol{x} + r\boldsymbol{e}_p\}$, where $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p\}$ are standard unit basis vectors. The objective function is evaluated at the $p + 1$ vertices, and their values are compared. We then update the worst vertex or shrink the simplex. The procedures are standard in Nelder-Mead method and are depicted in Figure 10. Details can be found in Nelder and Mead (1965). All the vertices with evaluations on f in the simplex are recorded into a candidate list $\boldsymbol{L}$.

2.3.2 Mode-Finding

As Joseph, Dasgupta, Tuo and Wu (2015) suggested, the first point is chosen as the mode of the posterior distribution. Sometimes the posterior distribution may have multiple modes. All the modes need to be identified first.

We start from multiple initial points. Space-filling designs, such as Maximin Latin hypercube designs (Morris and Mitchell, 1995) and Maximum projection Latin hypercube designs (Joseph, Gul and Ba, 2015) are good choices for the initial points. In this chapter, we adopt Maximum Projection Latin Hypercube Designs (MaxProLHDs) as the initial points. From each one point in the initial points, simplex search is applied to find a local maximum of f, which is equivalent to the optimum of (3.3.9) since there is no points in $\boldsymbol{D}$. All the vertices as well as their evaluations on f in the simplex are stored into $\boldsymbol{L}$.

2.3.3 Design Construction

After the simplex search in mode finding is finished, the first MED point is the one which has maximal $f(\boldsymbol{x})$ from $\boldsymbol{L}$. Suppose we have obtained n MED points $\boldsymbol{D}$, the

Algorithm 1 Simplex update

$\max \psi(\boldsymbol{x})$ is the objective function;
$\alpha \leftarrow 1, \gamma \leftarrow 2, \rho \leftarrow -\frac{1}{2}, \sigma \leftarrow \frac{1}{2}, \boldsymbol{L} \leftarrow \{\}$;
$Iteration \leftarrow 1$, $\boldsymbol{x}_1 \leftarrow d_i$, $\{\boldsymbol{x}_2, \cdots, \boldsymbol{x}_{p+1}\} \leftarrow d_i + \{e_1, \cdots, e_p\}$;
while $Iteration \leq MaxIteration$ **do**
    $Iteration \leftarrow Iteration + 1$;
    Order $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{p+1}\}$ according to the objective function values, such that $\psi(\boldsymbol{x}_1) \geq \psi(\boldsymbol{x}_2) \geq \cdots \geq \psi(\boldsymbol{x}_{p+1})$;
    $\boldsymbol{L} \leftarrow \boldsymbol{L} \cup \{(\boldsymbol{x}_1, f(\boldsymbol{x}_1)), \cdots, (\boldsymbol{x}_{p+1}, f(\boldsymbol{x}_{p+1}))\}$;
    $\boldsymbol{x}_c \leftarrow \frac{1}{p} \sum_{j=1}^{p} \boldsymbol{x}_j$;
    $\boldsymbol{x}_{new} \leftarrow \boldsymbol{x}_c + \alpha(\boldsymbol{x}_c - \boldsymbol{x}_{p+1})$, $\boldsymbol{L} \leftarrow \boldsymbol{L} \cup \{(\boldsymbol{x}_{new}, f(\boldsymbol{x}_{new}))\}$;
    if $\psi(\boldsymbol{x}_1) \geq \psi(\boldsymbol{x}_{new}) > \psi(\boldsymbol{x}_p)$ **then**
        $\boldsymbol{x}_{p+1} \leftarrow \boldsymbol{x}_{new}$
    else if $\psi(\boldsymbol{x}_{new}) > \psi(\boldsymbol{x}_1)$ **then**
        $\boldsymbol{x}_e \leftarrow \boldsymbol{x}_c + \gamma(\boldsymbol{x}_c - \boldsymbol{x}_{p+1})$, $\boldsymbol{L} \leftarrow \boldsymbol{L} \cup \{(\boldsymbol{x}_e, f(\boldsymbol{x}_e))\}$;
        if $\psi(\boldsymbol{x}_e) > \psi(\boldsymbol{x}_{new})$ **then**
            $\boldsymbol{x}_{p+1} \leftarrow \boldsymbol{x}_e$;
        else
            $\boldsymbol{x}_{p+1} \leftarrow \boldsymbol{x}_{new}$;
        end if
    else if $\psi(\boldsymbol{x}_{new}) \leq \psi(\boldsymbol{x}_p)$ **then**
        $\boldsymbol{x}_{con} \leftarrow \boldsymbol{x}_c + \rho(\boldsymbol{x}_c - \boldsymbol{x}_{p+1})$, $\boldsymbol{L} \leftarrow \boldsymbol{L} \cup \{(\boldsymbol{x}_{con}, f(\boldsymbol{x}_{con}))\}$;
        if $\psi(\boldsymbol{x}_{con}) > \psi(\boldsymbol{x}_{p+1})$ **then**
            $\boldsymbol{x}_{p+1} \leftarrow \boldsymbol{x}_{con}$;
        else
            $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_1 + \sigma(\boldsymbol{x}_i - \boldsymbol{x}_1), i = 2, 3, \cdots, p+1$;
            $\boldsymbol{L} \leftarrow \boldsymbol{L} \cup \{(\boldsymbol{x}_2, f(\boldsymbol{x}_2)), \cdots, (\boldsymbol{x}_{p+1}, f(\boldsymbol{x}_{p+1}))\}$
        end if
    end if
end while

Figure 10: Algorithm of simplex update

initial point for searching the $(n+1)$th MED point is selected from $\boldsymbol{L}$ given by

$$\boldsymbol{x}_{ini} = \arg\max_{\boldsymbol{x}\in\boldsymbol{L}} \min_{\boldsymbol{x}_i\in\boldsymbol{D}} \frac{1}{2}\log f(\boldsymbol{x}) + \frac{1}{2}\log f(\boldsymbol{x}_i) + \sum_{l=1}^{p} \log|x_l - x_{il}|. \tag{2.3.3}$$

Choose $\min\{n, p\}$ closest MED points to $\boldsymbol{x}_{ini}$ from $\boldsymbol{D}$, and compute the average Euclidean distance r. We then construct a simplex $S(\boldsymbol{x}_{ini}, r)$, which requires p new evaluations. Find the optimum of (3.3.9) by simplex search. The number of new evaluations for updating the simplex is limited to $(p+1)$, which is the same as the number of the vertices and has been found successful in simulations. Meanwhile, all the vertices with their evaluations on f are added to $\boldsymbol{L}$ as candidate points. For searching one MED point, the posterior distribution f is evaluated $(2p+1)$ times.

The choice of the initial point for each iteration is crucial for the performance of optimization. The greedy algorithm chooses the average of the last design point and one point from a space-filling design or low discrepancy sequence, which makes the initial point not close to the last MED point. However, the initial point still can be bad because not all the current MED points are considered. In MESA, the initial point is the candidate point that has the best MED criterion value of (2.3.3) based on all the current MED points. The initial point itself is already a good choice as the next MED point. Simplex search then starts from this point to find a better one. Moreover, since the initial point is selected from the candidate list, no more evaluations are needed.

Note that a limit on the number of updates of the simplex is set in this stage. This early termination of optimization can affect the performance of one single point. But since we are constructing a design which consists of multiple points, the overall performance is much less affected. Simulation results show that this compromise is reasonable, given that it can save a large number of evaluations.

2.4 Local Approximation

In this section, we introduce a local approximation method for MESA to reduce function evaluations. The idea is as follows. Since MED points can represent the posterior distribution, based on the current MED points, we can fit a local surrogate model around the initial point to approximate the posterior distribution. The following evaluations are on the surrogate model instead of the true posterior distribution, which saves considerable evaluations. A similar idea can be found in Joseph, Dasgupta, Tuo and Wu (2015) for sequentially constructing MEDs for expensive functions. However, they fit a global Gaussian process model on all the current MED points, which may lead to a poor performance on some local regions. Whereas we construct a local model in a neighborhood of the initial point and find the optimal MED point locally.

Suppose we have n MED points $\boldsymbol{D}$. The initial point $\boldsymbol{x}_{ini}$ is firstly found by (2.3.3). A local surrogate model around $\boldsymbol{x}_{ini}$ is then built. We fit a quadratic regression model on $\log f(\boldsymbol{x})$ given by

$$g^{(n)}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{j=1}^{p} \beta_j x_j^2, \tag{2.4.1}$$

where the unknown coefficients $\beta_0, \beta_i, \beta_j$ are estimated using ordinary least squares

$$\min_{\beta_0,\beta_i,\beta_j} \sum_{\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_{ini})} \left\{\log f(\boldsymbol{x}_i) - g^{(n)}(\boldsymbol{x}_i)\right\}^2, \tag{2.4.2}$$

where $\mathcal{N}(\boldsymbol{x}_{ini})$ is a set of points around $\boldsymbol{x}_{ini}$ that includes: $\min\{n, 2p\}$ closest MED points to $\boldsymbol{x}_{ini}$ and all the candidate points that are inside the sphere with center $\boldsymbol{x}_{ini}$ and radius $d(\boldsymbol{x}_{ini}, \boldsymbol{x}_m)$, where $\boldsymbol{x}_m$ is the farthest point among all the closest MED points. Denote the fitted model by $\widehat{g}^{(n)}(\boldsymbol{x})$.

Starting from $\boldsymbol{x}_{ini}$, we always run optimization on the local surrogate model $\widehat{g}^{(n)}(\boldsymbol{x})$. The optimal point $\boldsymbol{x}^*$ can be found with the objective function

$$\max_{\boldsymbol{x} \in \mathcal{N}(\boldsymbol{x}_{ini})} \min_{\boldsymbol{x}_i \in \boldsymbol{D}} \frac{1}{2}\widehat{g}^{(n)}(\boldsymbol{x}) + \frac{1}{2}\widehat{g}^{(n)}(\boldsymbol{x}_i) + \sum_{l=1}^{p} \log|x_l - x_{il}|. \tag{2.4.3}$$

There is no reason to still use simplex search on $\widehat{g}^{(n)}$, since evaluations on $\widehat{g}^{(n)}$ are assumed to be cheap. Because the gradient information of $\widehat{g}^{(n)}$ is easily available, any gradient-based optimization algorithms can be used. Here we adopt quasi-newton algorithm. Then, we evaluate $\boldsymbol{x}^*$ on the true posterior distribution f and add it into $\boldsymbol{L}$. Since all the evaluations in optimization are made on $\widehat{g}^{(n)}$, for searching one MED point, the number of evaluations on f is only one.

Because $\boldsymbol{x}^*$ is found based on the surrogate model $\widehat{g}^{(n)}$, it is possible that $\boldsymbol{x}^*$ is not good when evaluated on f. We compare the criterion values (3.3.9) for $\boldsymbol{x}_{ini}$ and $\boldsymbol{x}^*$. If $\boldsymbol{x}^*$ has a better criterion value, it is selected as the $(n+1)$th MED point $\boldsymbol{x}_{n+1}$. Otherwise, if $\boldsymbol{x}_{ini}$ is better, we add $\boldsymbol{x}^*$ into $\boldsymbol{L}$ and fit $\widehat{g}^{(n)}(\boldsymbol{x})$ again. This procedure can iterate many times until we find a point that is better than $\boldsymbol{x}_{ini}$. However, considering that $\boldsymbol{x}_{ini}$ is already the best point in all the current candidate points, we iterate only once. Then, if $\boldsymbol{x}_{ini}$ is still better than the new $\boldsymbol{x}^*$, take $\boldsymbol{x}_{n+1} = \boldsymbol{x}_{ini}$.

The mission of local approximation is to generate more MED points at almost no cost. In local approximation, only one point $\boldsymbol{x}^*$ is evaluated on f and is added into $\boldsymbol{L}$ for searching one MED point. We lose most of the ability to update the candidate points. It is important to choose when to start local approximation. When the local approximation starts, MED points can roughly represent the true posterior distribution, so that the surrogate model is able to provide a good approximation. Based on our experience, $0.5n$ is a good choice to start local approximation, where n is the total number of MED points.

As the iteration goes, the number of points in the candidate list increases. Note that in each iteration, the initial point is selected by computing the energy between the current MED points and every point in the candidate list. This computation also becomes heavier. By applying local approximation, the number of candidate points is controlled, and the computation can be reduced as well.

2.5 Examples

2.5.1 Multivariate Normal Distributions

We begin with a standard example. Joseph, Dasgupta, Tuo and Wu (2015) considered multivariate normal distributions with mean vector zero and variance-covariance matrix $\mathbf{\Sigma}$ where the (ij)th entry $\sigma_{ij} = 0.9^{|i-j|}$. The dimensions p were from two to ten. For each one case, $n = 25p$ MED points were generated by MESA and the greedy algorithm. Comparisons were made on sampling performance and computational time.

To quantify the discrepancy between the empirical distribution of the MED points and the true distribution, we transformed the MED points to uniform distribution $[0, 1]^p$, and then computed the center L_2 discrepancy measure. The results are shown in Figure 11. As p increases, MESA significantly outperforms the greedy algorithm. When $p = 10$, the discrepancy of the points by MESA is only one tenth of the greedy algorithm. See Figure 12 for an example in five dimensions. We can see that the histograms of the points by MESA are much closer to the true distribution. The reasons of the improvement are two-fold. First, MESA uses the generalized MED criterion (2.2.6), where $s = 0$ has better sampling performance than the original $s = 2$. Second, for each iteration, MESA has a better initial point that is the best choice given the current MED points; while the greedy algorithm does not take into account of the current MED points when choosing the initial point. The choice of the initial point becomes more important when the dimension is high.

Figure 13 shows the CPU time in a laptop with a 2.6 GHz processor. We can see that MESA takes more time than the greedy algorithm. When $p = 10$, MESA took about $14,147$ seconds, which is still acceptable compared to $1,280$ seconds for the greedy algorithm. Besides the difference of programming languages, the extra computation in each iteration in MESA is the computation of the energy between the current MED points and each one point in the candidate list. The candidate list

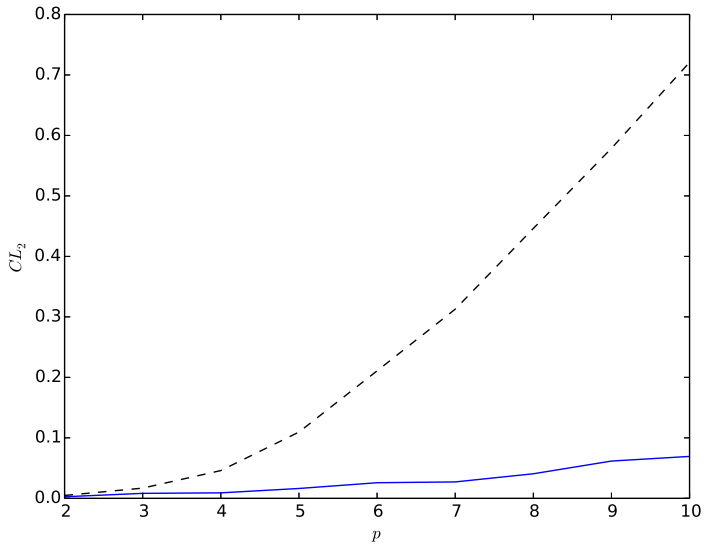


Figure 11: Centered L_2 discrepancies of designs generated by MESA (blue solid line) and the greedy algorithm (black dashed line) for multivariate normal distributions.

becomes larger as we have more MED points, which takes more time. One solution is to use local approximation after $0.5n$ points proposed in Section 2.4.

The number of function evaluations is of more interest. Compared to the greedy algorithm, MESA dramatically decreases the number of required evaluations in all dimensions, which is shown in Figure 14. In ten dimensional cases, MESA requires about $10,950$ evaluations, whereas the posterior distribution is evaluated $1,472,797$ times in the greedy algorithm.

2.5.2 Banana Example

The second example is a two-dimensional banana shaped distribution (Haario et al., 2001). The target distribution

$$f(\boldsymbol{x}) \propto \exp\left\{-\frac{1}{2}\frac{x_1^2}{100} - \frac{1}{2}(x_2 + 0.03x_1^2 - 3)^2\right\}.$$

See Figure 15 for an illustration of the target distribution. Because of the strongly nonlinear shape, it is very difficult to have enough samples on the two tails. We

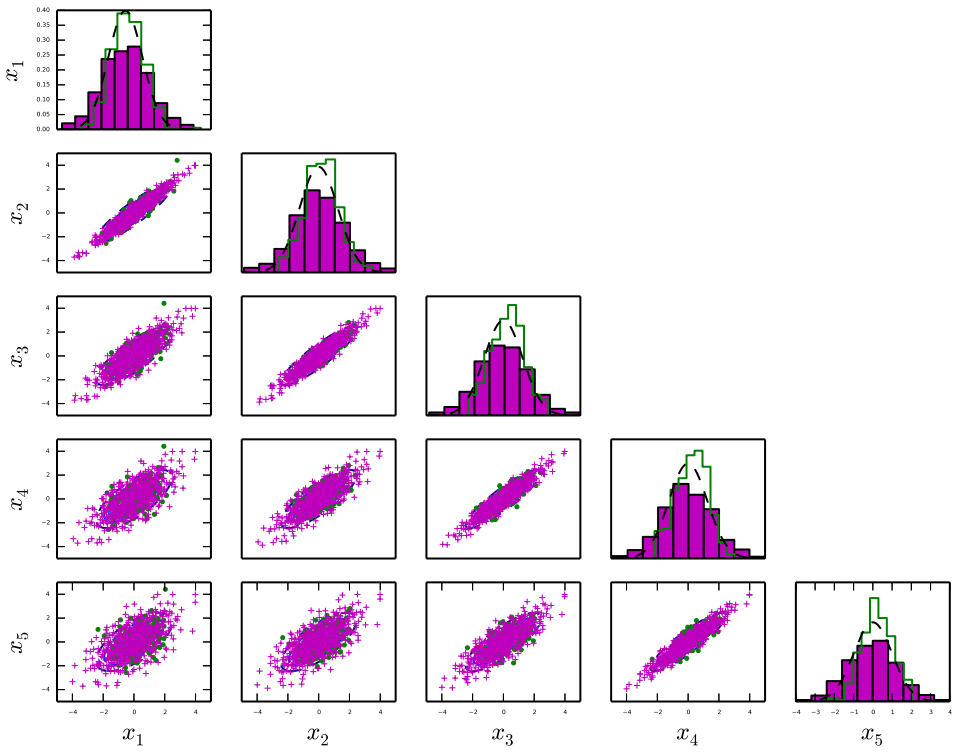


Figure 12: Histograms of the MED points generated by MESA (green dots) and the greedy algorithm (purple plus signs) for multivariate normal distributions.

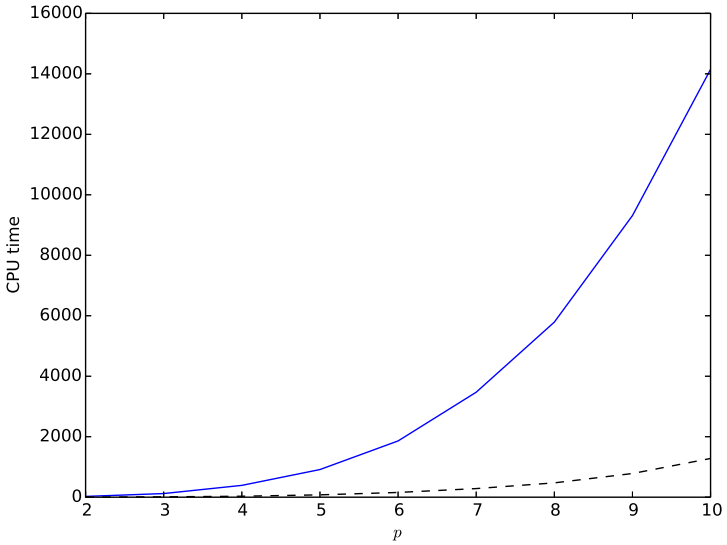


Figure 13: CPU times (in seconds) of MESA (blue solid line) and the greedy algorithm (black dashed line) for multivariate normal distributions.

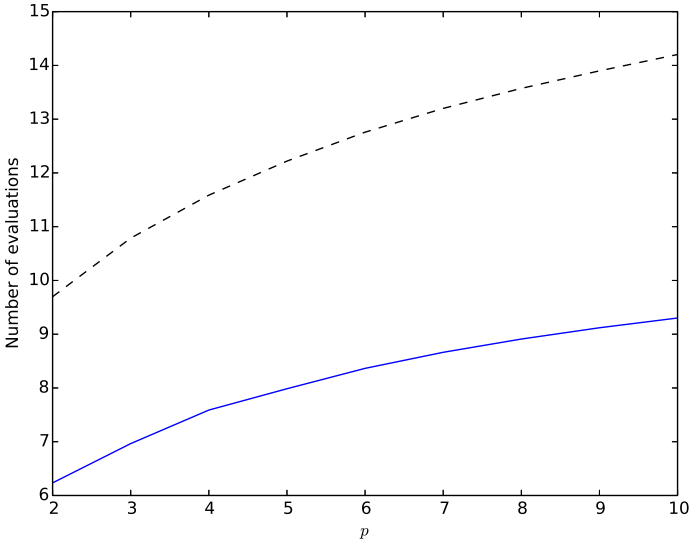


Figure 14: Number of function evaluations (on log-scale) of MESA (blue solid line) and the greedy algorithm (black dashed line) for multivariate normal distributions.

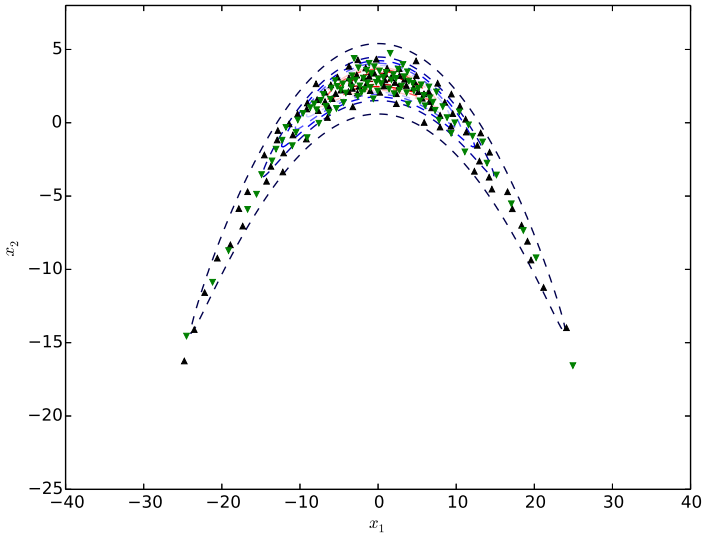


Figure 15: MED points generated by MESA (green inverted triangles) and the greedy algorithm (black triangles) for the banana example.

generated $n = 50p$ MED points by MESA, the greedy algorithm, and the combination of MESA and local approximation, which generates first $25p$ MED points by MESA, and uses local approximation for the next $25p$ MED points.

We replicated 100 times of simulations. In each replication, for MESA, the initial MaxProLHD in mode finding was different, whereas the initial point for each iteration in the greedy algorithm was selected from a different MaxProLHD. The results are shown in Table 5. Standard deviations are given in parentheses. The first column is the Mean Squared Errors (MSEs) of the mean. We can see that MESA is a bit worse than the greedy algorithm. The number of function evaluations are given in the third column of Table 5. MESA requires much fewer evaluations than the greedy algorithm. Considering the savings of evaluations, the compromise on the performance is acceptable. Another thing that we want to point out is that the combination of MESA and local approximation only increases the error (in variance) a little compared to the full MESA. At the same time, the number of evaluations is much smaller. So after enough points capture the overall shape of the target

distribution, building a surrogate model and evaluating on it is a good alternative to further save function evaluations without much loss of performance, rather than continuing evaluating the true function.

Table 5: Results for the banana example.

	MSE of mean	MSE of "> 90%"	Number of evaluations
MESA	0.6170(0.1386)	0.0051(0.0013)	977
MESA+local	0.5483(0.2244)	0.0051(0.0013)	597
Greedy	0.3917(0.1189)	0.0074(0.0015)	98,635

To investigate the performance in low probability regions, we counted the number of samples that hit the confidence region outside 90%. The percentage was computed by the number of samples in the region over the total number of samples. The second column of Table 5 compares the MSEs between the percentage in the MEDs and the true percentage, which is 10%. Both MESA and the combination of MESA and local approximation are smaller than the greedy algorithm. It is also illustrated in Figure 15, where we can see that MESA has more points on the tails. In mode finding, the initial design points are well spread out. They are all stored in the candidate list and can be selected as the initial points for each iteration. Thus, we have a much larger probability to find extreme points than the greedy algorithm.

2.6 Conclusions

In this chapter, we have improved two limitations on MED. One is its efficiency in integration. The integration error rate using MED points is low. We have defined a generalized distance and have used it to generalize the MED criterion. With a proper choice of the tuning parameter, the efficiency of the generalized MEDs is greatly improved.

The other limitation is the construction algorithm. An MED is constructed by a one-point-at-a-time greedy algorithm, where a global optimization is required in each iteration. The function evaluations are too many to make MED competitive

to MC/MCMC methods. We have developed Minimum Energy Simplex Algorithm (MESA) for constructing MEDs with much less function evaluations. In each iteration, MESA constructs simplexes to search the optimal MED point while keeping all the evaluated points as a candidate list for finding good starting points in the next iteration. MESA is shown to have better performance using much less function evaluations.

After all the MED points have been generated by MESA, there may exist some better points in the candidate list, which are not selected because of the one-point-at-a-time fashion. The simulated annealing algorithm has been used to select MED points from finite MCMC samples (Gu and Joseph, 2016), which can be applied for further improving the performance of MESA. The basic idea is that in each iteration, the worst MED point in the current design is replaced, with a probability, by another possibly better point in the candidate list $\boldsymbol{L}$. Details can be found in Gu and Joseph (2016). One attractive feature is that no more evaluations are needed. However, it will cost more computational time. We will study the potential improvement in future research.

CHAPTER III

ROBUST PARAMETER DESIGN USING COMPUTER EXPERIMENTS

3.1 Introduction

Robust parameter design is a cost-efficient technique for quality improvement. Originally proposed by Taguchi (1987), the technique has been widely adopted in industries for system (product or process) optimization. The core idea is to first divide the factors in the system into two groups: control factors and noise factors. Control factors are those factors in the system that can be cost-effectively controlled. On the other hand, noise factors are those factors which are either impossible or too expensive to control. Since the noise factors are uncontrollable, they introduce variability in the output causing quality problems. Robust parameter design is a technique to find a setting of the control factors (also known as parameter design) that will make the system robust or insensitive to the noise factors. Thus, under a robust parameter design, the output becomes less affected by the noise variability even when the noise factors are left uncontrolled. This is why the approach using robust parameter design is less costly than other quality improvement techniques which try to directly control the noise factors in the system.

The key to a successful robust parameter design is in identifying important control-by-noise interactions of the system. Only when such interactions exist we can use the control factors to reduce the sensitivity of the noise factors. These interactions are usually unknown in practice and their existence need to be investigated through experimentation. Thus designing good experiments is a crucial step in robustness studies. Many efficient experimental design techniques are proposed in the literature

such as cross arrays Taguchi (1987) and single arrays (Shoemaker et al., 1991; Wu and Zhu, 2003; Kang et al., 2011). A thorough discussion of these techniques can be found in the books by Wu and Hamada (2009) or Myers et al. (2016).

The aforementioned experimental design techniques are mainly proposed for physical experimentation. Recently, computer experiments have become very common in industries. That is, if a computer model is available that can simulate the physical system, the experiments then can be performed in computers instead of the physical system. This can bring in tremendous cost savings because direct experimentation with the real physical system is always more expensive than investing on some computer time. However, there are several aspects of computer experiments that necessitate the use of a different experimental design technique or philosophy compared to those of physical experiments (Sacks et al., 1989). Since most computer models are deterministic in nature, randomization and replications are not needed. Fractional factorial and orthogonal array-based design techniques that are prevalent in physical experiments lead to replications when projected onto subspace of factors and thus are unsuitable for computer experiments. Split-plot designs that are considered to be useful in robustness studies (Bingham and Sitter, 2003) become unnecessary as run orders and restrictions on randomization will not affect the computer model outputs. This lead to the development of space-filling designs in computer experiments.

The existing work on robust parameter design using space-filling designs do not make any distinction between control and noise factors. A distinction is made only at the analysis stage (Welch et al., 1992; Apley et al., 2006; Bates et al., 2006; Chen et al., 2006; Tan, 2015). Sequential designs that directly attempt to find robust settings of control factors using expected improvement-type algorithms are proposed in the literature (Williams et al., 2000; Lehman et al., 2004), but we are not aware of any work on space-filling designs. It is important to develop space-filling designs that

distinguish control and noise factors because their distributional properties are entirely different. Noise factors are commonly assumed to follow a normal distribution, whereas control factors are assumed to follow a uniform distribution. Noise factors are intrinsically random and can vary over time and space. Different from them, control factors remain fixed once their levels are chosen. A uniform distribution is imposed on the control factors only to represent our indifference on the choice of level given the range of possible values for each control factor. Thus, unlike the control factors, most of the "action" in the noise factor space takes place in the center than at the tails. Therefore, space-filling designs that uniformily spread out points in the experimental region are not adequate for robust parameter design experiments. In this chapter we propose a space-filling design that puts more points in regions where probability mass for the noise distribution is higher and thus obtain better fitted models where it matters the most. However, nonuniform space-filling designs create challenges in model fitting using kriging or Gaussian process models (Santner et al., 2003). Since the design points are not equally-spaced, stationary covariance functions can lead to numerical instability in computation and tend to perform poorly in prediction. In this chapter, we propose a simple but efficient nonstationary Gaussian process that takes into account of the experimental design structure to solve this potentially difficult problem.

This chapter is organized as follows. In Section 3.2, we motivate the problem using a real computer experiment on packaging from the Procter & Gamble company. In Section 3.3, we propose the new experimental design method and in Section 3.4, we propose the new modeling method. The performance of the proposed methods is compared with the existing methods using some simulated examples in Section 3.5. In Section 3.6, we revisit the example of the computer experiment on packaging and illustrate the application of the proposed methods and we conclude the chapter with some remarks and future research directions in Section 3.7.

3.2 Motivating Example: Packing Experiment

The presence of noise factors is a very common occurrence with computer experiments in industry. Noise factors can include variation in either material properties or part dimensions. Other common noise factors can involve variation in product or package use by the consumer. Finally, noise factors can involve environmental variation like temperature and humidity, as well as process factors that are difficult to control.

The specific example we will use to motivate the proposed methods will involve a packing line at Procter & Gamble (P&G). The example has been slightly modified for simplicity. A computer simulation was developed for one critical transformation (or part) of an packing line. A computer experiment with nine input factors was performed. Six input variables are process variables which are defined as control factors, given that in practice, they remain fixed once they are chosen. In addition, there are other three variables, which are material properties. They are defined as noise factors given that there is variability in material properties of the packaging component. The output response from the computer simulation is a measurement of how well the packing is.

3.3 Experimental Design

3.3.1 Formulation of the Experimental Design Problem

Let $\boldsymbol{x} = (x_1, \ldots, x_p)'$ be the random vector for p control factors and $\boldsymbol{z} = (z_1, \ldots, z_q)'$ the random vector for q external noise factors. The term "external" will be explained later in the section. We will assume that $\boldsymbol{x} \in \mathcal{X} = [0,1]^p$ and $\boldsymbol{z} \in \mathcal{Z} = [0,1]^q$ after some re-scaling. The response y is a function of both control and noise factors given by $y = g(\boldsymbol{x}, \boldsymbol{z})$. Depending on the type of characteristic such as smaller-the-better, larger-the-better, or nominal-the-best, we can impose a loss function on y. Let $L(y)$ be such a loss function (Joseph, 2004). Then, the objective of robust parameter design is to find the setting of control factors that minimizes the expected loss, where the

expectation is taken with respect to the distribution of noise factors. Let $f(\boldsymbol{z})$ denote the probability density function of $\boldsymbol{z}$ with support in $\mathcal{Z}$. Then, the robust parameter design can be obtained by

$$\min_{\boldsymbol{x}} \int_{\mathcal{Z}} L\{g(\boldsymbol{x}, \boldsymbol{z})\} f(\boldsymbol{z}) d\boldsymbol{z}.$$

Since the function $g(\cdot, \cdot)$ is available only as a computer code, an experiment will be conducted to estimate it. Let $\boldsymbol{D} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ be the experimental design with n runs, where $\boldsymbol{u}_i = (\boldsymbol{x}_i', \boldsymbol{z}_i')'$, and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})'$, $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iq})'$ are the settings of the control factors and the noise factors for the ith run, respectively, for $i = 1, \ldots, n$. Let $\widehat{g}(\boldsymbol{x}, \boldsymbol{z})$ be the estimated response function from the experiment (also known as surrogate model, metamodel, or emulator). Then, the optimization can be simplified as

$$\min_{\boldsymbol{x}} \int_{\mathcal{Z}} L\{\widehat{g}(\boldsymbol{x}, \boldsymbol{z})\} f(\boldsymbol{z}) d\boldsymbol{z}. \tag{3.3.1}$$

It is also possible to incorporate the uncertainties in the estimation of $g(\cdot, \cdot)$ in the optimization as in Apley and Kim (2011) and Tan and Wu (2012). The problem we are trying solve is how to design the experiment $\boldsymbol{D}$ so that we can accurately estimate the solution to the optimization problem in (3.3.1).

A careful examination of (3.3.1) reveals an important insight on the experimental design problem. We need an accurate $g(\cdot, \cdot)$ only in the regions of $\boldsymbol{z}$ where $f(\boldsymbol{z})$ is large. In other words, if $f(\boldsymbol{z})$ is small in some regions, the inaccuracies in the estimation of $g(\cdot, \cdot)$ in those regions will not affect the robust parameter design. This makes the experimental design problem for robustness different from that of a usual computer experiment. Let us now see how to design such an experiment optimally.

Suppose, after the experiment, we fit a Gaussian Process (GP) model

$$y(\boldsymbol{x}, \boldsymbol{z}) \sim GP(\mu, \sigma^2 R(\cdot)), \tag{3.3.2}$$

where μ and σ^2 are the unknown mean and variance parameters, and $R(\cdot)$ is the correlation function. A commonly used correlation function is the Gaussian correlation

function given by

$$R(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{z}_i - \boldsymbol{z}_j) = \exp\{-\sum_{l=1}^{p} \theta_l^x (x_{il} - x_{jl})^2 - \sum_{m=1}^{q} \theta_m^z (z_{im} - z_{jm})^2\},$$

where $\boldsymbol{\theta}^x = (\theta_1^x, \ldots, \theta_p^x)'$ and $\boldsymbol{\theta}^z = (\theta_1^z, \ldots, \theta_q^z)'$ are the unknown correlation parameters of the control and noise factors. The mean squared error of prediction is given by

$$MSE(\boldsymbol{x}, \boldsymbol{z}) = 1 - r(\boldsymbol{x}, \boldsymbol{z})' \boldsymbol{R}^{-1} r(\boldsymbol{x}, \boldsymbol{z}) + \frac{(1 - r(\boldsymbol{x}, \boldsymbol{z})' \boldsymbol{R}^{-1} \mathbf{1})^2}{\mathbf{1}' \boldsymbol{R}^{-1} \mathbf{1}},$$

where $r(\boldsymbol{x}, \boldsymbol{z})$ is an $n \times 1$ vector with ith element $R(\boldsymbol{x} - \boldsymbol{x}_i, \boldsymbol{z} - \boldsymbol{z}_i)$, $\boldsymbol{R}$ is an $n \times n$ matrix with ijth element $R(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{z}_i - \boldsymbol{z}_j)$, and $\mathbf{1}$ is a vector of 1's having length n. We want to find $\boldsymbol{D}$ such that $MSE(\boldsymbol{x}, \boldsymbol{z})$ is small. However, since $MSE(\boldsymbol{x}, \boldsymbol{z})$ is a function of $\boldsymbol{x}$ and $\boldsymbol{z}$, it is not possible to find such a design over the entire experimental region. Instead, a feasible approach is to minimize the average of $MSE(\boldsymbol{x}, \boldsymbol{z})$:

$$\min_{\boldsymbol{D}} IMSE = \int_{\mathcal{X}} \int_{\mathcal{Z}} MSE(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{z}) d\boldsymbol{z} d\boldsymbol{x}. \tag{3.3.3}$$

This design criterion is the same as the integrated mean squared error (IMSE) criterion in the literature (Sacks et al., 1989; Santner et al., 2003) except that we use the density of $\boldsymbol{z}$ as a weight function. This is quite a natural modification of the IMSE criterion and agrees with our intuition that we should give more weights for regions where $f(\boldsymbol{z})$ is large.

A major drawback of the IMSE criterion is that it is a function of the unknown correlation parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}^{x\prime}, \boldsymbol{\theta}^{z\prime})'$. One approach to overcome this drawback is to adopt a Bayesian approach. Let $p(\boldsymbol{\theta})$ be the prior distribution of $\boldsymbol{\theta}$. Then, our design criterion becomes

$$\min_{\boldsymbol{D}} BIMSE = \int \int_{\mathcal{X}} \int_{\mathcal{Z}} MSE(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{z}) d\boldsymbol{z} d\boldsymbol{x} p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{3.3.4}$$

This criterion also has some drawbacks. First, it is extremely expensive to compute because analytical integration is not possible, especially for the integral on $\boldsymbol{\theta}$. Second,

the criterion is based on the assumed stationarity of the GP model in (3.3.2), which may not hold true in practice. Because of these reasons, space-filling designs are more commonly used in computer experiments. We will also do the same in the next section. However, the development of the optimal design criterion of this section is useful in the sense that it gives a solid formulation of the underlying design problem and can serve as an evaluation criterion for the other proposed designs.

Before proceeding further, we need to clarify a few issues related to robust parameter design. There are factors which have uncontrollable variability around their nominal values. This is called *internal* noise. Examples include, part-to-part variability within their tolerances and process parameter variability around their targets. On the other hand, *external* noise factors are completely uncontrollable including their nominal values. Examples of external noise factors include user conditions, incoming raw material properties, and so on. Different from external noise factors, internal noise factors need not be varied in the experiment because they can be easily introduced at the modeling stage (Kang and Joseph, 2009). Another aspect that should be clarified is about the existence of adjustment factors (Joseph, 2007). When the response is nominal-the-best type, one can almost always find adjustment factors which can be used for adjusting the mean to target. In such cases, we can ignore the mean model and focus completely on modeling the variance. This means that we can change our focus of estimating $g(\cdot, \cdot)$ accurately to estimating the derivatives of $g(\cdot, \cdot)$ with respect to $\boldsymbol{z}$ accurately (Kang and Joseph, 2009). We leave this problem as a topic for future research.

3.3.2 Space-Filling Designs

Space-filling designs aim at filling the experimental region evenly with as few gaps as possible. These designs are robust to modeling choices and thus, are widely used

as designs for computer experiments. See Joseph (2016) for a recent review of space-filling designs.

A popular choice for space-filling design is the Maximin Latin Hypercube Design (MmLHD) proposed by Morris and Mitchell (1995). In an MmLHD, all the factors take n levels $\{.5/n, 1.5/n, \ldots, (n-.5)/n\}$ and the design points are obtained by maximizing the minimum distance among the points

$$\max_{\boldsymbol{D}\in\mathcal{L}} \min_{i\neq j} \left\{ \sum_{l=1}^{p} (x_{il} - x_{jl})^2 + \sum_{m=1}^{q} (z_{im} - z_{jm})^2 \right\}^{\frac{p+q}{2}}, \tag{3.3.5}$$

where $\mathcal{L}$ denotes the class of Latin hypercube designs. The power $(p+q)/2$ in (3.3.5) has no effect on the result and is used only to facilitate the discussion below.

Let us now see how to modify the space-filling designs to suit the requirements for robustness studies. First, as alluded in the introduction, we need more points in the high probability region than in the low probability region. This can be easily achieved by using inverse probability transform. Assume that the noise factors are independent. Denote the settings of the lth and mth control and noise factors by $\boldsymbol{X}_l = (x_{1l}, \ldots, x_{nl})'$ and $\boldsymbol{Z}_m = (z_{1m}, \ldots, z_{nm})'$, respectively. Then, $\boldsymbol{D}$ can be written as $\boldsymbol{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_q\}$. Let $F_m(\cdot)$ be the distribution function of z_m, $m = 1, \ldots, q$. It is well-known that if z_m follows a uniform distribution, $F_m^{-1}(z_m)$ has the distribution $F_m(\cdot)$. Thus, the desired design can be obtained as

$$\boldsymbol{D}^* = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p, F_1^{-1}(\boldsymbol{Z}_1), \ldots, F_q^{-1}(\boldsymbol{Z}_q)\},$$

where $\boldsymbol{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_q\}$ is an MmLHD. Note that although we have relied on the uniformity of the points to apply the inverse probability transform, the space-filling design does not have to be a uniform design. The transformed design $\boldsymbol{D}^*$ can be viewed as a space-filling design in the new transformed space.

There is another way to obtain a space-filling design for a given probability distribution. The Minimum Energy Design (MED) proposed by Joseph, Dasgupta, Tuo

and Wu (2015) is given by

$$\max_{\boldsymbol{D}} \min_{i \neq j} \sqrt{f(\boldsymbol{z}_i) f(\boldsymbol{z}_j)} \left\{ \sum_{l=1}^{p} (x_{il} - x_{jl})^2 + \sum_{m=1}^{q} (z_{im} - z_{jm})^2 \right\}^{\frac{p+q}{2}}, \tag{3.3.6}$$

which can be viewed as representative points of the target distribution $f(\cdot) = \prod_{m=1}^{q} f_m(\cdot)$, where $f_m(\cdot)$ is the probability density of z_m. The asymptotic convergence of the limiting distribution of MED is recently proved by Tuo and Lv (2016). As discussed in Wang et al. (2016), the objective function in (3.3.6) can be interpreted as proportional to the probability of a spherical region defined by the points $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$. They proposed an extension of MED which uses a hyper-rectangular region instead of the spherical region and is given by

$$\max_{\boldsymbol{D}} \min_{i \neq j} \sqrt{f(\boldsymbol{z}_i) f(\boldsymbol{z}_j)} \prod_{l=1}^{p} |x_{il} - x_{jl}| \prod_{m=1}^{q} |z_{im} - z_{jm}|. \tag{3.3.7}$$

The validity of this criterion can be rigorously shown using a general result obtained by Tuo and Lv (2016). Although, both (3.3.6) and (3.3.7) can asymptotically produce the desired probability distribution, there is a major difference in terms of their space-filling property. The criterion in (3.3.7) is closely related to the maximum projection criterion (Joseph, Gul and Ba, 2015) and will produce designs with excellent projection properties, whereas the use of the criterion in (3.3.6) will lead to factorial-type designs which have poor projection properties. We will exploit this difference in the space-fillingness to our benefit.

As discussed in Section 3.1, control-by-noise interactions are important for robustness studies. Because the total information from the experiment is fixed, we can improve the estimation of control-by-noise interactions only if we can sacrifice the estimation of other interactions or higher-order effects. Thus, by sacrificing the higher-order effects and interactions in the noise factor space, we can hope to improve the estimation of those effects in the control factor space as well as between the control and noise factors. Translating into design language, we can sacrifice the projections

in the noise factor space and hope to get better projections in the control factor space and between control and noise factors. This suggests using a hyper-rectangular region in the control factor space and a spherical region in the noise factor space, which leads to the MED criterion:

$$\max_{\boldsymbol{D}} \min_{i \neq j} \prod_{l=1}^{p} |x_{il} - x_{jl}| \sqrt{f(\boldsymbol{z}_i) f(\boldsymbol{z}_j)} \left\{ \sum_{m=1}^{q} (z_{im} - z_{jm})^2 \right\}^{\frac{q}{2}}. \tag{3.3.8}$$

To summarize, in this section, we have proposed two ways of obtaining space-filling designs for robustness studies: using the inverse probability transform of a space-filling design and the other using the MED in (3.3.8). However, it is not clear which of these designs is better. The latter seems to capture the desirable properties better than the former at least in terms of estimating the control-by-noise interaction effects, but this needs further investigation. We will evaluate them using the BIMSE criterion in (3.3.4). Before that we need to explain how to construct the designs.

3.3.3 Optimal Design Algorithm

For the simplicity of notations, let

$$E(\boldsymbol{u}_i, \boldsymbol{u}_j) = \prod_{l=1}^{p} |x_{il} - x_{jl}| \sqrt{f(\boldsymbol{z}_i) f(\boldsymbol{z}_j)} \left\{ \sum_{m=1}^{q} (z_{im} - z_{jm})^2 \right\}^{\frac{q}{2}}.$$

As in Morris and Mitchell (1995), we use the criterion

$$\min_{\boldsymbol{D}} \psi(\boldsymbol{D}) = \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{E^k(\boldsymbol{u}_i, \boldsymbol{u}_j)} \right\}^{1/k}, \tag{3.3.9}$$

for searching for optimal design. The criterion in (3.3.9) approximates the criterion in (3.3.8) for large k and has the added benefit that it tends to minimize the pairs of points with the largest energy (also known as the index of the design).

We perform the optimization of $\psi(\boldsymbol{D})$ as follows. First we generate a space-filling design and obtain the initial design for optimization through inverse probability transform as discussed in the previous section. Specifically, we use the MaxProLHD

(Joseph, Gul and Ba, 2015) because it can give a nearly optimal design in the control factor space. Denote the initial design by $\boldsymbol{D} = (\boldsymbol{D}_x^{(0)}, \boldsymbol{D}_z^{(0)})$. To simplify the optimization, we alternately optimize $\boldsymbol{D}_x$ and $\boldsymbol{D}_z$. That is, suppose that we have finished $h-1$ iterations and the current optimal design is $(\boldsymbol{D}_x^{(h-1)}, \boldsymbol{D}_z^{(h-1)})$. We first fix $\boldsymbol{D}_z^{(h-1)}$ and obtain

$$\boldsymbol{D}_x^{(h)} = \arg\min_{\boldsymbol{D}_x} \psi(\boldsymbol{D}_x, \boldsymbol{D}_z^{(h-1)}), \tag{3.3.10}$$

and then fix $\boldsymbol{D}_x^{(h)}$ and obtain

$$\boldsymbol{D}_z^{(h)} = \arg\min_{\boldsymbol{D}_z} \psi(\boldsymbol{D}_x^{(h)}, \boldsymbol{D}_z). \tag{3.3.11}$$

These iterations are continued until convergence.

We use continuous optimization algorithms in both (3.3.10) and (3.3.11). The optimizations can be made much faster using gradient information, which can be analytically calculated. The gradients are given by

$$\frac{\partial \psi^k(\boldsymbol{D})}{\partial x_{rs}} = k \sum_{i \neq r} \left\{ \frac{1}{E^k(\boldsymbol{u}_i, \boldsymbol{u}_r)} \frac{1}{x_{is} - x_{rs}} \right\}, \tag{3.3.12}$$

for $r = 1, \ldots, n$, $s = 1, \ldots, p$, and

$$\frac{\partial \psi^k(\boldsymbol{D})}{\partial z_{rt}} = kq \sum_{i \neq r} \left\{ \frac{1}{E^k(\boldsymbol{u}_i, \boldsymbol{u}_r)} \frac{z_{it} - z_{rt}}{\sum_{m=1}^{q} (z_{im} - z_{rm})^2} \right\} - \frac{k}{2} f(\boldsymbol{z}_r)^{-1} \frac{\partial f}{\partial \boldsymbol{z}_{rt}}(\boldsymbol{z}_{rt}) \sum_{i \neq r} \frac{1}{E^k(\boldsymbol{u}_i, \boldsymbol{u}_r)}, \tag{3.3.13}$$

for $r = 1, \ldots, n$, $t = 1, \ldots, q$.

3.3.4 Design Evaluation

We compare the designs using the BIMSE criterion in (3.3.4). Three designs were compared: an existing space-filling design (we use the MmLHD in the chapter), the proposed transformed space-filling design denoted as T(MmLHD), and the MED. Two settings were considered: (i) $n = 40, p = 2, q = 2$, and (ii) $n = 80, p = 4, q = 4$.

Assume that the noise factors follow a normal distribution. Note that we have assumed them to be in $[0, 1]^q$ after re-scaling. If the standard deviation of the original distribution (before re-scaling) is known, this is easy to do. However, in many

practical problems, the practitioner may only be able to specify them in some intervals, where even the limits are somewhat vaguely defined. Therefore, we specify the normal distribution as follows. Let $[a, b]$ be the range of z^m in original scale. First it can be re-scaled to $[0, 1]$ by $(z^m - a)/(b - a)$. Now the normal distribution on $[0, 1]$ is defined as with mean 0.5 and standard deviation

$$\sigma = \frac{1 - 1/n}{\Phi^{-1}(1 - .5/n) - \Phi^{-1}(.5/n)}. \tag{3.3.14}$$

This specification ensures that the range of T(MmLHD) is $[0.5/n, 1-0.5/n]^{p+q}$, which allows for a fair comparison with the MmLHD. Throughout the chapter, the optimal MmLHDs were downloaded from https://spacefillingdesigns.nl/. MEDs were directly generated from the distributions by using the optimization algorithm described in the previous section.

We need to specify a prior distribution for $\boldsymbol{\theta}^x = (\theta_1^x, \ldots, \theta_p^x)'$. From our experience with different products and processes, the response is usually a smooth function over the noise factors. But the function can exhibit complex nonlinear relationships with the control factors. With this in mind, we let $\theta_l^x \sim Exp(1)$ for $l = 1, \ldots, p$, and $\theta_m^z \sim Exp(10)$ for $m = 1, \ldots, q$. In other words, the prior mean of θ_m^z is ten times as smaller than that of θ_l^x, which makes the realizations from the GP much smoother over the noise factors than over the control factors.

The integration in (3.3.3) is performed using quasi Monte-Carlo methods. We used 10,000 Sobol points, where the columns related to the noise factors are transformed using a truncated normal distribution with mean 0.5, standard deviation σ, and limits $[0, 1]$. To perform the integration over $\boldsymbol{\theta}$ in (3.3.4), we used a $10(p + q)$-run MaxProLHD with inverse probability transform using the specified exponential distributions. A boxplot of the $10(p + q)$ IMSE values are shown in Figures 16 and 17. The Monte Carlo average values (approximate BIMSEs) are given in Table 6. We can see that for both the settings, T(MmLHD) and MED have better performances than the existing approach using MmLHD. In terms of the mean, T(MmLHD) and

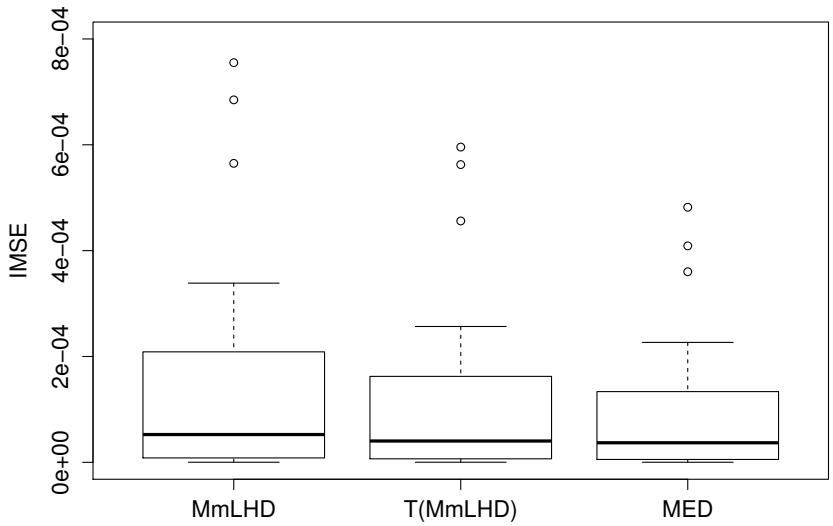


Figure 16: IMSEs for MmLHD, T(MmLHD) and MED for $n = 40, p = 2$, and $q = 2$ for different realizations of $\boldsymbol{\theta}$.

MED improve the performance by 22% and 43% for $n = 40$, and 19% and 21% for $n = 80$, respectively. Between the two proposed designs, MED is slightly better. Thus, according to the Bayesian IMSE criterion in (3.3.4), MED is the preferred choice for robust parameter design experiments.

Table 6: Bayesian IMSEs for MmLHD, T(MmLHD) and MED.

	MmLHD	T(MmLHD)	MED
$n = 40$	2.13×10^{-4}	1.66×10^{-4}	1.21×10^{-4}
$n = 80$	6.61×10^{-3}	5.37×10^{-3}	5.20×10^{-3}

3.4 Modeling

GP models in (3.3.2) are the standard choice for modeling in computer experiments (Sacks et al., 1989). However, there is a problem in using it in robustness studies. It can be explained as follows. Let $\boldsymbol{y} = (y_1, \ldots, y_n)'$ be the data from the experiment.

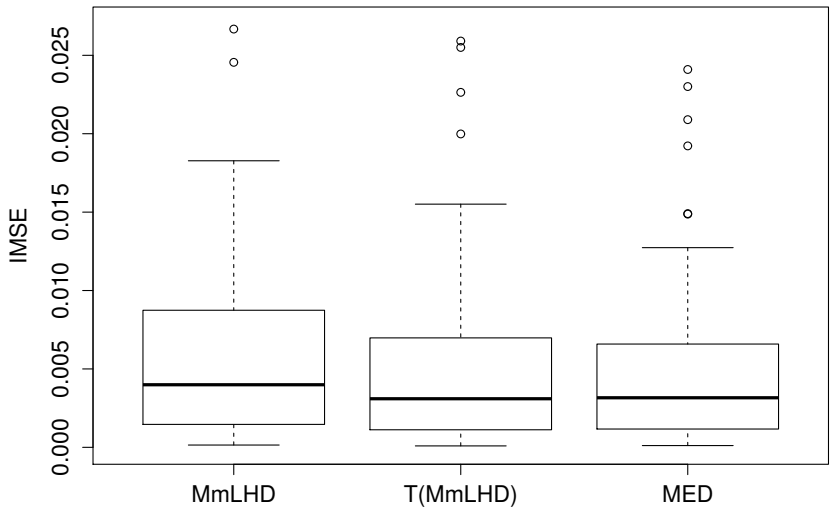


Figure 17: IMSEs for MmLHD, T(MmLHD) and MED for $n = 80, p = 4$, and $q = 4$ for different realizations of $\boldsymbol{\theta}$.

Then, the posterior mean of $y(\boldsymbol{x}, \boldsymbol{z})$ is given by

$$\widehat{y}(\boldsymbol{x}, \boldsymbol{z}) = \widehat{\mu} + \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{z})' \boldsymbol{R}^{-1}(\boldsymbol{y} - \widehat{\mu}\mathbf{1}),$$

where $\widehat{\mu} = \mathbf{1}'\boldsymbol{R}^{-1}\boldsymbol{y}/\mathbf{1}'\boldsymbol{R}^{-1}\mathbf{1}$. Let $\boldsymbol{c} = \boldsymbol{R}^{-1}(\boldsymbol{y} - \widehat{\mu}\mathbf{1})$. Then, the predictor can be written in the basis function expansion form

$$\begin{aligned} \widehat{y}(\boldsymbol{x}, \boldsymbol{z}) &= \widehat{\mu} + \sum_{i=1}^{n} c_i R(\boldsymbol{x} - \boldsymbol{x}_i, \boldsymbol{z} - \boldsymbol{z}_i) \\ &= \widehat{\mu} + \sum_{i=1}^{n} c_i \exp\left\{ - \sum_{l=1}^{p} \theta_l^x (x_l - x_{il})^2 \right\} \exp\left\{ - \sum_{m=1}^{q} \theta_m^z (z_m - z_{im})^2 \right\}. \end{aligned}$$

The correlation parameters are constants and do not vary with $\boldsymbol{x}$ or $\boldsymbol{z}$. This is acceptable in the control factor space because the points are evenly spread out in that space, whereas for noise factors we have more points in the high probability regions than in the low probability regions. Using a constant correlation parameter $\boldsymbol{\theta}^z$ does not make sense in this case and can lead to numerical instability and poor predictions. We would want $\boldsymbol{\theta}^z$ to be small when the points are closer and large when

the points are farther. However, changing $\boldsymbol{\theta}^z$ with respect to different z^m is not an easy proposal because such correlation functions are not well-defined (that is, they may not be positive definite). In this section, we propose an alternative fix to this problem.

3.4.1 Model Formulation and Prediction

We define two basis functions for $\boldsymbol{z}$: $\exp\{-\sum_{m=1}^{q}\theta_m^z(z_m - z_{im})^2\}$ and $\exp\{-\sum_{m=1}^{q}\alpha_m(z_m - z_{im})^2\}$, where $\alpha_m \geq \theta_m^z$. The first basis function has larger length-scale (or smaller θ_m^z) which is good for modeling in the low-probability regions, whereas the second basis function has smaller length-scale (or larger α_m) which is suitable for modeling in the high probability regions. In the GP modeling framework, these can be defined using a combination of two GPs, which is an idea proposed by Ba and Joseph (2012) and Harari and Steinberg (2014). See also the earlier work by Booker (2000). Borrowing the notations used in Ba and Joseph (2012), our proposed model is

$$\begin{aligned} y(\boldsymbol{x},\boldsymbol{z}) &= \mu + \delta_g(\boldsymbol{x},\boldsymbol{z}) + w(\boldsymbol{z})\delta_l(\boldsymbol{x},\boldsymbol{z}), \\ \delta_g(\boldsymbol{x},\boldsymbol{z}) &\sim GP(0,\tau^2 g(\cdot)), \\ \delta_l(\boldsymbol{x},\boldsymbol{z}) &\sim GP(0,\sigma^2 l(\cdot)), \end{aligned} \tag{3.4.1}$$

where μ is the constant mean, $\delta_g(\boldsymbol{x},\boldsymbol{z})$ is the GP for the global trend with variance parameter τ^2 and correlation function $g(\cdot)$, $\delta_l(\boldsymbol{x},\boldsymbol{z})$ is the GP for the local adjustments on noise factors with variance parameter σ^2 and correlation function $l(\cdot)$, and $w(\boldsymbol{z})$ is the weight function. This model is equivalent to

$$Y(\boldsymbol{x},\boldsymbol{z}) \sim GP(\mu, \tau^2 g(\cdot) + \sigma^2 w^2(\boldsymbol{z}) l(\cdot)). \tag{3.4.2}$$

We choose the weight function to be

$$w(\boldsymbol{z}) = \left(\frac{f(\boldsymbol{z})}{f_{max}}\right)^{\gamma}, \tag{3.4.3}$$

where $f_{max} = \max f(\boldsymbol{z})$ and γ is an unknown parameter in $(0, 1)$. The correlation functions are given by

$$\begin{aligned} g(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{z}_i - \boldsymbol{z}_j) &= \exp\left\{ -\sum_{l=1}^{p} \theta_l^x (x_{il} - x_{jl})^2 - \sum_{m=1}^{q} \theta_m^z (z_{im} - z_{jm})^2 \right\}, \\ l(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{z}_i - \boldsymbol{z}_j) &= \exp\left\{ -\sum_{l=1}^{p} \theta_l^x (x_{il} - x_{jl})^2 - \sum_{m=1}^{q} \alpha_m (z_{im} - z_{jm})^2 \right\}, \end{aligned} \tag{3.4.4}$$

where $\alpha_m \geq \theta_m^z$, for $m = 1, \ldots, q$. Note that if there exist no noise factors, the proposed model will degenerate to an ordinary GP model.

The proposed model is a much simplified version of the composite Gaussian process (CGP) model of Ba and Joseph (2012). The weight function is pre-defined up to a constant, whereas Ba and Joseph estimate the function nonparametrically from the data. Moreover, the correlation parameters for the control factors are the same for both the GPs which makes the parameter estimation simpler than that in CGP models.

The best linear unbiased predictor of the model (3.4.1) is derived as follows. Denote $\boldsymbol{Q} = \boldsymbol{G} + \lambda \boldsymbol{W}\boldsymbol{L}\boldsymbol{W}$, where $\boldsymbol{G}$ and $\boldsymbol{L}$ are two $n \times n$ correlation matrices with the (ij)th entry $g(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{z}_i - \boldsymbol{z}_j)$ and $l(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{z}_i - \boldsymbol{z}_j)$, respectively, and $\boldsymbol{W} = diag\{w(\boldsymbol{z}_1), \ldots, w(\boldsymbol{z}_n)\}$. Similar to CGP models, we have that

$$\widehat{\mu} = (\mathbf{1}'\boldsymbol{Q}^{-1}\mathbf{1})^{-1}\mathbf{1}'\boldsymbol{Q}^{-1}\boldsymbol{y}, \tag{3.4.5}$$

and

$$\widehat{y}(\boldsymbol{x}, \boldsymbol{z}) = \widehat{\mu} + \{\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}) + \lambda w(\boldsymbol{z})\boldsymbol{W}\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{z})\}' \boldsymbol{Q}^{-1}(\boldsymbol{y} - \widehat{\mu}\mathbf{1}), \tag{3.4.6}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)'$, $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}) = (g(\boldsymbol{x} - \boldsymbol{x}_1, \boldsymbol{z} - \boldsymbol{z}_1), \ldots, g(\boldsymbol{x} - \boldsymbol{x}_n, \boldsymbol{z} - \boldsymbol{z}_n))'$, $\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{z}) = (l(\boldsymbol{x} - \boldsymbol{x}_1, \boldsymbol{z} - \boldsymbol{z}_1), \ldots, l(\boldsymbol{x} - \boldsymbol{x}_n, \boldsymbol{z} - \boldsymbol{z}_n))'$, and $\lambda = \sigma^2/\tau^2$ is the ratio of variances. We refer the proposed model to New GP model thereafter.

3.4.2 Parameter Estimation

We adopt maximum likelihood to estimate the unknown parameters

$$(\mu, \tau, \sigma, \gamma, \theta_1^x, \ldots, \theta_p^x, \theta_1^z, \ldots, \theta_q^z, \alpha_1, \ldots, \alpha_q).$$

The log-likelihood function (up to an additive constant) of the model (3.4.1) is

$$\begin{aligned} & l(\mu, \tau, \sigma, \gamma, \theta_1^x, \ldots, \theta_p^x, \theta_1^z, \ldots, \theta_q^z, \alpha_1, \ldots, \alpha_q) \\ = & -\frac{1}{2}\left\{\log(\det(\tau^2 \boldsymbol{G} + \sigma^2 \boldsymbol{W}\boldsymbol{L}\boldsymbol{W})) + (\boldsymbol{y} - \mu\mathbf{1})'(\tau^2 \boldsymbol{G} + \sigma^2 \boldsymbol{W}\boldsymbol{L}\boldsymbol{W})^{-1}(\boldsymbol{y} - \mu\mathbf{1})\right\}, \end{aligned} \tag{3.4.7}$$

which is equivalent to

$$\begin{aligned} & l(\mu, \tau, \lambda, \gamma, \theta_1^x, \ldots, \theta_p^x, \theta_1^z, \ldots, \theta_q^z, \alpha_1, \ldots, \alpha_q) \\ = & -\frac{1}{2}\left\{n\log(\tau^2) + \log(\det(\boldsymbol{Q})) + (\boldsymbol{y} - \mu\mathbf{1})'\boldsymbol{Q}^{-1}(\boldsymbol{y} - \mu\mathbf{1})/\tau^2\right\}. \end{aligned} \tag{3.4.8}$$

The maximum likelihood estimators of μ and τ^2 can be derived from (3.4.8), which is given by

$$\widehat{\mu} = (\mathbf{1}'\boldsymbol{Q}^{-1}\mathbf{1})^{-1}\mathbf{1}'\boldsymbol{Q}^{-1}\boldsymbol{y}, \quad \widehat{\tau}^2 = (\boldsymbol{y} - \widehat{\mu}\mathbf{1})'\boldsymbol{Q}^{-1}(\boldsymbol{y} - \widehat{\mu}\mathbf{1})/n. \tag{3.4.9}$$

Then, by substituting them into (3.4.8), we have the log profile likelihood

$$l(\lambda, \gamma, \theta_1^x, \ldots, \theta_p^x, \theta_1^z, \ldots, \theta_q^z, \alpha_1, \ldots, \alpha_q) = -n\log(\widehat{\tau}^2) - \log(\det(\boldsymbol{Q})). \tag{3.4.10}$$

The maximum likelihood estimators for the unknown parameters are obtained by maximizing (3.4.10).

The ranges for the unknown parameters in the optimization are taken as follows. We set $\lambda \in [0, 1]$ because it is expected that $g(\cdot)$ always dominates the model rather than $l(\cdot)$. The correlation parameters for control factors θ_l^x are positive as in ordinary GP models. For noise factors, we assure $\alpha_m \geq \theta_m^z$ by setting $\alpha_m = \theta_m^z + \kappa_m$, $m = 1, \ldots, q$, where $\kappa_m \in [0, \infty)$. The upper bounds of θ_m^z, α_m, are decided by following the same rule in CGP models, that is, $\alpha_m = \log 100/d_{avg}^2$, where d_{avg}^2 is the average distance. In the weight function $w(\boldsymbol{z})$, we set $\gamma \in [0, 1]$ as described before.

In total we have $p+2q+2$ unknown parameters to be optimized. Compared to the ordinary GP model, where the number of unknown parameters in the optimization is p, it is acceptable since the number of noise factors q is usually relatively small in real examples.

3.5 Simulations

In this section, we study the performances of both the proposed designs and the proposed model by simulated examples.

3.5.1 Gaussian Process Simulations

Sample paths on $[0,1]^{p+q}$ from $GP(\mathbf{0}, \sigma^2 R(\cdot))$ were simulated, where $\sigma^2 = 25$, $R(\cdot)$ has the same form (3) with correlation parameters $\theta_l^x = 20$ for all $l = 1, \ldots, p$ and $\theta_m^z = 5$ for all $m = 1, \ldots, q$. The control factors were uniformly distributed on $[0,1]^p$, and the noise factors were from the normal distribution with mean 0.5 and variance given by (3.3.14). GP realizations were generated on the sites of both design points and test points. The design points were used to fit a model, and the test points were then used to calculate the prediction errors. The test points were $N = 100(p+q)$ points from Sobol' sequences after inverse probability transform on noise factors. We fitted ordinary GP models on MmLHD, T(MmLHD), and MED, and fitted New GP models on the last two. Three simulation settings were (i) $n = 40, p = 2, q = 2$, (ii) $n = 60, p = 3, q = 3$, and (iii) $n = 80, p = 4, q = 4$. The simulations were replicated 100 times.

The box plots of the absolute prediction errors for $n = 60, p = 3, q = 3$ are given in Figure 18. The absolute prediction errors were given by $|\widehat{y}(\boldsymbol{x}_i, \boldsymbol{z}_i) - y(\boldsymbol{u}_i)|$, $i = 1, \ldots, N$, where $\boldsymbol{u}_i$ is the ith test point. The results for the other two settings are similar and are hence omitted. Two groups of comparisons can be seen from Figure 18. First, using ordinary GP models, both T(MmLHD) and MED perform better than MmLHD, which shows improvements because of the usage of the new designs. Besides, by fitting New GP models on T(MmLHD) and MED, the performances are further improved. The comparisons between ordinary GP models and New GP models illustrate that New GP models work well on the proposed non-uniform designs.

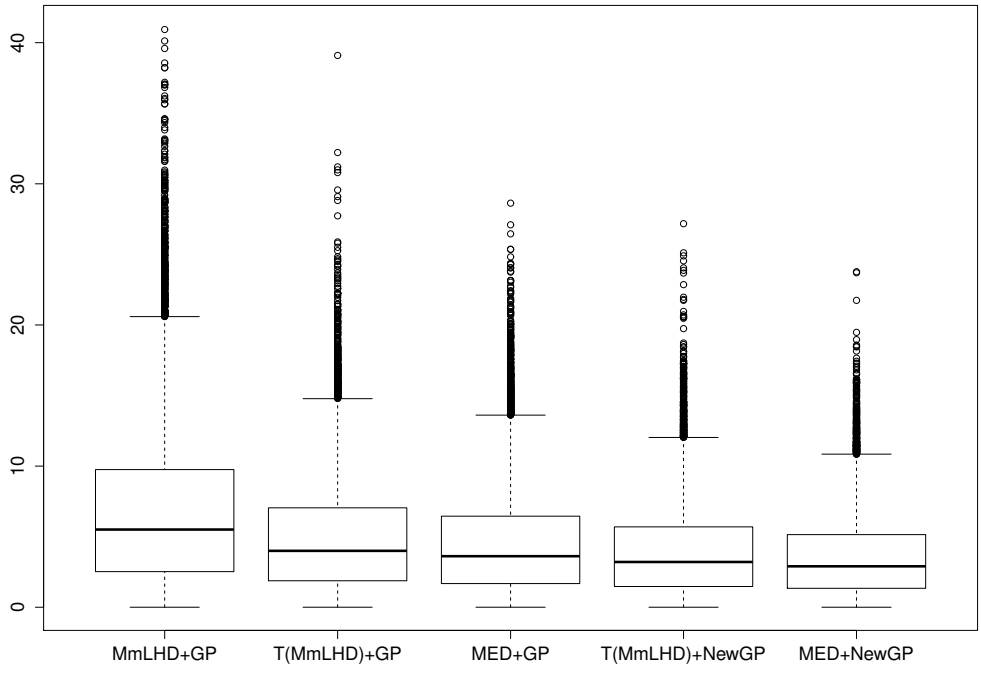


Figure 18: Absolute prediction errors for the GP simulations example.

3.5.2 Ishigami Function

Let us consider the Ishigami function (Ishigami and Homma, 1990). The form is given by

$$g(\boldsymbol{u}) = \sin x_1 + A \sin^2 x_2 + B z_1^4 \sin x_1 + 0 z_2, \tag{3.5.1}$$

where $A = 5$, $B = 1$, and $x_1, x_2, z_1, z_2 \in [-\pi, \pi]$ are the control factors and the noise factors scaled to $[0, 1]^{p+q}$, and follow the same distributions as in the GP simulations example.

The design setting was $n = 40$, $p = 2$, and $q = 2$. Ordinary GP models were fitted on MmLHD, T(MmLHD), and MED, and New GP models on the last two. The test points were $N = 100(p + q)$ points from Sobol' sequences after inverse probability transform on noise factors.

Table 7: RMSPEs for Ishigami function.

RMSPE	MmLHD	T(MmLHD)	MED
GP	6.53	5.57	5.41
NewGP	6.44	5.09	4.97

The results of Root Mean Squared Prediction Errors (RMSPE) are shown in Table 7. The RMSPE was calculated by $\left\{ \frac{1}{N} \sum_{i=1}^{N} (\widehat{y}(\boldsymbol{u}_i) - g(\boldsymbol{u}_i))^2 \right\}^{1/2}$, where $\boldsymbol{u}_i$ is the ith test point. From the results we can see that T(MmLHD) and MED improve the RMSPE by 14.7% and 17.2%, and New GP models further improve by another 7.4% and 6.7%, respectively. Based on all the simulation results, we recommend MED as the design for robustness studies.

3.6 Packing Example

In this section, the performance of the proposed design and modeling method on the packing line example from P&G is tested.

Computer simulations were run for P&G packing lines. In the computer simulations, nine input factors were used. The first six factors were control factors denoted

by $x_1, x_2, x_3, x_4, x_5, x_6$. The distributions of the control factors were assumed to be uniform distributions with corresponding ranges. The last three factors were noise factors denoted by z_1, z_2, z_3. They followed normal distributions, where the length of the range was six standard errors for each noise factor, respectively. The inputs were then scaled to a unit cube.

To study the performance of both design and modeling methods, we compared three different settings. Engineers usually fit an ordinary GP model on MmLHD, which was considered as the benchmark. It was compared with fitting an ordinary GP model on MED (which only changes the design), and fitting a New GP model on MED (which further changes the modeling method). The total number of data points was $n = 10p = 90$.

Figures 19 and 20 show the MmLHD and MED used in the simulations. We can see that for the noise space of the MED, the design points follow normal distributions, and more points concentrated in the center.

Since the computer simulations are relatively expensive, engineers usually do not run extra experiments for testing. The prediction performance is measured by Leave-One-Out Cross Validation (LOOCV) prediction errors defined by

$$y(\boldsymbol{u}_i) - \widehat{y}_{(-i)}(\boldsymbol{u}_i), \tag{3.6.1}$$

where $\widehat{y}_{(-i)}$ is the surrogate model fitted with all the data points except $(\boldsymbol{u}_i, y_i)$, for $i = 1, \ldots, n$.

We first plot the true response $y(\boldsymbol{u}_i)$ vs. the LOOCV predicted response $\widehat{y}_{(-i)}(\boldsymbol{u}_i)$, which are shown in Figures 21, 22 and 23. A method has good prediction performance and is desirable if points are located close to the 45 degree line. In Figures 21 and 22, we can clearly see that MED+GP is much better than MmLHD+GP . More points are close to the 45 degree line, which shows the strength of MED over MmLHD. The modeling methods are compared in Figures 22 and 23. Note that points in MED+NewGP and in MED+GP have the same $y(\boldsymbol{u}_i)$, but most points in MED+NewGP are closer

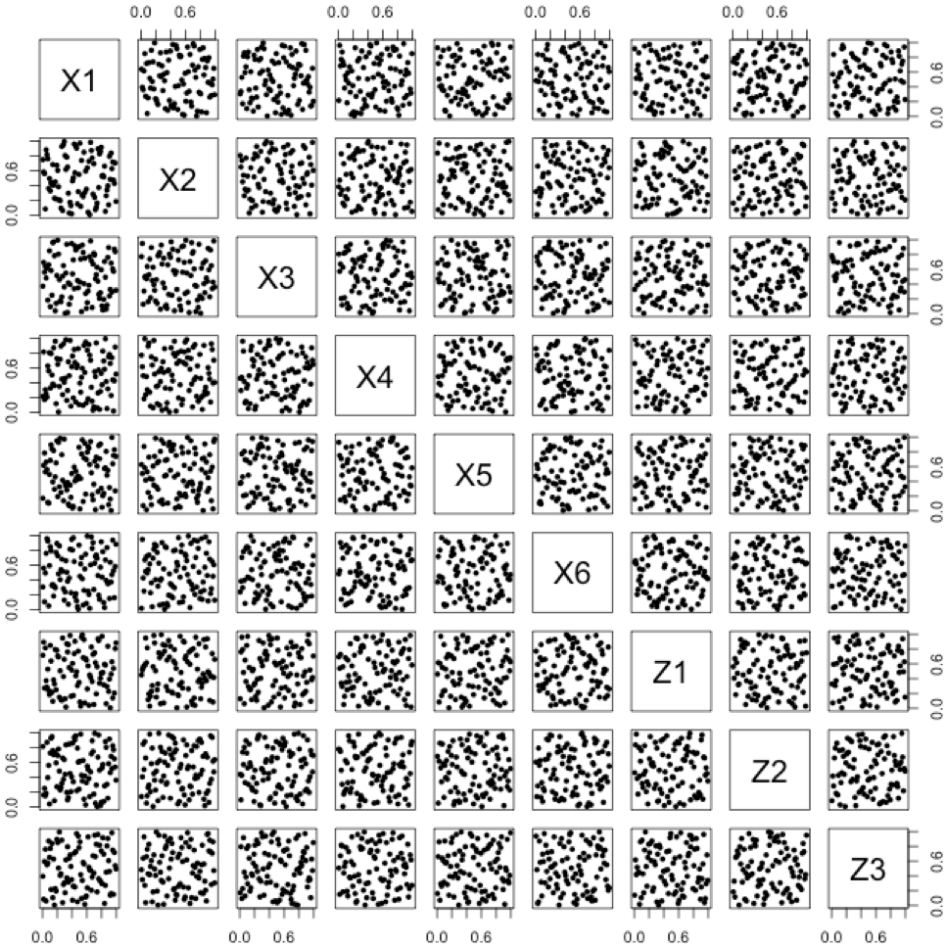


Figure 19: MmLHD for P&G packing example.

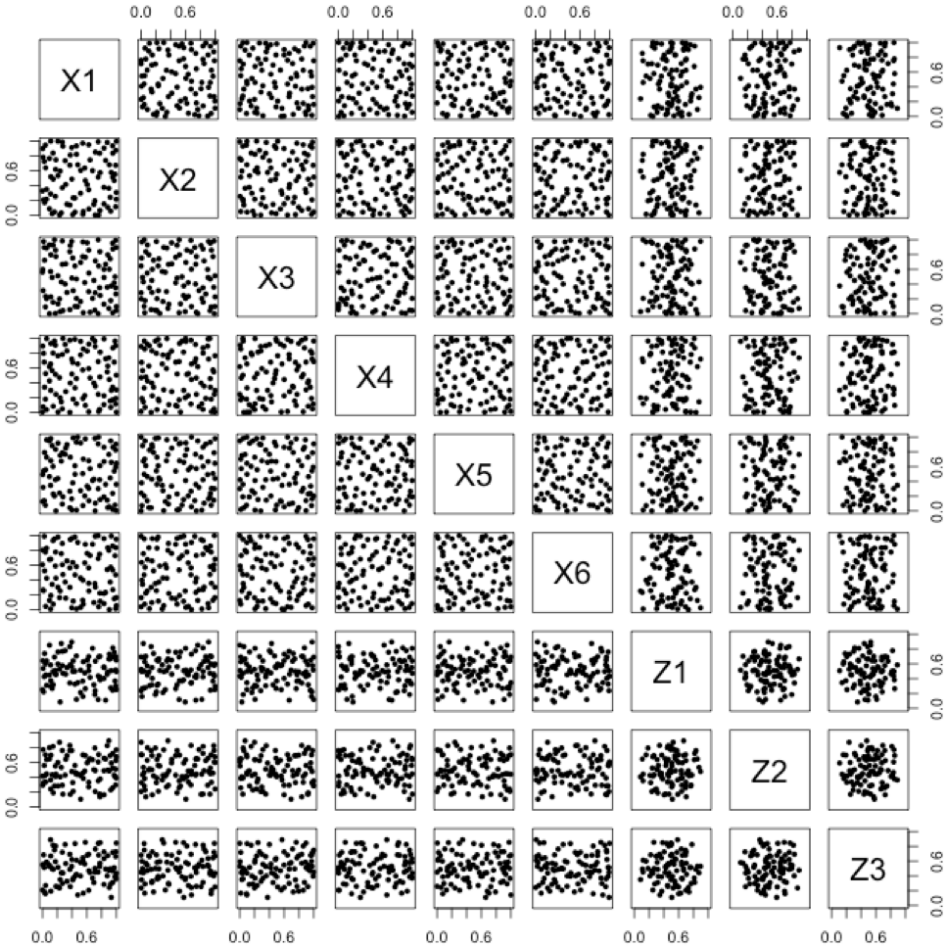


Figure 20: MED for P&G packing example.

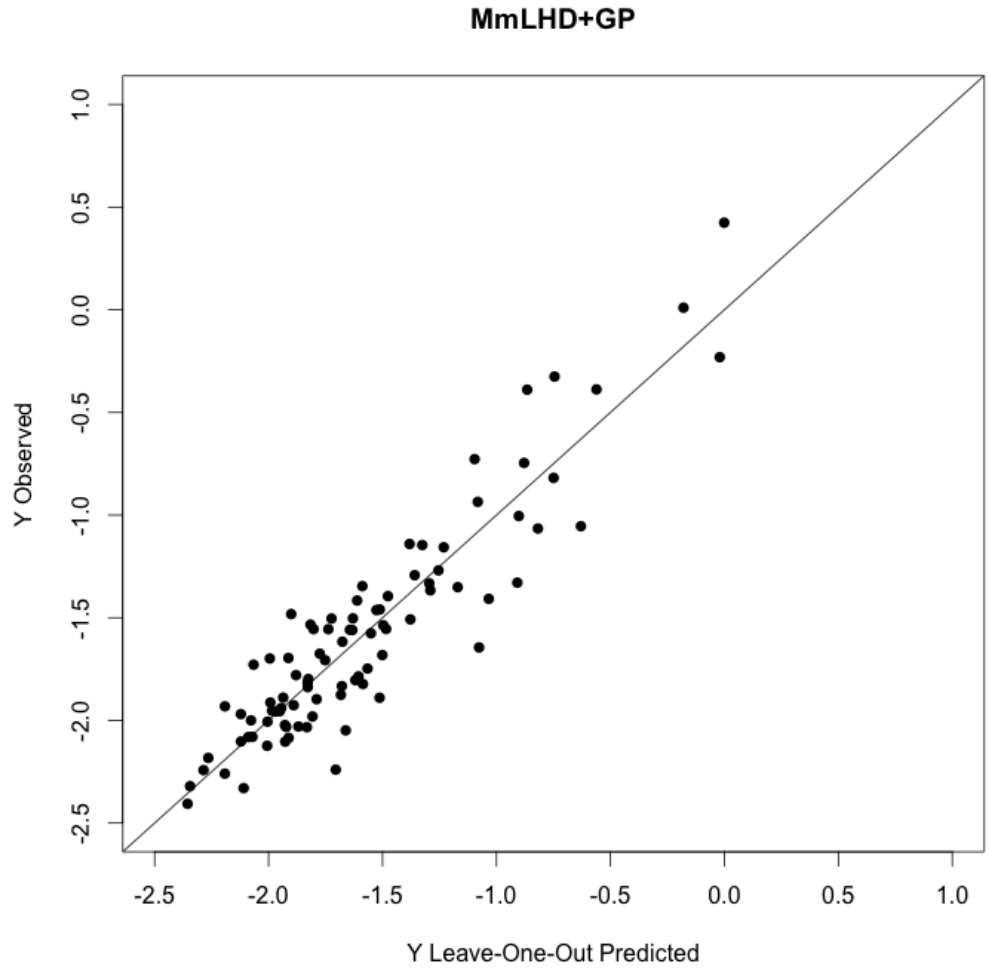


Figure 21: True response vs. LOOCV predicted response for P&G packing example. The design is MmLHD and the modeling method is ordinary GP model.

to the 45 degree line, which indicates New GP models further outperform ordinary GP models when MED is used.

These improvements are also confirmed by mean squared LOOCV prediction errors, which are 0.0479, 0.0260 and 0.0226, respectively. MED+NewGP improve by 45.7% from design and by further 13.1% from modeling. We point out that since noise factors in MmLHD do not follow normal distributions, mean squared LOOCV prediction errors for MmLHD+GP were calculated with the weights of normal distributions.

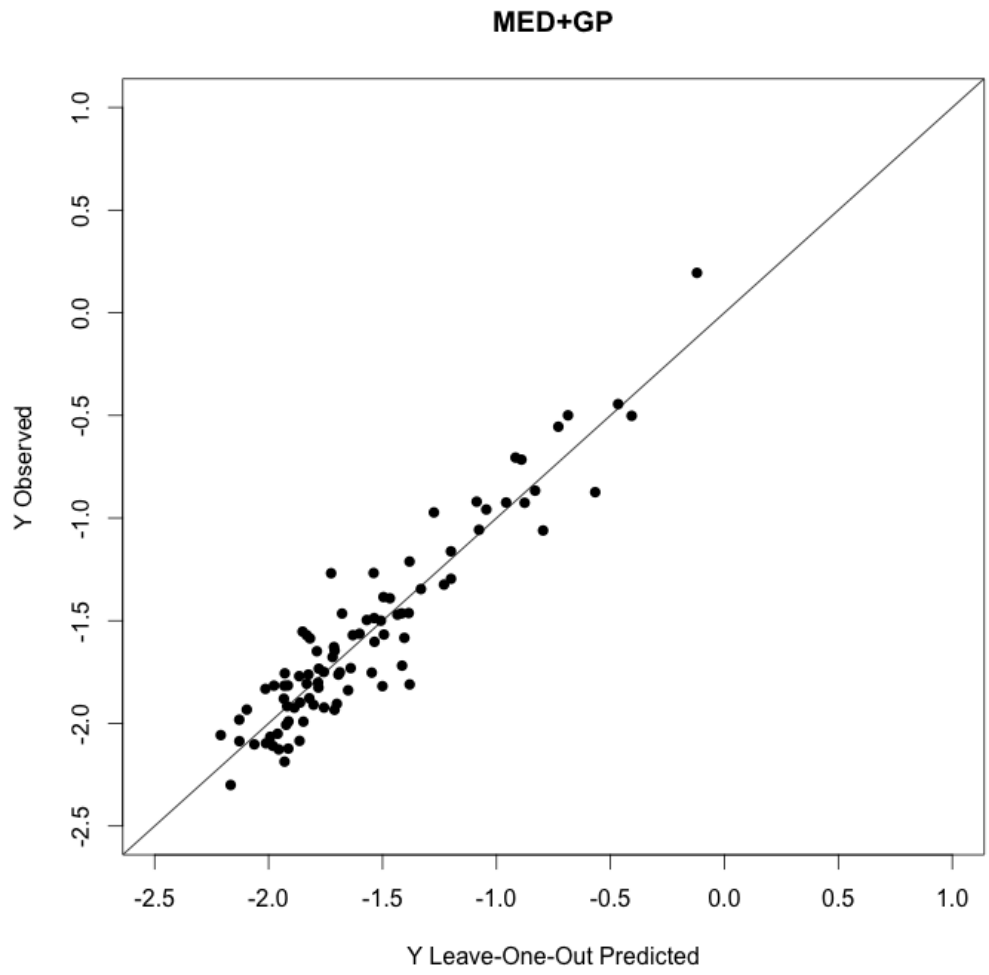


Figure 22: True response vs. LOOCV predicted response for P&G packing example. The design is MED and the modeling method is ordinary GP model.

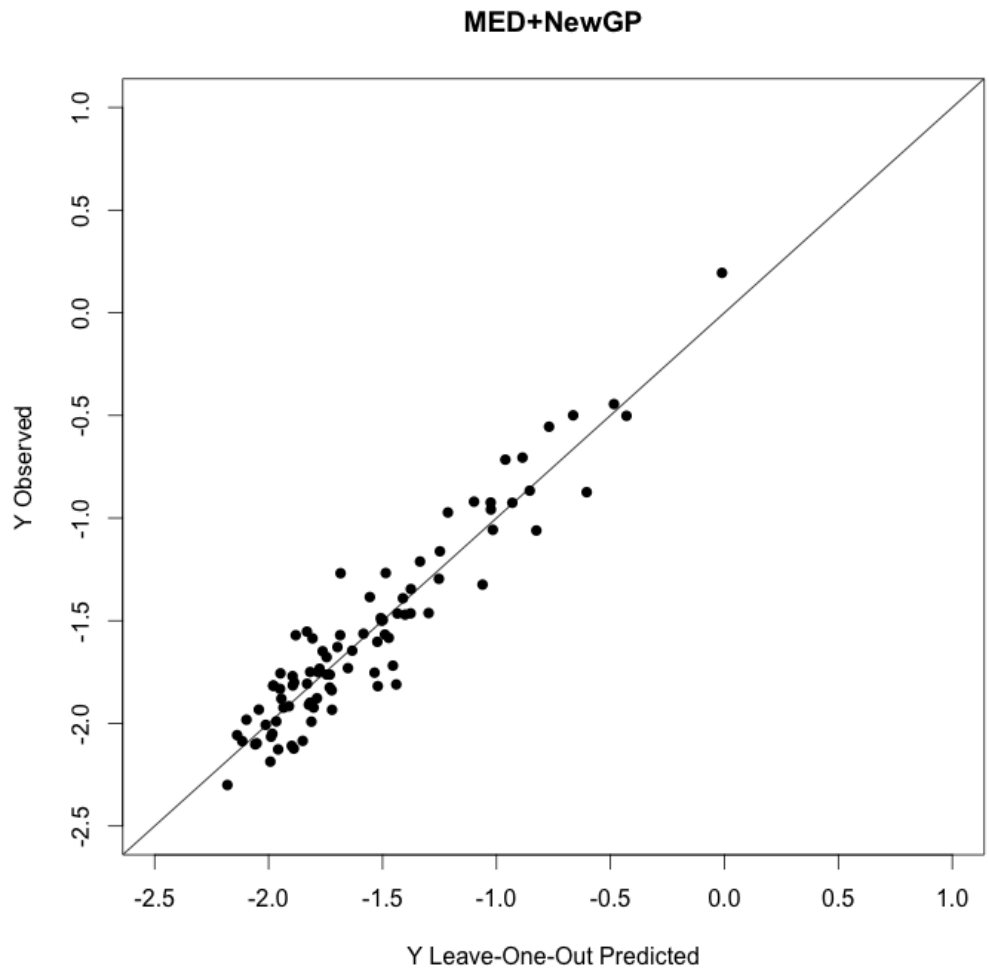


Figure 23: True response vs. LOOCV predicted response for P&G packing example. The design is MED and the modeling method is New GP model.

3.7 Conclusions

In this chapter, we have developed new design and modeling methods for robust parameter design in computer experiments. In the design part, a new design based on the generalized MED criterion has been proposed, where different tuning parameters are used for control and noise factors. Since the design points are not equally-spaced, stationary covariance functions can lead to numerical instability in computation and tend to perform poorly in prediction. In the modeling part, we have proposed a simple but efficient nonstationary Gaussian process that takes into account of the experimental design structure to solve this potentially difficult problem. Both the proposed design and model have been demonstrated to improve the performance over conventional methods using simulated examples and a real example on P&G packaging process.

REFERENCES

Apley, D. W., and Kim, J. (2011), "A Cautious Approach to Robust Design with Model Parameter Uncertainty," *IIE Transactions*, 43(7), 471–482.

Apley, D. W., Liu, J., and Chen, W. (2006), "Understanding the Effects of Model Uncertainty in Robust Design with Computer Experiments," *Journal of Mechanical Design*, 128(4), 945–958.

Ba, S., and Joseph, V. R. (2012), "Composite Gaussian Process Models for Emulating Expensive Functions," *The Annals of Applied Statistics*, 6(4), 1838–1860.

Bates, R. A., Kenett, R. S., Steinberg, D. M., and Wynn, H. P. (2006), "Achieving Robust Design from Computer Simulations," *Quality Technology and Quantitative Management*, 3(2), 161–177.

Bingham, D., and Sitter, R. R. (2003), "Fractional Factorial Split-Plot Designs for Robust Parameter Experiments," *Technometrics*, 45(1), 80–89.

Bliznyuk, N., Ruppert, D., Shoemaker, C., Regis, R., Wild, S., and Mugunthan, P. (2008), "Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation," *Journal of Computational and Graphical Statistics*, 17(2), 270–294.

Booker, A. (2000), "Well-Conditioned Kriging Models for Optimization of Computer Simulations," *Mathematics and Computing Technology Phantom Works - The Boeing Company*, Seattle, WA, M&CTTECH–00–002.

Bornkamp, B. (2011), "Approximating Probability Densities by Iterated Laplace Approximations," *Journal of Computational and Graphical Statistics*, 20(3), 656–669.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011), *Handbook of Markov Chain Monte Carlo*, Boca Raton, FL: CRC Press.

Chen, W., Jin, R., and Sudjianto, A. (2006), "Analytical Global Sensitivity Analysis and Uncertainty Propagation for Robust Design," *Journal of Quality Technology*, 38(4), 333–348.

Chib, S. (1995), "Marginal Likelihood from the Gibbs Output," *Journal of the American Statistical Association*, 90(432), 1313–1321.

Chivers, C. (2012), *MHadaptive: General Markov Chain Monte Carlo for Bayesian Inference Using Adaptive Metropolis-Hastings Sampling*, R package version 1.1-8, available at `http://cran.r-project.org/web/packages/MHadaptive/`.

Fang, K.-T., Li, R., and Sudjianto, A. (2006), *Design and Modeling for Computer Experiments*, Boca Raton, FL: Chapman & Hall/CRC.

Fielding, M., Nott, D. J., and Liong, S.-Y. (2011), "Efficient MCMC Schemes for Computationally Expensive Posterior Distributions," *Technometrics*, 53(1), 16–28.

Gelfand, A. E., and Smith, A. F. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85(410), 398–409.

Gelman, A., and Rubin, D. B. (1992), "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7(4), 457–511.

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.

Geyer, C. J. (1991), "Markov Chain Monte Carlo Maximum Likelihood," in *Computing Science and Statistics: Proc. 23rd Symposium on the Interface*, pp. 156–163.

Geyer, C. J., and Thompson, E. A. (1995), "Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference," *Journal of the American Statistical Association*, 90(431), 909–920.

Gilks, W. R., Roberts, G. O., and Sahu, S. K. (1998), "Adaptive Markov Chain Monte Carlo Through Regeneration," *Journal of the American Statistical Association*, 93(443), 1045–1054.

Gramacy, R., Samworth, R., and King, R. (2010), "Importance Tempering," *Statistics and Computing*, 20(1), 1–7.

Gu, L., and Joseph, V. R. (2016), "Exploratory Proposals for Independence Sampler," *Manuscript*.

Haario, H., Saksman, E., and Tamminen, J. (2001), "An Adaptive Metropolis Algorithm," *Bernoulli*, 7(2), 223–242.

Harari, O., and Steinberg, D. M. (2014), "Convex Combination of Gaussian Processes for Bayesian Analysis of Deterministic Computer Experiments," *Technometrics*, 56(4), 443–454.

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57(1), 97–109.

Ishigami, T., and Homma, T. (1990), "An Importance Quantification Technique in Uncertainty Analysis for Computer Models," in *Proceedings of the ISUMA '90, First International Symposium on Uncertainty Modeling and Analysis*, pp. 398–403.

Jin, R., Chen, W., and Sudjianto, A. (2005), "An Efficient Algorithm for Constructing Optimal Design of Computer Experiments," *Journal of Statistical Planning and Inference*, 134(1), 268–287.

Joseph, V. R. (2004), "Quality Loss Functions for Nonnegative Variables and Their Applications," *Journal of Quality Technology*, 36(2), 129–138.

Joseph, V. R. (2007), "Taguchi's Approach to Robust Parameter Design: A New Perspective," *IIE Transactions*, 39(8), 805–810.

Joseph, V. R. (2012), "Bayesian Computation Using Design of Experiments-Based Interpolation Technique," *Technometrics*, 54(3), 209–242.

Joseph, V. R. (2013), "A Note on Nonnegative DoIt Approximation," *Technometrics*, 55(1), 103–107.

Joseph, V. R. (2016), "Space-Filling Designs for Computer Experiments: A Review," *Quality Engineering*, 28(1), 28–35.

Joseph, V. R., Dasgupta, T., Tuo, R., and Wu, C. F. J. (2015), "Sequential Exploration of Complex Surfaces Using Minimum Energy Designs," *Technometrics*, 57(1), 64–74.

Joseph, V. R., Gul, E., and Ba, S. (2015), "Maximum Projection Designs for Computer Experiments," *Biometrika*, 103(1), 1–10.

Joseph, V. R., and Hung, Y. (2008), "Orthogonal-Maximin Latin Hypercube Designs," *Statistica Sinica*, 18(1), 171–186.

Kang, L., and Joseph, V. R. (2009), "Bayesian Optimal Single Arrays for Robust Parameter Design," *Technometrics*, 51(3), 250–261.

Kang, L., Roshan Joseph, V., and Brenneman, W. A. (2011), "Design and Modeling Strategies for Mixture-of-Mixtures Experiments," *Technometrics*, 53(2), 125–136.

Kou, S. C., Zhou, Q., and Wong, W. H. (2006), "Equi-Energy Sampler with Applications in Statistical Inference and Statistical Mechanics," *The Annals of Statistics*, 34(4), 1581–1619.

Lehman, J. S., Santner, T. J., and Notz, W. I. (2004), "Designing Computer Experiments to Determine Robust Control Variables," *Statistica Sinica*, 14(2), 571–590.

Li, W. W., and Wu, C. F. J. (1997), "Columnwise-Pairwise Algorithms with Applications to the Construction of Supersaturated Designs," *Technometrics*, 39(2), 171–179.

Liang, F., and Wong, W. H. (2001), "Real-Parameter Evolutionary Monte Carlo with Applications to Bayesian Mixture Models," *Journal of the American Statistical Association*, 96(454), 653–666.

Liu, J. S. (1996), "Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling," *Statistics and Computing*, 6(2), 113–119.

Marinari, E., and Parisi, G. (1992), "Simulated Tempering: A New Monte Carlo Scheme," *Europhysics Letters*, 19(6), 451–458.

Meng, X.-L., and Wong, W. H. (1996), "Simulating Ratios of Normalizing Constants via a Simple Identity: A Theoretical Exploration," *Statistica Sinica*, 6(4), 831–860.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, 21(6), 1087–1092.

Morris, M. D., and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43(3), 381–402.

Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2016), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, New York: Wiley.

Neal, R. M. (1999), "Erroneous Results in 'Marginal Likelihood from the Gibbs Output'," Technical Report, available at `http://www.cs.toronto.edu/~radford/chib-letter.html`.

Nelder, J. A., and Mead, R. (1965), "A Simplex Method for Function Minimization," *The Computer Journal*, 7(4), 308–313.

Owen, A. B. (1998), "Scrambling Sobol' and Niederreiter–Xing Points," *Journal of Complexity*, 14(4), 466–489.

Postman, M., Huchra, J. P., and Geller, M. J. (1986), "Probes of Large-Scale Structure in the Corona Borealis Region," *The Astronomical Journal*, 92, 1238–1247.

Rasmussen, C. E., Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M. (2003), "Gaussian Processes to Speed Up Hybrid Monte Carlo for Expensive Bayesian Integrals," in *Bayesian Statistics 7*, pp. 651–659.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4(4), 409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer.

Shoemaker, A. C., Tsui, K.-L., and Wu, C. F. J. (1991), "Economical Experimentation Methods for Robust Design," *Technometrics*, 33(4), 415–427.

Taguchi, G. (1987), *System of Experimental Design*, White Plains, FL: Unipub/Kraus International.

Tan, M. H. Y., and Wu, C. F. J. (2012), "Robust Design Optimization with Quadratic Loss Derived from Gaussian Process Models," *Technometrics*, 54(1), 51–63.

Tan, M. H. Y. (2015), "Robust Parameter Design With Computer Experiments Using Orthonormal Polynomials," *Technometrics*, 57(4), 468–478.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, 22(4), 1701–1728.

Tuo, R., and Lv, S. (2016), "Limiting Distributions of the Minimum Energy Designs," *Manuscript*.

Wang, D., Gu, L., and Joseph, V. R. (2016), "Bayesian Computation Using Minimum Energy Designs," *Manuscript*.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34(1), 15–25.

Williams, B. J., Santner, T. J., and Notz, W. I. (2000), "Sequential Design of Computer Experiments to Minimize Integrated Response Functions," *Statistica Sinica*, 10(4), 1133–1152.

Wu, C. F. J., and Hamada, M. S. (2009), *Experiments: Planning, Analysis, and Optimization*, New York: Wiley.

Wu, C. F. J., and Zhu, Y. (2003), "Optimal Selection of Single Arrays for Parameter Design Experiments," *Statistica Sinica*, 13(4), 1179–1199.

Ye, K. Q. (1998), "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association*, 93(444), 1430–1439.