Lívia Martins da Costa Furtado Pimentel

Efficient Stochastic Optimization Through Variance Reduction Techniques and Thorough Assessment of High-Dimensional Spaces

Belo Horizonte
2017

Lívia Martins da Costa Furtado Pimentel

Efficient Stochastic Optimization Through Variance Reduction Techniques and Thorough Assessment of High-Dimensional Spaces

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais, para a obtenção de Título de Doutor em Engenharia de Produção, na Linha de Pesquisa Modelagem Estocástica e Simulação.

Orientador: Leonardo Pereira Santiago
Co-Orientador: Pirooz Vakili

Belo Horizonte
2017

Martins da Costa Furtado Pimentel, Livia

Efficient Stochastic Optimization Through Variance Reduction Techniques and Thorough Assessment of High-Dimensional Spaces

167 páginas

Tese (Doutorado) - Escola de Engenharia da Universidade Federal de Minas Gerias. Belo Horizonte. Departamento de Engenharia de Produção.

1. Stochastic Optimization
2. Monte Carlo Simulation
3. Contro Variates
4. Metamodeling
5. High Dimensions

I. Universidade Federal de Minas Gerais. Escola de Engenharia. Departamento de Engenharia de Produção.

Comissão Julgadora:

Orientador

Dr. Leonardo Pereira Santiago

Membro Externo 1

Dr. Reinaldo Castro Souza

Membro Externo 2

Dr. Reinaldo Morabito Neto

Membro Interno 1

Dr. Alexandre Salles da Cunha

Membro Interno 2

Dr. Maurício Cardoso de Souza

Acknowledgments

I would first and foremost like to thank Professor Leonardo Santiago, who has been my advisor and mentor over the past five years. The transformation path he first offered is beyond engineering, and I will gladly carry valuable lessons for live. His genuine guide opened more doors than I could pass through, and if I went this far, is because of his beliefs on me.

I would also like to express my deepest gratitude to Professor Vakili. His ability in extracting the essence of the topics we have discussed, and his simplicity in giving intuitive explanations have impressed me. I will be always indebt for the knowledge and time he kindly shared with me, without which this thesis would not have been written.

I would like to thank my thesis committee, Professor Alexandre Salles, Carlos Andrey, Reinaldo Castro and Reinaldo Morabito for the goodwill and the opportunity in discuss my thesis topic. I am very obliged to Professor Tiago Schieber, as a good colleague and now kindly taking part of this committee. Last, but not least, I would like to extend my thanks and appreciation to Professor Maurício de Souza, who has been participating of my educational path since my first days as undergrad student.

I must thank to the community of the Federal University of Minas Gerais and to the Brazilian people who support this educational institution. I sincerely hope to reciprocate the trust and effort that has been disposed in my technical education. In particular, I would like to thank the professors of the Department of Production Engineering for their invaluable support at different moments of this journey.

I would also like to thank the academic community in general, for their service in creating and sharing knowledge from very different topics and perspectives. In particular, I would like to thank Paul Glasserman for his book (Glasserman (2004)), which I have rely on some many times.

I would like to thank my colleagues from LADEC, for their constant support as well as the interesting discussion on so many topics along the way. In particular, I am obliged to Hendrigo, for the companionship at all moments.

Finally, I thank my parents Beatriz e Geraldo for the fundamental support, and my grandmother Rilma for the devotion. My brother, my sisters-in-law, and all my family

also deserve special thanks for love and their unwavering support.

I dedicate all the time and effort disposed in this research to my husband Marcelo and to my son Pedro, whose love and encouragement are the greatest source of my strength.

Research supported by the Social Demand scholarship from CAPES Foundation, Ministry of Education, Brazil, and grant number BEX 3266/14-1.

Resumo

Otimização estocástica é uma área de pesquisa fértil e entusiástica. Seus métodos buscam soluções ótimas ou quase ótimas de problemas em que a incerteza não pode ser negligenciada. A otimização estocástica pode ser utilizada para modelar uma vasta gama de problemas, como sistemas de energia, manutenção, indústria química, suporte a tomada de decisão, geociências, saúde, cadeias de suprimentos, gestão de risco e gestão de filas. A despeito de sua abrangente aplicabilidade, a literatura atual indica uma demanda clara por técnicas mais eficientes que possam lidar com problemas mais complexos e de larga escala. Nós abordamos (i) a questão da dependência dos modelos em escolhas de usuários sobre parâmetros de entrada, que podem levar a modelagens ruins; (ii) exploramos o potencial de técnicas de redução de variância para aumentar a eficiência da simulação de Monte Carlo embutida nos algoritmos; (iii) e investigamos problemas de otimização em grandes dimensões. Para posicionar esta pesquisa na literatura atual, oferecemos uma revisão abrangente sobre métodos de otimização estocástica. Em particular, introduzimos as principais características de cada método, suas técnicas mais utilizadas, seus benefícios e limitações, as tendências atuais de pesquisa, e discutimos algumas lacunas ainda a serem investigadas. Em seguida, oferecemos três contribuições inter-relacionadas. Primeiro, um novo modelo de Metamodeling baseado em control variates é apresentado. A principal contribuição é propor uma formulação de metamodelo que é, ao mesmo tempo, computacionalmente eficiente e flexível o suficiente para possibilitar aplicação a uma ampla classe de problemas, com diferentes formatos da função objetivo e de comportamentos de incerteza (variância). A formulação proposta é menos dependente de parâmetros de entrada que os atuais metamodelos disponíveis. Nossa segunda contribuição é propor um procedimento via control variates para melhorar a eficiência de um método de busca aleatória. A novidade deste nosso procedimento híbrido é usar as saídas de pontos já amostrados para guiar a redução de variância em pontos a serem amostrados. O procedimento proposto é genérico no sentido de que pode ser aplicado a um conjunto mais amplo de métodos de otimização estocástica. Finalmente, mergulhamos em espaços de grandes dimensões para possibilitar o desenvolvimento de métodos de otimização estocástica mais sofisticados que possam lidar eficientemente com o aumento dimensional que caracteriza aplicações reais.

Abstract

Stochastic optimization is a fertile and exciting area of research. These methods aim at finding optimal or near optimal solutions of problems for which uncertainty cannot be neglected. Stochastic optimization can be used to model a vast range of problems, such as power system, maintenance, chemical industry, decision support, geosciences, health care, supply chain, risk management and queuing system. In spite of its wide applicability, current literature indicates a clear demand for efficient techniques that can handle large-scale and more complex problems. We address (i) the issue of model dependence at practitioners choice on input parameters which can lead to poor models; (ii) we explore potentials of variance reduction techniques to increase efficiency of Monte Carlo simulation embedded in algorithms; (iii) and we investigate high-dimensional optimization problems. To position this research on the current literature, we offer a comprehensive survey on stochastic optimization methods. In particular, we introduce the main features of each method, their most commonly used techniques, their benefits and limitations, explore current research trends, and discuss some gaps yet to be investigated. Then we offer three interrelated contributions. First, a novel Metamodeling framework based on vontrol variates is presented. The main contribution is to propose a metamodel formulation which is, at the same time, computationally efficient and flexible enough so that it can be applied to large class of problems, characterized by different shapes of objective function and uncertainty behavior (variance). We remark the proposed formulation is less dependent on practitioners choice on input parameters than the current available metamodels. Our second contribution is to propose a procedure via control variates to improve the efficiency of a random search method in finding optimal values. The novelty of our hybrid procedure is to use the output of already sampled points to guide a reduction of variance of the new sampled points. We remark that the proposed procedure is generic in the sense that it can be applied to a larger set of stochastic optimization methods. Finally, we take a deep dive into optimization in high-dimensional space. We derive properties of high-dimensional space to guide the design of more sophisticated stochastic optimization methods that can efficiently handle the dimensionality increase that characterize real applications.

Contents

List of Figures

List of Tables

List of Abbreviations

AAE	.....	Average Absolute Error
AHA	.....	Adaptive Hyperbox Algorithm
CV	.....	Control Variates
MAE	.....	Maximum Absolute Error
MPA	.....	Most Promising Area
MSE	.....	Mean Squared Error
SK	.....	Stochastic Kriging

Chapter 1

Introduction

There is a vast of situations in human activities where one must choose among available options the one that delivers the best response. Such situations are easily found in many fields, such as decision support, risk management, reliability, industry, health care, chemistry, supply chain, power system, among others. There may be an infinity number of options (or *decision variables*). Furthermore, the response is typically involved with some *uncertainty*. That is, the outcome of a choice is associated with a probabilistic distribution.

The major goal of stochastic optimization methods is to find such decision variables that give the best expected value of an objective function within a limiting budget (i.e., usually time or computational effort). More formally, the methods are looking for a solution $\boldsymbol{x}^*$ in the feasible space Θ such that:

$$J(\boldsymbol{x}^*) = \min_{\boldsymbol{x} \in \Theta} \mathrm{E}[Y(\boldsymbol{x}, \boldsymbol{w})],$$

where J is the objective function, Y is the response variable, and the expectation is with respect to the stochastic vector (or noise) $\boldsymbol{w}$. The only assumption on Y is that it can be modeled as a simulation system. That is, one can observe the system performance Y by running simulation experiments at $\boldsymbol{x}$ and randomly generating $\boldsymbol{w}$ according to the underlying probability measure.

The stochastic optimization methods have foundations on two areas of research that did not communicate much until recently: simulation, and deterministic optimization. Since the demand increased for methods that are able to handle more complex problem

within tractable computational time, the integration of deterministic optimization and simulation techniques became a very attractive area of research.

The cost (or effort) in generating each observation of the system response increases as problem complexity gets higher. Therefore, it is of great interest to take advantage of all the intrinsic information of system's outputs. The knowledge on how to extract and how to exploit such information draws on two broad strategies for increasing the efficiency of stochastic optimization methods: achieving better solutions (i.e., increasing the average on the values of best solutions found); and enhancing the ability of consistently finding good solutions at different noise outcomes (i.e., decreasing the variance on the values of best solutions found).

In this thesis, we focus on variance reduction techniques for increasing the efficiency of Monte Carlo simulations embedded in stochastic optimization methods. The main goal of this research is to provide flexible formulations for stochastic optimization methods that enable a greater exploitation of the information collected in all outputs of system's response.

The "control variates is among the most effective and broadly applicable technique for improving the efficiency of Monte Carlo simulation" (Glasserman (2004)). It allows the exploitation of information about the errors in estimates of a modeled system with known expectations (named control variables) to reduce the error in an estimate of another modeled system with unknown expected response (named variables of interest). The effectiveness of the control variates technique is determined by the strength of the correlation between the control variable and variable of interest. That is, the correlation is one measure that captures how informative are the errors of one variable to explain the errors of the variable of interest.

However, the requirement of known means for the control model limits its potential scope. Typically, the available variables that are very explicative about the variable of interest (i.e., highly correlated) are of a similar nature, and likewise do not have known means. Recently, in Zhao et al. (2007), Borogovac and Vakili (2008) and Borogovac (2009), proposed a control variates approach that relax the assumption of known control means. Such an approach is suitable for solving parametric estimation problems that requires estimating the same function at multiple parameter values.

In the context of stochastic optimization, the parametric estimation problem can be seen as estimating the stochastic function to be optimized at different decision variables (or solution points). We see the opportunity to invest computational effort in simulation outputs of a particular set of solution points in a stochastic optimization problem to make the estimation of function value at other solution points less costly. "The idea is to computationally learn to be computationally more efficient" (see Borogovac (2009), Chapter 5).

Next, we consider high-dimensional optimization problems, which is one of the most frequently barriers of stochastic optimization methods. There is a growing demand of unsolved stochastic optimization problems because current methods can not scale-up well as dimensions increase. The poor performance of stochastic optimization methods in high-dimensional space is also related to our difficulties in visualizing more than 3-dimensional spaces. "Such a limit hinders the development of intuitive sampling approaches, and also hinders our understanding of such a vast space" (see Shan and Wang (2010)).

A deeper theoretical analysis of the high-dimensional space is felt needed. It could guide the development of more efficient/generic simulation and optimization techniques. We go further on a deep dive into such a vast space to understand the effects of problem dimension on the value of output's information. We bring light into the roots of why the performance of stochastic optimization methods deteriorates with an increase of dimensionality.

1.1 Contributions of the Thesis

Our main contributions regarding the frontier of stochastic optimization and variance reduction techniques is twofold: (i) we derive a metamodeling formulation that has control variates in its foundations; (ii) and we provide a procedure that enables the use of control variates technique to a wide set of stochastic optimization methods. Further, we conduct a seminal investigation on the effects of problem dimension on the performance of stochastic optimization algorithms.

A Metamodeling Based on Control Variates

Metamodeling is one of the most commonly used tools for stochastic optimization. We

formulate a novel metamodel based on the variance reduction technique of control variates. Our formulation is flexible in the sense that it can be applicable to a large set of objective functions without requiring assumptions such as smooth surfaces or homogeneous noise. The formulation is less dependent on input parameters in comparison to current popular metamodels, and therefore is less susceptible to model misspecifications.

In the core of our formulation, we use database control variates technique to reduce the variance of response estimates at prediction points. As control variables, we use the outputs of a set of design points. In other words, the formulation consists in allocating a larger effort (more simulation outputs) in estimating the response value at a set of design points. Then, we use the correlation between design and prediction outputs to guide a variance reduction on the estimates at the latter points. Results show significant gain in both model accuracy (lower mean squared errors) and robustness (lower standard deviation of mean squared errors).

Moreover, we conduct a thorough analysis on using multiple controls in our metamodel. In particular, we investigate the negative effects that multicollinearity can cause in the estimates when the outputs of more than one design point are used to guide the variance reduction at a prediction point. Finally, we provide a procedure for better choosing the location and set size of design points. The intuition is that more design points are needed in regions of the surface with low correlations, whereas redundant design points can be discarded in regions with high correlations. The goal is to allocate in a more efficient manner the simulation budget among design and prediction point.

A Hybrid Fomulation of Random Search Method and Control Variates

We propose a hybrid method that combines random search stochastic optimization to database control variates. Random search is a method under the umbrella of the meta-heuristic class of stochastic optimization that aims at solving highly complex problems such as NP-hard ones. Our main contributions lies in the proposal of general framework that allows the direct application of database control variates to a larger set of stochastic optimization methods.

In random search, such as in other stochastic optimization methods, a solution can be revisited. That is, as the algorithm runs, a solution may receive additional samples (or

outputs). The intuition of our hybrid framework is to use the outputs of solutions that have received larger simulation effort to play the role of controls, in the database control variates technique. Therefore, the errors raised between outputs and estimated function value at these control points can guide a variance reduction of the estimates at solutions that have received a lower number of samples.

Experimental results on canonical problems show a notable gain in the ability of the hybrid method of consistently finding good solutions at different replications in comparison to a standard random search procedure.

A Finite Time Analysis of the High-Dimensional Space

We conduct a seminal analysis of optimization in high-dimensional problems under the perspective of finite time measure in contrast to the asymptotic convergence rate measure. In the essence, we believe that the latter measure tell us how the algorithm behaves once it has reached the vicinity of the optimum. That is, it does not give us an idea of how the algorithm moves from an arbitrary starting point to somewhere close to optimum. In particular, we are interested in understanding how dimensions influences such finite time measure.

We first introduce an elementary algorithm of gradient-based optimization with a single sample that nonetheless captures some key elements of the effects of the problem dimension on their performance. In our findings, we derive the probability of the algorithm in not detecting a better solution within a single iteration, and the amount of movement it can achieve. We demonstrate that the key element of why the optimization algorithm may be very inefficient in high-dimensional problem is the effect on dimension on the cosine of the angle between the random direction of movement and the gradient direction.

Moreover, we introduce an elementary algorithm based on linear approximation with multiple samples. We demonstrate that there is a tradeoff between obtaining more accurate estimates of the gradient which provides a longer step towards the optimum, and moving based on more noisy estimates of the gradient that requires smaller expenditure of computational resources.

1.2 Organization of the Thesis

In Chapter 2, we review stochastic optimization methods. Because this field of research is relatively new, the few available surveys do not cover all classes of techniques. We were careful to present the very recent and important developments to guarantee update of this thesis to the state-of-art because stochastic optimization is evolving rapidly. We organize our survey on five sections dedicating each one to a class: stochastic approximation, metamodeling, sample average approximation, ranking and selection, and metaheuristics. The survey offers main characteristics, limitations and benefits of the most commonly used tools for stochastic optimization problems. We give perspectives of future developments for each tool. It is important to understand their differences to allow comprehension of where our proposed approaches can be applied. Moreover, we offer our view on global potential directions of research in stochastic optimization based on the knowledge we have learned while reviewing the references herein. We use these global directions as guide for our proposals in the next chapters.

In Chapter 3 we present our main results from using a variance reduction technique as foundations to formulate a new metamodel framework. We examined our framework at four template deterministic functions by adding four type of noise to each one of them. We also utilize two classical stochastic problem - path-dependent options and M/M/1 queues - to illustrate practical applications. To conduct the experimental analysis, we utilized four performance measures that evaluate local and global prediction accuracy, robustness and efficiency of our metamodel. We compare the performance measures to the stochastic kriging tool, which is the metamodel that most drew attention in the past few years. Specific contributions of this Chapter are:

- Proposal of a metamodel tool based on database control variates. In our framework, the output of design points are used as control when estimating the function value at prediction points. Our metamodel is flexible in the sense that it does not require as many input parameters as in other metamodels. That is, the performance of our metamodel is less dependent on practitioner choices, which are sources of model misspecification that can lead to poor predictions.

- We investigate the possibility of using multiple controls in our metamodel. Using

connections to linear regression, in particular to multicollinearity theory, we demonstrate the reasons why using multiple controls is not appropriate. The induced correlation raised from the use of Common Random Numbers in the database control variates can increase significantly the variance of control variates coefficient.

- We introduce a procedure to iteratively select location and number of design points. A well-chosen location guarantees a minimal level of correlation between design points and prediction points, which improves the efficient of the control variates technique. By choosing carefully locations, redundant design points can be eliminated. Therefore, more simulation budget can be allocated to relevant design points, improving the quality of control mean estimates.

In Chapter 4, we propose a hybrid method of database control variates and random search to improve efficiency of the latter stochastic optimization method in finding the optimal solution. Analyzes are conducted using five template problems: inventory management problem, multimodal problem, singular function problem, high-dimensional problem, and high-dimensional multimodal problem. The performance measures are: average ability in finding good solutions, best and worst solutions among all replications, and standard deviation of solutions. Specific contributions of this Chapter are:

- We provide a variance reduction procedure embedded in a random search algorithm to improve its efficiency. In general, random search methods, which falls into the stochastic optimization class of metaheuristics, revisit a considerable number of times some solution points of the design space. We utilize these points as control variates to improve efficiency in estimating function value at other points as the algorithm randomly evolves towards the optimum. Experimental results show that our procedure has brought benefits not only at finding better solutions within small simulation budgets, but also to the robustness of search. That is, consistently finding better good solutions at different replications.

- The procedure is very general in the sense that it can be applicable to a larger set of stochastic optimization methods, such as: stochastic approximations, sample average approximation and metamodeling, and ranking and selection.

In Chapter 5 we analyze two elementary algorithms of gradient-based optimization. The objective is to better understand why algorithms may be very inefficient in high-dimensional search spaces. In particular, we propose a theoretical analysis of effects of dimension under the perspective of finite time measures in contrast to the asymptotic rate of convergence measure. Specific contributions of this Chapter are:

- We introduce two elementary algorithms that allows simplicity in capturing some key elements of the influence of the dimension of the search space on their performance.
- We provide finite time measures to analyze the performance of the two algorithms in each iteration. Our goal is to understand a local behavior of optimization algorithms under the effects of dimension. In particular, we derive the probability of finding a better solution in each iteration, and we derive the length of movement the algorithm can achieve in each iteration.
- We conduct experimental analyzes to better illustrate the relevant implications of dimension in the latter two finite time measures on the performance of the algorithms.

In Chapter 6 we summarize the key messages of each Chapter. Perspectives on research directions are discussed.

Chapter 2

Survey on Stochastic Optimization

Stochastic optimization is one of the most exciting research areas, due to its mathematical foundations and practical applications (see Fu et al. (2015)). It offers opportunities and challenges for researchers in many fields, from operations research/management science to mathematics to statistics to computer science to economics to most engineering fields (Chau et al. (2014)). It has also been referred to as simulation-based optimization, simulation optimization, parametric optimization, black-box optimization, and Optimization via Simulation, where the continuous and discrete versions are accordingly known as Continuous Optimization via Simulation and Discrete Optimization via Simulation (Amaran et al. (2014)).

Stochastic optimization approaches are a class of techniques that aims at solving optimization problems on set of decision variables associated to a performance function using simulation outputs or real-life experimentation. Many times the performance function and also its gradient (or higher order derivatives) are not known analytically, however, noisy samples obtained from simulation are available. The problem of interest is to perform optimization under such 'noisy' information, many times without knowing the system model (Bhatnagar and Prashanth (2015)).

In real applications, most processes are highly complex, making it impossible to develop realistic analytical models of the system, thus computer models become one of the best choices to represent these real complex systems (Yuan et al. (2013)). In the past 5 years, stochastic optimization approaches have been applied to a vast number of practi-

cal problems such as power systems, renewable energy, maintenance, chemical industry, decision support, geosciences, health care, supply chain, transportation, manufacturing, reliability, structural systems, risk management and queue systems.

The main goal of this Chapter is to position this thesis in the current literature. Because combining simulation and optimization techniques is a relatively new area of research, there has not been many surveys that cover a large part of stochastic optimization methods. We based our survey on the Handbook Fu (2014), on recent surveys that treat only some methods such as Fu (2002), Schueller and Jensen (2008), Wang and Shi (2013), Barrientos et al. (2014), Amaran et al. (2014), Chau et al. (2014) and Viana et al. (2014), and specially on a great number of published papers in the past few years.

There are many methods under the umbrella of stochastic optimization. Similarly to Fu (2014), we divide stochastic optimization methods into five categories: stochastic approximation (Section 2.1), Metamodeling (Section 2.2), sample average approximation (Section 2.3), ranking and section (Section 2.4) and metaheuristics (Section 2.5). Specifically, we provide an overview of each category by introducing their most commonly used tools and discuss the pros and cons of each of the described methods. In addition, we explore current research trends and highlight opportunities for future research. We close this Chapter by assessing the key aspects of each optimization category, drawing special attention to their respective research trends, and then connecting each one of the thesis' Chapters to the gaps in the literature.

2.1 Stochastic Approximation

The stochastic approximation method was introduced by Robbins and Monro (1951) to solve noisy root-finding problems. It mimics the deterministic descent method using unbiased direct gradient estimates. The stochastic zeroth-order method addressed by Kiefer and Wolfowitz (1952) uses the finite-difference gradient estimates. It differs from the Robbins-Monro algorithm in that it does not require additional information on the system dynamics or input distributions (Chau et al. (2014))

The stochastic approximation approach has been widely used to solve the problems on which the only information available of the objective function is noisy observations. Com-

pared to other methods such as genetic algorithm (see 2.5.1, metaheuristics), it is easier to understand, implement, and automate. Moreover, this approach can generally find good solutions within a reasonable computational search time (Yuan et al. (2013)). Because the data used in this approach is directly obtained from the simulation model, it avoids the bias introduced when building metamodels (Yuan et al. (2013)). According to Chau et al. (2014), the stochastic approximation requires very little memory and is currently one of the most widely applicable and most useful methods for stochastic optimization.

The main limitations of stochastic approximation procedures include, but are not limited to: some techniques are not suitable for large-scale problems; for problems with quick changes in the gradient and in the objective function, some techniques may present slow convergence or may diverge; and a performance dependence on the choice of initial parameters.

Nowadays, stochastic approximation has a wide variety of applications in areas such as adaptive signal processing, adaptive resource allocation in communication networks, system identification, adaptive control, and others (Granichin (2015)). Many well-known techniques are special cases of such a method, including neural network backpropagation, perturbation analysis for discrete-event systems, recursive least squares and least mean squares and some forms of simulated annealing (Spall (2000)). An overview of stochastic approximation can be found in Chau et al. (2014).

2.1.1 Stochastic Approximation Methods

The most commonly used variations of stochastic approximation methods are: finite-difference stochastic approximation; simultaneous perturbation stochastic approximation; iterate averaging (also referred to as robust stochastic approximation); and Kesten's rule. Next, we introduce each of the above techniques.

Finite-difference stochastic approximation

Finite-difference stochastic approximation (Spall (2003)) is a gradient based optimization algorithm designed to optimize objective functions under uncertainty. This method can be thought of as a generalization of line search optimization algorithms for stochastic objective functions (McGill et al. (2015)). It uses the finite-difference method to estimate

the gradient of the objective function and, thus, it has its basis on the Kiefer-Wolffowitiz algorithm.

This method perturbs the control variables only once at a time, so that its computational cost in every iteration is proportional to the number of control variables. In other words, it requires $2d$ simulations for one estimate of the performance gradient with respect to an d-dimensional decision variable optimization (see Glasserman (2004) for a review on finite-difference approximations). Therefore, it is difficult to use in high-dimensional problems (Zhou et al. (2013b)).

There have not been many recent developments in the finite-difference stochastic approximation algorithm. We can cite Yan and Reynolds (2014), that proposes an algorithm in which the components of largest magnitude of the stochastic gradient are replaced by a finite-difference approximation of the pertinent partial derivatives. Samadi et al. (2014) propose an iterative approach to design two real-time pricing algorithms based on finite-difference and simultaneous perturbation methods, respectively. In Khong et al. (2015), three discrete-time multivariate stochastic approximation algorithms (finite-difference stochastic approximation, random directions stochastic approximation, and simultaneous perturbation stochastic approximation) are adapted within a periodic sample-data framework .

Simultaneous perturbation stochastic approximation

An alternative approach to finite-difference stochastic approximation is the simultaneous perturbation stochastic approximation, initially proposed by Spall (1992) and successfully applied in the optimization of a variety of stochastic systems. The method approximates the gradient with only two successive measurements of the objective function independently of the dimension d, and therefore significantly saves computational time for large-scale problems over traditional stochastic approximation methods, in which the computational time directly depends on the problem dimension (Lu et al. (2015)).

The theoretical convergence of the simultaneous perturbation stochastic approximation algorithm along with several variations of it - including discrete simultaneous perturbation stochastic approximation (Wang and Spall (2013)), adaptive (second-order) simultaneous perturbation stochastic approximation (Spall (2000)), and global search simultaneous per-

turbation stochastic approximation (Maryak and Chin (2008)) - has been reported in the literature (Li et al. (2013)).

Due to the fact that this method is less impacted by the problem's dimensionality, it has been extensively investigated as one of the most suitable stochastic approximation methods to handle high-dimensional problems. However, current limitations of this method needs to be overcome in order to apply it to some of the open high-dimensional problems. In Chapter 5, we make a thorough discussion on why the performance are deeply affected by the increase of dimensions. Next, we discuss main disadvantages of simultaneous perturbation stochastic approximation.

While simultaneous perturbation stochastic approximation is relatively simple to implement, its performance depends on a set of parameters that need to be properly determined. For example, its performance is sensitive to the selection of the initial algorithmic parameters, the scale of decision variables, and the shape of response surface (i.e., objective function) and associated gradient. Another limitation is that the choice of the step size for updating the solution and other parameters can make the algorithm very slow if the function is steep. As a result, especially in cases where the gradient changes quickly, simultaneous perturbation stochastic approximation may not be as stable or even diverge (Tympakianaki et al. (2015)).

Now, we highlight the research considered more relevant to our thesis, among recent work on simultaneous perturbation stochastic approximation. Zhou et al. (2013a) proposed an improved simultaneous perturbation stochastic approximation method guided by a finite-difference gradient. The method adjusts the ratio among perturbation steps during the iterations, in order to guarantee similar magnitude contributions of different decision variables to the overall change in the objective function.

Lu et al. (2015) presents a simultaneous perturbation stochastic approximation algorithm that incorporates the information of spatial and temporal correlation in traffic network with a weight matrix to reduce the gradient approximation error and improve convergence and robustness. In Tympakianaki et al. (2015), a modified simultaneous perturbation stochastic approximation is proposed in order to improve its convergence and stability. The main idea of the modified algorithm is the clustering of the unknown variables into a small number of "homogeneous" clusters (based for example on their initial

values).

In Bhatnagar and Prashanth (2015), a new Hessian estimator based on simultaneous perturbation procedure is present. The estimator requires only three system simulation regardless of the decision variable dimension (the original second-order simultaneous perturbation stochastic approximation proposed by Spall (2000) requires four system simulations to estimate the Hessian). This Hessian estimator is used in Newton's-based stochastic optimization algorithms.

Iterate averaging

Iterate averaging approaches stochastic approximation from a different angle. Instead of fine-tuning the step sizes to adapt the function characteristics, it takes a larger step size to make estimates oscillating around the optimum, so the average of iterates will result in a good approximation to the true optimum. The idea of such a technique is simple, can be very effective, and is easy to adapt for other stochastic optimization methods. In order to be efficient, iterates must surround the optimum in a balanced manner, and the domain for which iterates oscillate must decrease as the number of samples increases. It is also expected that averaging trajectories reduces the sensitivity to initial step size choice (Chau and Fu (2014)).

The benefits of iterate averaging algorithms can be summarized by the following items: (i) the technique reduces the dependence on the choice of step size sequences by providing a systematic approach; (ii) with the use of a large step size the algorithm forces estimates to move towards the optimal decision variables more quickly; (iii) it alleviates the noise effect and reduces its variance (Yin et al. (2013)).

However, it is important to note that if averaging starts before oscillation, the average estimates might be worse than the standard procedure (Swersky et al. (2010)). In other words, the iterate averaging is dependent on the choice of when the averaging process starts, which is an practitioner choice. A recent overview of the iterate averaging can be found in Kushner and Yin (2003) and Chau and Fu (2014).

There has not been much research on iterate averaging recently. In our search, we highlight Lee and Wright (2013), that propose a method based on Iterate averaging as a subgradient algorithm for training support vector machines; and Yin et al. (2013), which

introduced a post-averaging algorithm to achieve asymptotic optimality in convergence rates of stochastic approximation algorithms with constraints. This algorithm involves two stages: (i) a coarse approximation obtained using a sequence of large step size; and (ii) a refinement by averaging iterates from the first stage. A weighted version of iterate-averaging can be found in Nedic and Lee (2014).

Kesten's rule

It is well-known that the choice of step size sequences has a significant impact on the performance of stochastic approximation algorithms (Chau et al. (2013)). Therefore, it could be advantageous to consider adaptive step sizes that make adjusts based on the ongoing performance of the algorithm. The main idea is to adapt the step size to characteristics of the response surface at the current decision variables, and in the neighborhood of optimal solution. The most used adaptive rule is the one proposed by Kesten (1958).

The notion behind Kesten's rule is that, if the iterates continue in the same direction, there is a reason to believe they are reaching the vicinity of the optimum, and the step size should not be decreased in order to accelerate the convergence. If the errors in estimate values change signs, it is an indication that either the step size is too large and the iterates are experiencing long oscillation periods, or iterates are in the vicinity of true optimum; either way, the step size should be reduced to get closer to optimum (Chau and Fu (2014)). The multi-dimensional variant of Kesten's Rule is provided in Delyon and Juditsky (1993).

One important limitation of Kesten's rule is its dependence to initial parameters values (Chau et al. (2013)). In Xu and Dai (2012), such a method is compared to other three stochastic approximation algorithms with adaptive step size. The results show that the proposed algorithms are more efficient than the Kesten's algorithm in most cases. An overview on Kesten's rule can be found in George and Powell (2006) and Chau and Fu (2014). A recent analysis of step size selection in stochastic approximation algorithms can be found in Wang (2015).

There are a few studies on the Kesten's rule. More recently, Chau et al. (2013) conducted an empirical investigation of the sensitivity of Kesten's rule. In this research, problem characteristics that exert a strong impact on the algorithm performance were identified, even in the presence of theoretical guarantees. Wang et al. (2015) considered

the multi-dimensional Kesten's rule. The latter research established convergence guarantees for the algorithm studied in Xu and Dai (2012).

2.1.2 Research Trends

Based on the references herein, we sort opportunity directions of stochastic approximation method in two topics: (i) comparison to other studies to assess overall performance; (ii) and broadly improvement of SA methods. In the first topic - comparison to other studies to check overall performance - we remark as future work: carrying out experiments of the stochastic approximation models on a larger set of template problems, over a large number replications, and to a variety of noise characteristics (Swersky et al. (2010)); comparing the proposed discrete simultaneous perturbation stochastic approximation results with the rate of convergence of other algorithms (Wang and Spall (2013)); benchmark the presented results of stochastic approximation method against the performance of current best practices in stochastic optimization (Li and Reveliotis (2015));

The second topic - improvement of stochastic approximation methods in a general manner - we remark as future work: developing quasi-Newton algorithms for stochastic optimization, and studying their performance characteristics as quasi-Newton algorithms are known to have lower computational requirements than pure Newton methods (Bhatnagar and Prashanth (2015)); reducing the search space and improving the convergence of simultaneous perturbation stochastic approximation by using information about the bounds of variables, and using parallel processing to improve computational efficiency (Lu et al. (2015)).

Final remarks on stochastic approximation

Stochastic approximation is one of the first methods developed to optimize problems under uncertainty. Recently, the simultaneous perturbations stochastic approximation algorithm has gain significant attention due to its good performance in high-dimensional problems. This method have still a long way to go to become a robust approach: there is a dependence on a set of initial parameters that need to be properly determined, mainly based on practitioner experience; the algorithm may be slow or diverge according to the response surface. Stochastic approximation research on accelerated approach, such as iterate-averaging and

Kesten's rule, has demonstrated improvement in the results when combined with other methods.

2.2 Metamodeling

The general idea of metamodeling (or surrogate modeling) is an analytical approximation of the objective function. In other words, it is a *model of the model* (Kusiak et al. (2015)). Traditionally, there are two essential procedures in metamodel optimization methods. First, the metamodel is fitted based on a set of simulated observations. Second, an optimization procedure is conduct and generates a trial point. The objective function at the trial point can be evaluated by simulation, which leads to new observations. As new observations become available, the accuracy of the metamodel can be improved. As result, better trial points are detected (Osorio and Bierlaire (2013)). Metamodeling techniques are often classified as being either global or local methods: global approximations are valid throughout the entire design space (or a large portion of it); whereas local approximations are only valid in the neighborhood of a particular point (Viana et al. (2014)). A nice review of metamodeling can be found in Viana et al. (2014). In Tabatabaei et al. (2015), is presented an overview and comparison of metamodeling methods.

In Chapter 3, we propose a metamodel based on variance reduction techniques. The intuitions to develop such a framework lies on current limitations of the available metamodeling tools. These limitations are discussed in the following section, while we introduce most used metamodeling tools.

2.2.1 Metamodel Methods

There exists a vast of metamodeling methods developed in literature. The most used ones in stochastic optimization context are: response surface methodology (also known as lower-order regression method), stochastic kriging, radial basis functions, multivariate adaptive regression splines, neural networks and support vector regression. A comparison of response surface methodology, stochastic kriging and artificial neural network can be found in Kusiak et al. (2015). Next, we describe each one of the above methods based on the latter research and Jin et al. (2001). An overview on various metamodeling techniques

can be found in Wang and Shan (2007).

Response surface methodology

The response surface methodology is one of the more popular methods in stochastic optimization. It was first proposed by Box and Wilson (1951). It is a stepwise heuristic that uses first-order polynomial regression to approximate the response surface locally. An estimated of local gradient is provided by the metamodel, and is utilized in steepest descent (or ascent) to decide on the next local experiment. When the method approaches the neighborhood of the optimum, the first-order polynomial regression is replaced by a second-order one, and a stochastic variation of Newton's method is applied (Kleijnen (2014)). The main assumptions of response surface methodology are: independent and normally distributed outputs; constant variance over the design space; and a number of observed points larger than problem dimension. As far as we know, there has been no convergence proofs. An overview of response surface methodology can be found in Kleijnen (2014).

Stochastic kriging

Stochastic kriging (also known as the Gaussian process model) is a very popular metamodel form for stochastic optimization. It aims to eliminate a basic difficulty in using response surface methodology: the selection of appropriate basis function. Such a method is based on the idea that the value of a given point can be estimated on the basis of an average of known values in the neighboring points. It is assumed that the influences of these points are proportional to the distance to the considered point. In other words, the approximation procedure has to follow the trends of experimental data, and the metamodel function should increase if an increment is observed in the outputs of vicinity points (Kusiak et al. (2015)). An important limitation of stochastic kriging is its dependence on spatial parameters, that must be subjectively chosen and this choice can be time-consuming. An overview on stochastic kriging can be found in Staum (2009). An object-oriented stochastic kriging implementation can be found in Couckuty et al. (2014).

Radial basis function

Radial basis function has been developed for scattered multivariate data interpolation. Such a method uses linear combinations of a radially symmetric function based on Euclidean distance or other metric to approximate the objective function. Radial basis function have been shown to produce good fits to different objective function (Jin et al. (2001)). The most commonly used radial functions are: Gauss function, second-order function, and inverse second-order function (Kusiak et al. (2015)). One of the main interesting features of this method is that the resulting optimization problem can be efficiently divided into linear and nonlinear subproblems (Cheng et al. (2015)). On the other hand, the basic difficulty in using radial basis function is the selection of an appropriate base function. If not well specified, the fitted base function may result in the increase of approximation error (Kusiak et al. (2015)). An extensive study on radial basis function can be found in Buhmann (2003).

Multivariate adaptive regression splines

Multivariate adaptive regression splines was proposed by Friedman (1991) for high-dimensional modeling. It provides a flexible statistical modeling method that employs forward and backward search algorithms to identify the combination of basis functions that best fits the data, and simultaneously conducts a search for best decision variables. After selection of the basis function is complete, the method applies a smoothing procedure to achieve continuity in the approximated function (Martinez et al. (2015)). Compared to other techniques, the use of such a method is relatively new. The major advantages of using multivariate adaptive regression splines is gain in prediction accuracy and less computational effort required when constructing the metamodel. However, its performance deteriorates significantly when simulation budget becomes small. Moreover, users need to configure initial parameters, which may deteriorate its performance depending on problem features (Jin et al. (2001)). An overview on multivariate adaptive regression splines can be found in Martinez et al. (2015).

Artificial neural networks

(Kusiak et al. (2015)) describe the artificial neural network as follows: it is an information processing system built with a given number of single elements called artificial neurons.

The neuron input vector is composed of a finite number of signals and each of these signals is multiplied by a synaptic weight coefficient. These weight parameters control the impact of the input signal on the neuron output. The goal of the artificial neuron is to generate the proper output signal that depends on the input signal, and that is close to the observed output of objective function. The neuron output signal depends on its activation function and the synaptic weights. While activation function is selected at the first step of the neuron design, the weight parameters undergo the variations during the neuron learning process. The learning target is to estimate the value of weight parameters that enables the trained neuron to react to the input signals as well as the response surface. The main benefit of such a method is its ability of learning and, as a consequence, achieving a good prediction capacity of nonlinear stochastic functions. Therefore, has been useful in modeling complex systems, and also computationally efficient. However it is worth noting that, in the case of artificial neural network metamodeling, the location of design points plays important role in the accuracy of prediction. According to Cheng et al. (2015), artificial neural network metamodeling is heavily dependent on the structure of the underlying network, and as a result require considerable tuning, similar to previously metamodel that dependent on a basis function. A guideline on artificial neural networks for engineering applications can be found in Rafiq et al. (2001).

Support vector regression

Support vector machine is a kind of machine learning technique with successful applications in regression. As described by (Chen and Yu (2014)), support vector regression is specifically used to predict unknown stochastic functions through nonlinear Kernel functions and a number of identified support vectors. Basically, such a method searches for the nonlinear regression function that is linear in high-dimensional space by solving a quadratic programming problem. The formulation embodies structural risk minimization principle, which has been shown to be superior to traditional empirical risk minimization principle employed by typical artificial neural networks. Therefore, support vector regression has enhanced ability in prediction and can avoid over-fitting issue. However, according to Kazem et al. (2013), the main challenge of support vector regression is determining its hyperparameters, which requires practitioner experience. Unsuitably chosen

Kernel functions or hyperparameters setting may lead to significant poor performance.

2.2.2 Research Trends

There are many recent developments on metamodelling approaches/research. We focus on two research streams: enhancing stochastic kriging, and variable-fidelity metamodeling. Both topics are discussed in Chapter 3, when we introduce a novel metamodel formulation. We use the insights gained while revising variable-fidelity metamodel to build our framework. Further, we compare the performance of our metamodel with stochastic kriging, which is the method that has mostly draw research focus in the past years. We close this subsection by discussing future research trends that were highlighted in the recent literature on metamodeling.

Enhancing Kriging

Stochastic kriging enhancement, in the context of stochastic optimization, is an active area of metamodeling research. Recently, Quan et al. (2013) investigates the incorporation of optimal computing budget allocation techniques in kriging method. In Sun et al. (2014), a sequential sampling approach was introduced to improve the fitness accuracy of such a metamodel. Chen and Kim (2014) proposed a stochastic kriging extension in recognition of bias present in simulation response estimates. The associated impact of the standard stochastic kriging predictors on the mean squared error is analyzed. Error estimation of stochastic kriging in the focus in Hernandez and Grover (2013). It provides an analysis of the error estimation properties in stochastic kriging when the simulated data observations contain measurement noise.

There is a number of research focusing in combining stochastic kriging with other techniques. For example, such a method is combined with principal component analysis in Jia and Taflanidis (2013) for solving problems with high-dimensional outputs. The principal component analysis is used to extract a much smaller number of latent outputs to approximate the initial high-dimensional response. A separate metamodel is then developed for each latent output. In Okobiah et al. (2014), an algorithm based on simulated annealing (see Section 2.5.1, metaheuristics) is used to optimize the stochastic kriging metamodel.

Variable-fidelity

Variable-fidelity metamodeling, which has gained increasing popularity, is a modeling process enhanced through the incorporation of knowledge (Zadeh et al. (2009)). In this approach, the model is hierarchical, in the sense that one set of data (the experiments) is considered to be more reliable and it is labeled as high-fidelity data; and the other set (the simulations) is labeled as low-fidelity data. As example of recent applications, we cite the approach developed in Zheng et al. (2014) where low-fidelity output serves as a prior-knowledge of the real response function and is used as inputs of the least squares support vector regression. Zhou et al. (2015) developed a generalized objective-oriented sampling strategy to adaptively probe and sample more points in the interesting regions, where the differences between the high-fidelity and low-fidelity models are multi-model, non-smooth and have abrupt changes. In Colosimo et al. (2015), data coming from simulation (low-fidelity) are used to produce a first stage metamodel with a kriging predictor. Then, a second-stage model is used in order to correct the prediction of the first model according to real experimental data observed (high-fidelity).

Research trends

Among the research oportunities we present next, we emphasize the need of: more efficient metamodels to handle more complex problems in general (addressed in Chapters 3, 4 and 5); procedures for selecting design points (addressed in Chapter 3); variance reduction techniques applicable to Metamodeling methods (addressed in Chapters 3 and 4); suitable metamodels for high-dimensional problems (addressed in Chapter 5).

According to Viana et al. (2014), Metamodeling and optimization have still a long way to go to become common tool in industry. Five challenges on future research in Metamodeling are posted: (i) the curse of dimensionality still exists as problems have just gotten larger; (ii) computational complexity still exists as problems have gotten more complex and/or we are trying to do more; (iii) there are still issues with numerical noise, which appear to be getting worse due to added computational complexity of many analysis and also poses additional challenge when performing model validation; (iv) the challenge of handling mixed discrete/continuous variables still exists, and may have gotten worse

due to the nature of problems now being investigate; and (v) validation of metamodels and the underlying model is as critical as before.

The following challenges in future research directions are identified in Tabatabaei et al. (2015): (i) handling noisy black-box function; (ii) capturing a nonconvex and disconnected Pareto frontier; (iii) handling a high number of objective and constraint functions as well as decision variables; (iv) providing the most preferred solution for a decision maker when solving computationally expensive multi-objective optimization problems; and (v) developing computationally expensive benchmark problems.

Quan et al. (2013) presents two possible avenues for future research in Kigring: (i) developing adaptive schemes that dynamically distributes the budget for each iteration; and (ii) studying in detail the convergence results of the Kriging algorithm. Similarly, Chen and Kim (2014) include in potential future research in Kriging: (i) a full theoretical treatment of stochastic kriging; (ii) selection of design points; (iii) effects of estimation spatial parameters, in particular when simulated responses are biased; and (iv) Metamodeling for steady-state simulations.

Regarding Metamodeling uncertainty, Zhang et al. (2013) proposes: (i) developing new sequential sampling techniques that consider the compound effect of Metamodeling and parametric uncertainty; and (ii) extending its proposed formulation to problems with uncertainty in noise variables rather than in design variables. Dimensionality limitations have been recently reported in Chang et al. (2013), Okobiah et al. (2014) and Hannah et al. (2014). Multi-objective and multi-constraints designs have been reported as a possible research direction in Okobiah et al. (2014), Chang et al. (2014), Chang (2015) and Zhou et al. (2015). The use of variance reduction techniques in future research has been cited in Chang et al. (2013), Hsieh et al. (2014) and Chang et al. (2014). Future research involving adaptations for solving quantile objective function have been reported in Chang et al. (2014), Chang (2015) and Kersaudy et al. (2015). The theoretical properties, in especial accuracy measurement, are pointed as potential research in Hernandez and Grover (2013) and Acar (2015).

Final remarks on metamodeling

Metamodeling is the most popular method to address stochastic optimization problems

most due to its user-friendly algorithms. Although demonstrates good results for low-dimension problems, such a method is not suitable for the high-dimensional ones. That is, an exponential computational time is added when dimension d is increased. Moreover, similarly to other approaches, metamodels usually have a heavy dependence on initial parameters, which are selected based on practitioner experience. Remarkably, there are no convergence proofs so far for it most used procedure (response surface methodology).

2.3 Sample Average Approximation

Sample average approximation (also referred to as the sample path optimization or the retrospective method) is a well-known method from stochastic programming for solving optimization problems under uncertainty via Monte Carlo simulation. In this technique, the objective function is approximated by a sample average estimate derived from simulation outputs. The approximated objective function, which is deterministic, can then be solved either by special purpose algorithms or by standard deterministic optimization techniques. By repeating the optimization with different outputs, feasible solutions and statistical estimates of their optimality gaps can be obtained (Evers et al. (2014)) .

The sample average approximation theoretical analysis appeared in the 1990s. Under relatively mild assumptions global and local minimizers, the values of the approximated function almost surely tend to the original values of the stochastic program as the sample size increases to infinity. The asymptotic distribution of minimizers, minimum values, and related quantities for the sample average problem are also known under additional assumptions (Royset and Szechtman (2013)).

The standard procedure, while effectively used in many applications, can lead to poor solution quality if the simulation budget is not sufficiently large. On the other hand, larger sample sizes become intractable due to the significant computational effort required. Moreover, it is important to note that the sample average approximation procedure selects the best performing sampled solution and discards the remaining outputs, which contain valuable information about the problem's uncertainty (Aydin and Murat (2013)). Such a method induces sampling error, caused by replacing an expectation by an ordinary sample average; as well as an optimization error due to approximating the solution of the

underlying sample average problem (Royset and Szechtman (2013)).

Therefore, the sample average approximation method could benefit from variance reduction techniques while performing Monte Carlo simulation mainly by using the information of discarded outputs. We explore this possibility of performance gain in Chapter 4.

For a guide in SAA, see Kim et al. (2014). For more details, see Shapiro (2003), Shapiro et al. (2014) and Shapiro (2013).

2.3.1 Research Trends

Recent Developments in Sample Average Approximation

The combination of sample average approximation with other optimization techniques is the target of some interesting research in the last years. Examples of hybrid approaches are: Shapiro et al. (2013) proposed a multistage stochastic programming problems based on the stochastic dual dynamic programming where the true problem is approximated via sample average approximation. A hybrid method that combines the particle swarm (see Section 2.5, metaheuristics) and sample average approximation is proposed in Aydin and Murat (2013). In Ozdemir et al. (2013), a capacitated supply scenario is formulated as a network flow problem embedded in a stochastic optimization problem, which is solved through a sample average approximation method. In Huan and Marzouk (2014), a gradient-based stochastic optimization method combined with sample average approximation for design of experiments on a continuous decision variable space is developed.

Research opportunities

Based on the references herein, we highlight three opportunities in sample average approximation research: (i) the use of variance reduction techniques (addressed in Chapter 4), (i) taking correlation behavior into account (also addressed in Chapter 4), (iii) and multiperiod extensions (addressed in Chapter 5).

Regarding (i) variance reduction techniques, Huan and Marzouk (2014) remark potential future work as employing a common random approach to obtain a lower variance estimate of optimality gap. According to Long et al. (2015), a possible direction is to ex-

plore new hybrid efficient designs in order to enhance the performance of sample average approximation methods combined with Latin hypercube sampling.

Regarding (ii) taking correlation into account, (Benyoucef et al. (2013)) remark the need to consider correlation of demand in distribution network design, which could lead to more realistic but very complex joint facility location and supplier selection models based on sample average approximation. Similarly, Osmani and Zhang (2014) report potential future research including the development of sample average approximation to correlate biomass price with supply/demand level;

Regarding (iii) multiperiod extensions, we have as example: multiperiod network design and the convergence properties of the sample average approximation algorithm need to be investigated (Benyoucef et al. (2013)); multiperiod alternatives are also considered as potential research in Aydin and Murat (2013) under the context of sample average approximation combined with particle swarm; multiperiod extensions combined with the integer L-shaped method is considered as a research direction in Chen et al. (2015).

Final remarks on sample average approximation

Sample average approximation may be a good option for practitioners used to determinist optimization. This method uses brute force to approximate the objective function via Monte Carlo simulation and, consequentially, turning it into a deterministic problem. Then, sophisticated tools of mathematical programming may be applied. Because of its significant computational time required, the method is not suitable for small computational budget.

2.4 Ranking and Selection

Ranking and selection procedures aim to select the best decision variables from a set of competing ones. One important assumption is the possibility to simulate all solutions at least a few times. The search for the best solution is exhaustive and the central problem is controlling statistical selection error. A comprehensive review of procedures is available in Kim and Nelson (2006). More recently, an overview is presented in Fu (2014).

The methods are applied in problems with finite and relatively small number of so-

lutions. The "relative small" number is considered by literature up to 1,000 solutions (e.g., Fu (2014), Hong et al. (2015) and Kim and Nelson (2006)). However, depending on how much it takes to simulate an alternative, there are some recent reports of procedures solving practical problems with more than 20,000 feasible solutions (e.g., Luo et al. (2015)).

Typically, ranking and selection procedures require a normality assumption of output data distribution (e.g., Mattila and Virtanen (2015), Hong et al. (2015), Xiao et al. (2015) and Diaz et al. (2016)). General output distributions, on the other hand, acknowledge only a few procedures (e.g.,Hunter and Pasupathy (2013), Lee and Nelson (2014) and Pasupathy et al. (2014)).

Classical results in ranking and selection procedures focuses on asymptotic convergence properties of the estimated best system (e.g., Wang and Kim (2013), Hunter and Pasupathy (2013), Lee and Nelson (2014), Pasupathy et al. (2014), Barut and Powell (2014), Xiao et al. (2014), Cheng et al. (2015), Xiao et al. (2015)). Regarding finite-time convergence properties of procedures, relatively little has been written (e.g., Andradottir and Kim (2010), Batur and Kim (2010), Healey et al. (2013) and Healey et al. (2014)).

2.4.1 Ranking and Selection Methods

Prior research on ranking and selection may be classified under one of four categories: the indifference-zone procedure; the value information procedure; the optimal computing budget allocation; and the large-deviations formulation. Similar grouping can be found in Hunter and Pasupathy (2013), Frazier (2014), Xiao et al. (2014) and Xiao et al. (2015). Basic procedure examples of each of these categories can be found in Chau et al. (2014).

The indifference-zone procedure was first formulated by Bechhofer (1954). In this procedure, a difference between designs is considered to be significant if it is larger than a specified indifference-zone parameter. The probability of correct selection guarantee is with respect to the probability of selecting the true best, subject to the condition that the mean of the true best is better than the mean of all of the other alternatives by at least the indifference-zone parameter (Chen et al. (2014)). Recent research on indifference-zone procedures includes: finding the best decision variables under a primary performance measure, while also satisfying stochastic constraints on secondary performance measures

(e.g., Healey et al. (2013), Healey et al. (2014), Hong et al. (2015) and Healey et al. (2015)); using hybrid approaches to solve particular problems (e.g., Tsai and Zheng (2013) and Diaz et al. (2016)); adapting of ranking and selection procedures to a high-performance (parallel) computing setting (e.g., Ni et al. (2013) and Ni et al. (2014)); tightening bounds on probability of correct selection (e.g., Wang and Kim (2013) and Frazier (2014)); and general-purpose ranking and selection regarding the system performance measure and assumed output distribution (e.g., Lee and Nelson (2014)).

The value information procedure was first proposed by Gupta and Miescke (1996) and subsequently developed by Frazier et al. (2008). It uses the Bayesian posterior distribution to describe the evidence of correct selection, and allocates further replications by maximizing the value information. Recent research on value information procedure includes: combining such a method with response surface methodology to improve efficiency (Barut and Powell (2014) and Cheng et al. (2015)); adapting of ranking and selection procedures to a high-performance (parallel) computing setting (e.g., Kaminski and Szufel (2014)); determining upper bounds of best function value (e.g., Xie and Frazier (2013)); and improving learning probabilities (Kaminski (2015)).

The optimal computing budget allocation, proposed by He et al. (2007), focuses on the efficiency of simulation by intelligently allocating further replications based on mean and variance. It aims to maximize the lower bound of correct selection probability (Xiao et al. (2014)). The procedures are often easy to apply and have a good empirical performance (Frazier (2014)). Recent research on optimal computing budget allocation includes: combining such a method with response surface methodology to improve efficiency (e.g., Brantley et al. (2014) and Xiao et al. (2015)); combining optimal computing budget allocation with genetic algorithm to improve efficiency (e.g., Xiao and Lee (2014)); finding the best decision variables with multiple objective functions through the usage of weights (e.g., Mattila and Virtanen (2015)); adapting the procedure to apply on microgrid problems (Bastani et al. (2014)); improving efficiency to solve complete ranking problems (Xiao et al. (2014)).

The large-deviations approach, formulated by Glynn and Juneja (2004), provides an asymptotically optimal sample allocation in the context of general light-tailed distributions (Hunter and Pasupathy (2013)). That is, the procedures do not request normal output

distribution. They are designed to maximize the probability of correct selection in an asymptotic sense as the sample size grows large (Frazier (2014)). Examples of recent research on large-deviations approach are Hunter and Pasupathy (2013) and Pasupathy et al. (2014).

2.4.2 Research Trends

There are several potential avenues of research on ranking and selection procedure reported in the references herein. For example, the research in Ni et al. (2013) includes matching problem features to efficient parallel algorithm designs and creating new algorithms explicitly designed to exploit parallel platforms. Additional improvements in efficiency of the ranking and selection algorithm are also possible by employing variance reduction techniques (Tsai and Zheng (2013)), which we address in Chapter 4. Tsai and Zheng (2013) remark as possible extension the development of more efficient ranking and selection procedures to determine the feasibility of candidate solution to reduce the sampling cost.

Better stopping rule that consider the tradeoff between optimality of the obtained solution and the required sample size is suggested as development opportunity on ranking and selection procedures in Tsai and Zheng (2013). The need of comparison sampling strategies to other optimal computing budget allocation-based methods is remarked in Healey et al. (2013).

Research trend on hybrid methods include: integrating the proposed ranking and selection method with some multi-dimensional search one (which we address in Chapter 5) as the stochastic trust region gradient-free method (Brantley et al. (2014)). Ranking and selection procedures that are able to deal with higher dimensional problems are reported as promising research in Bastani et al. (2014) and Cheng et al. (2015). Rigorous theoretical analysis of procedures as complementation to numerical comparison is pointed out as literature gap in Kaminski (2015).

Final remarks on ranking and selection

The main limitation of ranking and selection is the need to sample at least a few times each feasible solution. Therefore, recent research on this topic include how to use computational

budget in a more efficient way, and how to address problems with greater number of solutions. Because ranking and selection treats solutions as categorical, it may be one interesting method for problems with such a type of decision variables.

2.5 Metaheuristics

When the problem complexity is high, such as NP-hard problems, it is generally useful to apply metaheuristics methods because of their ability to support managers in decision-making with approximate solutions to complex problems in a quick way. Metaheuristic methods also produce quality solutions in the multi-objective context, but this is usually at the expense of longer computation times (Banos et al. (2013)).

Metaheuristic are the most commonly used methods in simulation software. When combining them with simulation models, the latter can be seen as a black box, i.e., some decision variables are defined in the black box. Then the simulation models provide some observations or outputs, which can be used to guide the search process (Wang and Shi (2013)).

Recent studies have demonstrated that hybrid metaheuristics work better than individual one for solving nonlinear models (Diabat (2014)). In the particular case of this thesis, we propose a hybrid formulation that combines ametaheuristic method with variance reduction techniques. We introduce our hybrid stochastic optimization method in Chapter 4.

2.5.1 Metaheuristic Methods

There is a vast of approaches based on deterministic methods that falls under the umbrella of metaheuristics, including: simulated annealing, genetic algorithms, tabu search, particle swarm, pattern search, Nelder-mead simplex (or downhill simplex), ant colony optimization, nested partitions, stochastic branch-and-bound methods, adaptive random search, controlled random search, differential evolution, coordinate search, scatter search and harmony search algorithm. For more regarding metaheuristics, see Fu (2014) and the references therein. Next, the two most commonly used algorithms are presented: simulated annealing and genetic algorithm.

Simulated annealing

With the first archaeological records dating back more than 6000 years, thermal annealing is likely to be the oldest optimization method in human history. First heating a material and then letting it cool down slowly can relieve internal stresses and allow the material to achieve a lower-energy state (Heim et al. (2015)). Inspired by thermal annealing, the simulated annealing algorithm dates back to the pioneering work by Metropolis et al. (1953). Since then, a large literature has appeared on simulated annealing, including important work by Kirkpatrick et al. (1983), Mitra et al. (1986), Hajek (1988), and others (Andradottir (2014)). An overview of simulated annealing can be found in Andradottir (2014). For a recent guide, see Yang (2014).

As described by (Wang and Shi (2013)), simulated annealing can be seen as a global optimization method based on the simulation of the physical annealing process to solve combinatorial optimization problems. The search process moves from one solution to the next until the terminating condition is satisfied. To avoid local optimum, a inferior solution is accepted with a probability. That is, for each iteration, the probability that a solution is accepted follows an exponential distribution. If it is inferior, it may be accepted or rejected according to a probability that is inversely proportional to the difference between the performance values of the two solutions, and proportional to the current temperature. Even though simulated annealing is commonly used to solve the deterministic optimization problems, there are numerous studies about its applications to stochastic optimization ones.

Implementation of simulated annealing procedures requires choosing parameters such as the initial and final temperatures, the rate of cooling, and number of function evaluations at each temperature. Implementing a simulated annealing procedure is an easy task and it remains a popular technique used by several commercial simulation optimization packages (Amaran et al. (2014)). It is important to remark that the size of its initial population must be large, which means the time to calculate the efficiency of every solution is also very long (Zhang et al. (2016)).

There is a vast literature on simulated annealing. Some recent research focuses on solving complex problems through adaptations and combinations of the simulated annealing

algorithm. For example, in Zhang et al. (2016) simulated annealing was combined with the Rosen projection method to optimize the parameters of an heliostat field. An algorithm that couples metamodeling procedures with evolutionary search, simulated annealing and elder-mean simplex is introduced in Tsoukalas et al. (2016). In Diabat (2014), a hybrid genetic and simulated annealing algorithm is proposed to address the vendor managed inventory issue, which has a nonlinear and non-convex objective function.

Genetic algorithm

As described in Amaran et al. (2014), genetic algorithms (Whitley (1994) and Reeves (1997)) use concepts of mutation and selection from theory of evolution. In general, the algorithm creates a population of strings and each of these strings are called chromosomes. Each of these chromosome strings is basically a vector of decision variables in the feasible space. New chromosomes are created by using selection, mutation and crossover functions. The selection process is guided by evaluating the objective function (or system performance) of each chromosome and selecting the chromosomes according to their function values. Additional chromosomes are then generated using crossover and mutation functions. The cross over and mutation functions ensures that a diversity of solutions is maintained.

The genetic algorithm is very popular, known to have the ability of generating good solutions when the feasible space is very large. The other attractive feature is that it is simple to code (Fu (2014)). Moreover, the method is used in several commercial simulation optimization software packages (Amaran et al. (2014)). The main difference between the genetic algorithm and other metaheuristics is that a population of solutions, rather than a single one, is manipulated (Wang and Shi (2013)).

There has been recently a great number of research combining genetic algorithms with other methods in order to solve complex problems. For example, Huajun et al. (2016) proposed a method that combines generic algorithm with simultaneous perturbation stochastic approximation to solve stochastic optimization problem with linear constraints. Such an hybrid method uses the genetic algorithm to search for optimum over the whole feasible region, and simultaneous perturbation stochastic approximation to search at local region.

An hybrid approach combining genetic algorithms and artificial neural network is proposed in Chandwani et al. (2014). In order to solve the multi-objective model, the non-dominant sorting of genetic algorithm is employed by Rajabi-Bahaabadi et al. (2015) and its parameters are tuned by the Taguchi method. Moreover, a dynamic n-point crossover (instead of the traditional one-point crossover) is developed to enhance the search capability of the the genetic algorithm.

2.5.2 Research Trends

Some of the interesting research trends reported in the references herein on metaheuristics tools are now presented. The inclusion of more realistic elements is remark as future work in Diabat (2014), Rajabi-Bahaabadi et al. (2015), Li and Demeulemeester (2016), Nogueira et al. (2016) and Yang et al. (2016). Developing a multi-objective of the proposed metaheuristics algorithm is reported as interesting future work in Aydemir-Karadag and Turkbey (2013), Rajabi-Bahaabadi et al. (2015) and Tsoukalas et al. (2016).

Because of lack of a similar problem in literature, the results in Li and Demeulemeester (2016) must be used as a benchmark for future similar studies. Taking spatial correlation among the variable distributions is pointed out as future work in Rajabi-Bahaabadi et al. (2015). The extension of the proposed metaheuristics to a parallel computing approach is remarked as promising future research to speed up the solution process and improve solution quality.

Final remarks on metaheuristics

There is a vast number of methods under the umbrella of metaheuristics, which have been used to deal with high complexity problems, such as NP-hard problems. There is little probabilistic or statistical consideration incorporated in these methods. Usually, the algorithms are not easy to understand, require longer computational budget and provide no performance guarantees.

2.6 Final Discussion

We close this Chapter with a discussion on key opportunities of research on stochastic optimization. The gaps in literature that we present next are general in the sense that they embrace most stochastic optimization categories. To position this thesis, we remark the connections between each Chapter and the associated research opportunity we are investigating.

- Using Paul Glasserman words in Glasserman (2004), it is easier to survey the topic of stochastic optimization than to answer the question that brings a reader to such a survey: "Which technique should I use?" That is rarely a simple answer to this question. Therefore, it strengthens the need to establish template problems in which the comparison between different algorithms and approaches can be foment. Many factors have direct impact on the choice of the stochastic optimization technique, including: the behavior of objective functions; the type of variance across control variables; the presence of correlation among control variables; the source of data, that it, from direct experiments or from computational simulations; the amount of computational budget; problem dimensionality; discrete or continuous control variables; numerical or categorical control variables. Building such templates is not an easy task, but a relevant one to guide effort on promising directions.

The numerical problems we use in Chapters 3 and 4 to analyze and illustrate the proposed formulations have been also utilized in other research. We referred to them as "templates". These templates allow us to conduct a fair comparison of our research to others. Similarly, we make use of performance measures proposed by other investigations in stochastic optimization. We make specific reference to them when appropriate.

- Variance reduction techniques are frequently used in problems where Monte Carlo simulation are applied. It is interesting to note that there are only a few stochastic optimization studies making use/analysis of such methods, although many of them have Monte Carlo simulations embedded in their algorithms. Exploiting specific features of algorithms or building a generic procedure to reduce variance of simulation may be one latent potential avenue for stochastic optimization research.

Variance reduction techniques play special role in this thesis. In Chapter 3, we propose an efficient metamodeling formulation based on control variates technique. In Chapter 4, we propose a hybrid method that combines a random search algorithm with a control variance technique.

- Most current research on stochastic optimization have based the quality and efficiency of the underlying approach on numerical results and limited comparisons to other methods. Quality measures are usually based on expected value. It could be also interesting to have alternatives. For example, evaluating the risk associated to the best solution. Moreover, there is a lack of research on analytical results that could improve the reliability on these methods.

Although finite-time properties are difficult to characterize, they provide important analyzes for the development or more efficient/sophisticated stochastic optimization methods. In Chapter 5, we use finite time theory to analyze the effects of high-dimensional space in the performance of optimization algorithms.

- Many recent studies have resorted to one common approach: ensemble/hybrid algorithms. They have demonstrated that ensemble/hybrid methods work better than individual ones for solving more complex problems. Algorithms with strong global search ability have been combined to the ones with local search ability (e.g., genetic algorithm and simulated annealing). To increase both fitting and predictive capacity of response surface models, several Metamodels have been combined. Regarding techniques that deal with noisy gradient estimates, for example, there has been proposed the use of finite-difference gradient to guide Simultaneous Perturbation Stochastic Approximation method. Sample average approximation have been combined to metamodels, metamodels have been combined to metaheuristics, metaheuristics have been combined to ranking and selection. Possibilities seem to be infinity.

In Chapter 4, we propose a hybrid formulation that combines a metaheuristic and variance reduction techniques. We remark that our formulation is generic in the sense that it can

easily be extended to a larger set of stochastic optimization methods, such as stochastic approximation, metamodeling, sample average approximation and ranking and selection.

- There is a vast demand on the development of stochastic optimization algorithms that can handle large-scale and more complex problems, which naturally continue to arise in real applications. Some generic examples are: taking into account multi-objectives; developing multi-period models, modeling complex behavior and/or elements; and adding stochastic constraints to the model.

Chapter 5 is dedicated to a deeper understanding of the high-dimensional space and its implications to the performance of stochastic optimization methods. One of the specific contributions of this Chapter is to provide solid insights to guide the design of more sophisticated stochastic optimization methods that can efficiently handle the dimensionality increase that rises from real applications.
Complementary suggestions for future research can be found in Wang and Shi (2013), Amaran et al. (2014) and Homem-de Mello and Bayraksan (2014). A recent discussion on the state of art in stochastic optimization can be found in Fu et al. (2015).

Chapter 3

Metamodeling via Control Variates

Among techniques that combine simulation and optimization, the large most used is response surface methodology (a.k.a. lower-order regression method). This method falls under the umbrella of Metamodeling, a class of stochastic optimization in which the main characteristic is building an analytical approximation of the stochastic objective function. Despite being very popular and user friendly, response surface methodology has well-known limitations and drawbacks. For instance, it is suitable to deal with low-dimensional problems and smooth objective functions. However, the method' becomes intractable in higher-dimensional problems and/or with highly nonlinear responses.

Recently, kriging (also known as the Gaussian process) and its variants such as stochastic kriging (see Staum (2009)) have draw attention in stochastic optimization research. Among the research that shows kriging performance overtake response surface methodology are Staum (2009), Li et al. (2010), Qu and Fu (2012), Elsayed and Lacor (2014) and Chen and Kim (2014).

Although its recent popularity for stochastic optimization applications, kriging variants have important limitations and require significant prior knowledge to be used in real applications. In order to apply kriging techniques, one must choose to use or not a model trend (i.e., to use a linear combination of the components of a known function, likewise response surface metamodeling) based on prior knowledge about the underlying prob-

lem. Also, kriging modeling requires the choice of a spatial correlation function, which includes the Gaussian, exponential, generalized exponential, linear, spherical, and cubic spline functions.

Moreover, once the structure of the Gaussian random field of kriging method is specified, one must determine parameters of the correlation function. This step requires a nonlinear optimization, in which one of the main inputs is a starting point. A bad choice of parameter starting point can lead to a poor model. Therefore, there are many sources of practitioner's choice, and the possibility of model misspecification (i.e., failure of assumptions to describe the data well) must not be neglected.

Variance reduction techniques are frequently used in problems where Monte Carlo simulation is applied. It is interesting to note that there are only a few stochastic optimization studies that use/analyzes such methods, although many of them have Monte Carlo simulation embedded in their algorithms. Exploiting specific features of algorithms or building a generic procedure to reduce variance of simulation has been frequently pointed out as one potential avenue for stochastic optimization research (e.g., Chang et al. (2013), Tsai and Zheng (2013), Wang and Shi (2013), Hsieh et al. (2014), Amaran et al. (2014) and Fu (2014)).

The main contribution of this Chapter is the proposal of a metamodeling method based on a variance reduction technique. Our goal is to use the information of samples at design to estimate at function value at prediction points via CV technique. Control variates have proven useful in a broadly applications of Monte Carlo simulation, as it has been effective is reduction the variance of estimates. It takes advantage of the sampled error between outputs and known means of a simulation system to correct the outputs of other one for which means are unknown and must be estimated (for details, see Glasserman (2004)).

We build on the work Borogovac and Vakili (2008), and Rosenbaum and Staum (2016). In our approach, metamodeling is seen as a control variable estimation problem, in which the same technique of response approximation is performed on a set of multiple decision variables (i.e., on a set of design points). In such situations, as argued in Borogovac (2009), even a large prior investment of computational effort on estimation at one or more control variable values is justified if it makes estimation at predict points less costly or time-consuming, in which "*the idea is to computationally learn to be computationally more*

efficient".

Our approach is similar to variable-fidelity metamodeling (e.g., Zadeh et al. (2009), Zheng et al. (2014), Zhou et al. (2015) and Colosimo et al. (2015)), where the modeling process is enhanced through the incorporation of knowledge (Zadeh et al. (2009)). The model is hierarchical, in the sense that one set of data (the experiments) is considered to be more reliable and it is labeled as high-fidelity, and the other set (the simulations) is labeled as low-fidelity. We propose to use a small set of design points (high-fidelity data) as controls to better estimate the underlying stochastic function at a larger set of points (low-fidelity data).

This Chapter is structured as follows. First, we review key elements of classical CV technique and its variants in Section 3.1. In Section 3.2 we formulate the design of our metamodel and evaluate its performance. Section 3.3 considers the effects of multicollinearity on a CV metamodel with multiple controls. An iterative allocation rule is presented in Section 3.4. A final discussion and directions for future research are provided in Section 3.5.

3.1 Preliminaries

Now, we introduce the main properties of classical control variates (Section 3.1.1) and three of its variants: (i) biased control variates (Section 3.1.2), (ii) estimated control variates (Section 3.1.3), and (iii) database control variates (Section 3.1.4). The following reviews are used as basis: Schmeiser et al. (2001), Glasserman (2004), Zhao et al. (2007), Borogovac and Vakili (2008), Borogovac (2009) and Pasupathy et al. (2012). We make reference to these reviews when appropriate.

3.1.1 Classical Control Variates

The CV technique (see Glasserman (2004) for details) exploits the error information of a tractable simulation model to adjusts the outputs of an intractable model. We describe the classic CV assuming, for simplicity, that a single control Z is used. Let $w_1, ..., w_n$ be some noise. Let $\boldsymbol{Y} = \{Y(w_1), ..., Y(w_n)\}$ be a vector with the output from n replications of a simulation model, and $\bar{Y}$ be its ordinary sample average. Suppose that the objective

is to estimate E[Y]. Suppose also that we can generate Z along with the same set of noise ($\boldsymbol{Z} = \{Z(w_1), ..., Z(w_n)\}$, $\bar{Z}$ be its ordinary sample average), and that the expectation E[Z] is known. Then, for any fixed β, we can compute the estimator

$$\hat{Y}(\beta) = \bar{Y} - \beta(\bar{Z} - \mathrm{E}[Z]). \tag{3.1}$$

This is the CV estimator, in which the observed error $Z(w_i) - \mathrm{E}[Z]$ serves as a control in estimating E[Y]. Each $\hat{Y}(w_i, \beta)$ has variance

$$\begin{aligned} \mathrm{Var}[Y(w_i, \beta)] &= \mathrm{Var}\left[\mathrm{Y(w_i)} - \beta(\mathrm{Z(w_i)} - \mathrm{E[Z]})\right] \\ &= \mathrm{Var}[Y] + \beta^2 \mathrm{Var}[Z] - 2\beta \mathrm{Cov}[Z, Y]. \end{aligned}$$

The optimal coefficient β^* that minimizes the variance (3.2) is given by

$$\beta^* = \frac{\mathrm{Cov}[Z, Y]}{\mathrm{Var}[Z]}. \tag{3.2}$$

Hence, the variance of the CV estimator given by β^* is

$$\mathrm{Var}[\hat{Y}] = \mathrm{Var}[Y](1 - \mathrm{Corr}^2[Z, Y]) \tag{3.3}$$

It is important to note that the effectiveness of a CV, as measured by the variance reduction rate $1 - \mathrm{Corr}^2[Z, Y]$, is determined by the magnitude of correlation between the variable of interest Y and the control Z. In practice, typically the measures Var[Y] and Cov[Z, Y] are known. In spite of this, an estimate of β^* has proven to be advantageous. As reported in Glasserman (2004), a frequently accepted estimator for this case is

$$\hat{\beta} = \frac{\mathrm{Cov}[\boldsymbol{Z}, \boldsymbol{Y}]}{\mathrm{Var}[\boldsymbol{Z}]}. \tag{3.4}$$

A discussion of classical CV can be found in Lavenberg and Welch (1981), Nelson (1990) and Glasserman (2004), and includes estimation of the minimal-variance coefficient β^*, its link between linear regression, and extensions to higher dimensions. It is important to observe that, in order to apply the classical CV, the control mean E[Z] must be known.

At the same time, there are not many variables with known means that are informative about variable Y (i.e., are highly correlated to Y) because usually they are of similar type and complexity as Y. Therefore, the assumption of known mean is often difficult to satisfy and hinders the applicability of the technique. There are three approaches for relaxing such fundamental requirement: (i) biased control variates (see Schmeiser et al. (2001) for details); (ii) estimated control variates (see Pasupathy et al. (2012) for details); and (iii) database control variates (see Zhao et al. (2007) for details). The key features of these techniques will be now introduced.

3.1.2 Biased Control Variates

In the biased control variates approach (Schmeiser et al. (2001)), the control mean $\mathrm{E}[Z]$ is replaced by an approximation $\hat{\mu}_B$. It can be yield from a numeric or closed-form analysis of an *approximated model* of the true model of variable Z. As remarked in Borogovac and Vakili (2008), although the current method relax the assumption of known mean, it assumes the existence of an approximated simulation model for which means can be computed analytically. That is, a new strong assumption is required.

It is important to note that, in the biased CV approach, the expectation of the approximated control mean does not equal the original mean (i.e., $\mathrm{E}[\mu_B] \neq \mathrm{E}[Z]$). This fact adds a bias to the controlled estimators proportional to the *approximation error* $\hat{\mu}_B - \mathrm{E}[Z]$. Since the approximation $\hat{\mu}_B$ is exogenous to sampled data (i.e., computed prior to the joint simulation of (Y, Z)), obtaining more samples (i.e., increasing n) to reduce the variance does not decrease the approximation error. According to Borogovac (2009), the biased CV approach is beneficial only if the bias raised by the approximated model is sufficiently small compared to the gains in variance reduction.

The biased CV coefficient is given by

$$\hat{Y}_B(\beta_B) = \bar{Y} - \beta_B(\bar{Z} - \hat{\mu}_B).$$

$\hat{Y}$ is a biased estimator of $\mathrm{E}[Y]$ with bias $\beta_B(\hat{\mu}_B - \mathrm{E}[Y])$ and the mean squared error

(MSE) of the estimator is

$$\text{MSE}(\hat{Y}_B(\beta_B)) = \text{Var}[\hat{Y}_B(\beta_B)] + [\beta_B(\hat{\mu}_B - \text{E}[Y])]^2 . \tag{3.5}$$

The optimal coefficient β_B^* is the one that minimizes the mean squared error in (3.5) and, as in classical CV, needs to be estimated from sampled data. The performance of β_B^* and the classical $\hat{\beta}$ (3.4) is compared in Schmeiser et al. (2001). It is reported that coefficient $\hat{\beta}$ performs better than β_B^* if the bias raised from the approximation error is small compared to $\text{Var}[X]/n$.

3.1.3 Estimated Control Variates

Differently from biased CV, the true simulation model of Z is used to estimate the control mean $\text{E}[Z]$ in the estimated CV approach. It begins with a set-up stage that estimates the control mean. That is, the control mean estimator $\hat{\mu}_E$ is estimated prior to the joint simulation of (Y, Z), such as in biased CV.

In the set-up stage, a set $\{w_1, ..., w_N\}$ of noise is generated according to the underlying probability measure. Note that the sample size of the set-up stage is different from the estimation stage, with $N >> n$. In this case, the use of effort in estimating $\hat{\mu}_E$ can only be justified if the resulting variance reduction is large enough. Let $Z(w_1), ..., Z(w_N)$ be the underlying output of variable Z. Then, the estimate of $\text{E}[Z]$ is the following ordinary sample average

$$\hat{\mu}_E = \frac{1}{N} \sum_{i=1}^{N} Z(w_i).$$

The *estimation error* $\hat{\mu}_E - \text{E}[Z]$ is, as in biased CV, fixed and unknown. However, in estimated CV $\text{E}[\mu_E] = \text{E}[Z]$.

The estimation stage of estimated CV, likewise in classical and biased versions, is performed with sample size n. The ECV coefficient is defined as

$$\hat{Y}_E(\beta_E) = \bar{Y} - \beta_E(\bar{Z} - \hat{\mu}_E). \tag{3.6}$$

The mean squared error of the estimator is

$$\mathrm{MSE}(\hat{Y}_E(\beta_E)) = \frac{\mathrm{Var}[Y]}{n} + \beta_E^2 \frac{\mathrm{Var}[Z]}{2} - 2\beta_E \frac{\mathrm{Cov}[Z,Y]}{n} + \beta_E^2 \frac{\mathrm{Var}[Z]}{N},$$

and the minimizer coefficient is given by

$$\beta_E^* = \frac{\mathrm{Cov}[Z,Y]}{\mathrm{Var}[Z]} \left(\frac{N}{N+n} \right).$$

The MSE of the estimator with β_E^* is of the form

$$\mathrm{MSE}(\hat{Y}_E(\beta_E^*)) = \frac{\mathrm{Var}[Y]}{n} \left[1 - \mathrm{Corr}^2[Z,Y] \left(\frac{N}{N+n} \right) \right].$$

Hence, variance expression is similar to the classical one (3.3), except of the correlation loss factor $N/(N+n)$. Likewise, the expression for the MSE using the classical coefficient β (3.2) is as follows:

$$\mathrm{MSE}(\hat{Y}_E(\beta)) = \frac{\mathrm{Var}[Y]}{n} \left[1 - \mathrm{Corr}^2[Z,Y] \left(\frac{N-n}{N} \right) \right].$$

This fact induces an acceptable use of the estimated coefficient from classical CV (3.4) instead of β_E in (3.6).

3.1.4 Database Control Variates

In the database CV approach, the problem of estimating control mean is replaced by a transformed one, for which we may calculate control means exactly, and it is rich enough to closely approximate the original problem. This is accomplished by replacing the original probability measure P that rules all simulation variables by an approximating measure $P_{\mathtt{W}}$ on which integration is tractable. The procedure that allows the chance in probability measure is now introduced.

The main difference between estimated CV and database is that, for the latter, the noise samples of the set-up stage (i.e., the set $\{w_1, ..., w_N\}$) are stored into a *database* `W` to enable its retrieval in the estimation stage (estimating $\mathrm{E}[Y]$). The output $Z(w_1), ..., Z(w_N)$ may also be stored in `Z`, which is advantageous in simulations with a high cost per sample.

The estimate of $\mathrm{E}[Z]$ in the set-up stage is equal to the one in estimated CV (i.e., $\hat{\mu}_D = \hat{\mu}_E$):

$$\hat{\mu}_D = \frac{1}{N}\sum_{i=1}^{N} Z(w_i).$$

Clearly, the estimation error $\hat{\mu}_D - \mathrm{E}[Z]$ has the same properties of the one in estimated CV. As remarked in Borogovac and Vakili (2008), $\hat{\mu}_D$ can be viewed as the expected value of random variable Z restricted to the probability space $\mathtt{W}$ with respect to a uniform measure $P_{\mathtt{W}}$ on a discrete probability space. Thus

$$\hat{\mu}_D = \mathrm{E}_{\mathtt{W}}[Z], \tag{3.7}$$

where $\mathrm{E}_{\mathtt{W}}$ denotes expectation with respect to $P_{\mathtt{W}}$.

For this reason, in the estimation stage, the underlying probability measure P is replaced by the uniform measure $P_{\mathtt{W}}$ with probability space $\mathtt{W}$, where

$$P_{\mathtt{W}}(w) = \frac{1}{N}, \qquad \forall\ w \in \mathtt{W}.$$

Therefore, let us denote the set of noise $\boldsymbol{W} = \{w_1, ..., w_n\}$ as the n uniformly selected elements from $\mathtt{W}$. Let us now retrieve from $\mathtt{Z}$ the correspondent vector $\boldsymbol{Z} = Z(w_1), ..., Z(w_n)$ and the ordinary sample average $\bar{Z}$. Accordingly, evaluate $\boldsymbol{Y} = Y(w_1), ..., Y(w_n)$ and compute the ordinary sample average $\bar{Y}$. Then, the DCV estimated coefficient is the CV one:

$$\hat{\beta} = \frac{\mathrm{Cov}[\boldsymbol{Z}, \boldsymbol{Y}]}{\mathrm{Var}[\boldsymbol{Z}]}.$$

The DCV estimator is given by:

$$\hat{Y}_D(\hat{\beta}) = \bar{Y} - \hat{\beta}(\bar{Z} - \hat{\mu}_D). \tag{3.8}$$

The mean squared error of the estimator using the classical one in (3.2) is given by:

$$\mathrm{MSE}(\hat{Y}_D(\beta)) = \frac{\mathrm{Var}[Y]}{n}\left(1 - \mathrm{Corr}^2[Z, Y]\right) + \frac{\mathrm{Var}[Y]}{N}, \tag{3.9}$$

which is the variance of the classical CV estimator plus a term $(\mathrm{Var}[Y]/N)$ that arise from

the error $\mathrm{E}_{\mathtt{W}}[Z] - \mathrm{E}[Z]$. Such as in estimated CV, we know that this error approximates one sample of the normal $\mathcal{N}(0, \mathrm{Var}[Y]/N)$.

3.2 The Control Variate Metamodel

In this Section, we propose a metamodel based on Control variates technique. In general, the CV metamodel has no requirement on the underlying function and little dependence on practitioner's choice in comparison to stochastic kriging. The formulation, which is presented in Section 3.2.1, is performed in two phases: a sampling procedure at the design points, and a sampling procedure at low-fidelity points. The performance measures and a thorough comparison to SK is presented in Section 3.2.2. Results show that overall performance of the CV metamodeling is notably better than the SK one.

3.2.1 Method Procedure

Stochastic optimization methods aim at finding a configuration or design that minimizes the following objective function

$$\min_{\boldsymbol{x} \in X} J(x) = \mathrm{E}[Y(\boldsymbol{x}, w)],$$

where x is a vector of dimension d and denotes the decision variables (also referred to as input variables or parameters) of the simulation model, Y is the model output, w is the noise and represents the sample path of the simulation model, and J is the objective function. In real applications, J is frequently of complex nature without explicitly known form. Stochastic optimization methods, including the metamodeling class, often serve the purpose of optimizing J very well. Our goal in this Chapter is to build a CV metamodel that efficiently approximates $\mathrm{E}[Y(x, w)]$ in a bounded space X.

Defining Design Points

Similarly to response surface methodology and SK, one must choose a set of *design points* $\tilde{X} = \{\tilde{x}_1, ..., \tilde{x}_K\}$ which constitutes the experiment design. In our metamodel, the design points (or high-fidelity points) play the role of control in the CV technique.

In both response surface methodology and SK, the total simulation budget (i.e., the total number of computational output) is totally allocated to design points. The amount of simulation budget that is allocated among these points can be optimized and is not the focus of our study (for example, see Quan et al. (2013) for optimal computation budget allocation in SK).

In our CV metamodel, the total budget is divided in two equal shares. One allocated in estimating Y at design points, and other allocated in a set of *low-fidelity points* (which will be soon introduced). Let N be the number of simulation output allocated at each design points, and thus $2NK$ be the total simulation budget.

Defining Low-Fidelity Points

Choose a set of low-fidelity points $X = \{x_1, ..., x_L\}$, where $L \gg K$. The simulation budget allocated at each low-fidelity is $n = \lfloor (NK)/L \rfloor$, with $n \ll N$.

Simulation at Design Points

The scheme version of the algorithm is given below. Note that we are using common random numbers to simulate Y at different points. In a few steps, the vector `W` will be used again to induce correlation.

Simulation Algorithm at Design Points

---

0 Generate a set $\{w_1, ..., w_N\}$ according to the underlying probability measure, and store into a vector `W` to enable retrieval.

1 For $k = 1, ..., K$

1.1 Perform a simulation with effort N at design point $\tilde{x}_k$, generating

$$\{Y(\tilde{x}_k, w_1), ..., Y(\tilde{x}_k, w_N)\}.$$

1.2 Compute the estimator

$$\hat{\mu}(\tilde{x}_k) = \frac{1}{N} \sum_{i=1}^{N} Y(\tilde{x}_k, w_i). \tag{3.10}$$

1.3 Now, compute a vector using the first n outputs $\tilde{\mathtt{Y}}_k = \{Y(\tilde{x}_k, w_1), ..., Y(\tilde{x}_k, w_n)\}$.

1.4 Compute the ordinary sample average

$$\bar{\tilde{Y}}_k = \frac{1}{n}\sum_{i=1}^{n} Y(\tilde{x}_k, w_i).$$

Simulation at Low-Fidelity Points

For each low-fidelity point, we apply the CV method choosing as control the design point that presents the higher correlation to the current low-fidelity point. It is important to note that only one at a time control is used to enhance estimation at each low-fidelity point. The use of multiple controls is further discussed in Section 3.3. As mentioned before, the first elements $\{w_1, ..., w_n\}$ are retrieved from the vector `W` in order to induce correlation between the CV (design points) and the low-fidelity points to enhance CV efficiency. A scheme version of the simulation step is given below.

Simulation Algorithm at Low-Fidelity Points

---

0 Retrieve from `W` the first n elements $\{w_1, ..., w_N\}$.

1 For $l = 1, ..., L$

1.1 Compute the following vector and sample average

$$\mathtt{Y}_l = \{Y(x_l, w_1), ..., Y(x_l, w_n)\} \qquad \text{and} \qquad \bar{Y}_l = \frac{1}{n}\sum_{i=1}^{n} Y(x_l, w_i).$$

1.2 Set

$$k^* = \arg\max_{k=1,...,K} \text{Corr}[\tilde{\mathtt{Y}}_k, \mathtt{Y}_l].$$

1.3 Compute the CV coefficient:

$$\hat{\beta} = \frac{\text{Cov}[\tilde{\mathtt{Y}}_{k^*}, \mathtt{Y}_l]}{\text{Var}[\tilde{\mathtt{Y}}_{k^*}]}. \tag{3.11}$$

1.4 The estimator of $\mathrm{E}[Y(x_l)]$ is given by

$$\hat{Y}(x_l) = \bar{Y}_l - \hat{\beta}\left(\bar{\tilde{Y}}_{k^*} - \hat{\mu}(\tilde{x}_{k^*})\right).$$

3.2.2 Test on Metamodeling Performance

The performance measures and simulation problems used here are based on Li et al. (2010), where stochastic kriging is extensively compared to other popular metamodeling techniques (artificial neural network, radial basis function, support vector regression, and multivariate adaptive regression splines). The motivation for using such a comparison framework lies on the need of template problems to facilitate and foment the understanding of different metamodeling algorithms, and guide the effort on subsequent research.

We utilize the four canonical determinist problems used in Li et al. (2010) as basis for constructing performance tests and comparisons. They are: (i) P1: 4 dimensional welded beam design (Rao (1996)); (ii) P2: 3 dimensional helical compression spring (Arora (1989)); (iii) P3: 2 dimensional sinusoidal function (Hussain et al. (2002)); and (iv) P4: 8 dimensional asymmetric function (Nicolai and Dekker (2009)). Theses functions are known to be complex with varying degrees of nonlinearity and dimensionality.

We add four different noisy terms onto each deterministic problem to produce four different stochastic problem types because the noise type of the underlying function (homogeneous / heterogeneous) and the noise size (large / small) are two important characteristics of stochastic behavior (see Li et al. (2010)). They are named as small+homogeneous noise (S+Ho), small+heterogeneous noise (S+He), large+homogeneous noise (L+Ho), and large+heterogeneous noise (L+He). Thus, there are a total of 16 stochastic problems.

Moreover, we evaluate two classical problems from finance (pricing an Asian call option), and from queue theory (M/M/1 queue problem). The motivation to evaluate the SK and CV methods in these two additional problems lies on the fact that all template problems in Li et al. (2010) have noise incorporated to a determinist function in an additive form. As a consequence, the correlation between outputs at different solution points is typically very high. On the other hand, correlation at the first two classical problems may

not be as high. We recall that a high magnitude of correlation between outputs is a key factor for the effectiveness of database CV (see equation (3.9)). Therefore, it is important to analyze the performance of CV metamodel in objective functions where noise is not incorporated in an additive form.

Next, we provide a succinct description of each determinist problem, and its noise properties can be found in Table 3.1. Likewise, we provide a description of the Asian call option and M/M/1 queue system.

Welded beam design

The welded beam design problem is taken from Rao (1996) and utilized in Li et al. (2010). A welded beam is designed for *minimum* cost while subjected to constraints on shear stress (τ), bending stress in the beam (σ), bucking load on the bar (P_c), end deflection on the beam (δ), and side constraints. There are four decision variables ($\boldsymbol{x} = (x_1, ..., x_4)^\top$). The problem can be represented as:

$\min_{\boldsymbol{x}\in X} J(\boldsymbol{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$

Subject to:

$\tau(\boldsymbol{x}) - \tau_{\max} \leq 0$

$\sigma(\boldsymbol{x}) - \sigma_{\max} \leq 0$

$0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5 \leq 0$

$\delta(\boldsymbol{x}) - \delta_{\max} \leq 0$

$P - P_c(\boldsymbol{x}) \leq 0$

$0.1 \leq x_1 \leq 1$

$2 \leq x_2 \leq 10$

$6 \leq x_3 \leq 8$

$0.3 \leq x_4 \leq 0.8$

where

$\tau(\boldsymbol{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}$

$\tau' = \frac{P}{\sqrt{2} x_1 x_2},\ \tau'' = \frac{MR}{Q}$

$M = P\left(L + \frac{x_2}{2}\right),\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$

$Q = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$
$\sigma(\boldsymbol{x}) = \frac{6PL}{x_3^2x_4}$
$\delta(\boldsymbol{x}) = \frac{4PL^3}{Ex_3^3x_4}$
$P_c = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$
$P = 6000\text{lb},\ L = 14\text{in},\ E = 3.0 \times 10^7\text{psi},\ G = 1.2 \times 10^7\text{psi}, \tau_{\max} = 1.36 \times 10^4\text{psi},$
$\sigma_{\max} = 3.0 \times 10^4\text{psi},\ \delta_{\max} = 0.25\text{in}$

The noise is incorporated in the objective function in an additive form, as follows:

$$J(\boldsymbol{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) + W,$$

where W is the stochastic component. Table 3.1 shows four different definitions of W and their corresponding problem types.

Helical compression spring

This problem is taken from Arora (1989). The objective is to *minimize* the weight of a tension/compression spring. The design constraints are on minimum deflection, shear stress, surge frequency, and outside diameter. The design variables are the wire diameter (x_1), the mean coil diameter (x_2), and the number of active coils (x_3). The problem can be represented as:

$\min_{\boldsymbol{x}\in X} J(\boldsymbol{x}) = (x_3 + 2)x_2x_1^2$

Subject to:

$1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$
$\frac{4x_2^2 - x_1x_2}{12566(x_1x_2^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$
$1 - \frac{140.45x_1}{x_2^3x_3} \leq 0$
$\frac{x_1+x_2}{1.5} - 1 \leq 0$

The noise is incorporated in the objective function in an additive form, as follows:

$$J(\boldsymbol{x}) = (x_3 + 2)x_2x_1^2 + W,$$

where W is the stochastic component. Table 3.1 shows four different definitions of W and their corresponding problem types.

Sinusoidal function

This problem is taken from Hussain et al. (2002). The design variables are x_1 and x_2, and the problem can be represented as:

$\min_{\boldsymbol{x} \in X} J(\boldsymbol{x}) = x_1 \sin(x_2) + x_2 \sin(x_1)$

Subject to:

$-2\pi \leq x_1 \leq 2\pi$

$-2\pi \leq x_2 \leq 2\pi,$

The noise is incorporated in the objective function in an additive form, as follows:

$$J(\boldsymbol{x}) = x_1 \sin(x_2) + x_2 \sin(x_1) + W,$$

where W is the stochastic component. Table 3.1 shows four different definitions of W and their corresponding problem types.

Asymmetric function

This problem is taken from Nicolai and Dekker (2009). The design variables are $x_1, ..., x_8$, and the problem can be represented as:

$\min_{\boldsymbol{x} \in X} J(\boldsymbol{x}) = \sum_{i=1}^{8} \left[2^{x_i - 4} + (6 - x_i)\right]$

Subject to:

$0 \leq x_i \leq 10, \quad i = 1, 2, ...8$

The noise is incorporated in the objective function in an additive form, as follows:

$$J(\boldsymbol{x}) = \sum_{i=1}^{8} \left[2^{x_i - 4} + (6 - x_i)\right] + W,$$

where W is the stochastic component. Table 3.1 shows four different definitions of W and

their corresponding problem types.

Asian Call Option

The details of this problem can be found in Glasserman (2004). An Asian option is an option on a time average of the underlying asset. Asian calls have payoffs $(\bar{S}-K)^+$, where the strike price K is constant.

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{m} S(t_i)$$

is the average price of the underlying asset over the discrete set of monitoring dates $0 < t_1 < ... < t_m = T$, with T the date at which the payoff is received.

$$S(t_{j+1}) = S(t_j)\exp\left[(\mu - \frac{1}{2}\sigma^2)(t_{j+1}-t_j) + \sigma\sqrt{t_{j+1}-t_j}Z_{j+1}\right],$$

is the price of the asset at time t_{j+1}, $Z_1, ..., Z_m$ are independent standard normal random variable. $S(0)$, T and K are given. The variables of interest are $x = (\mu, \sigma)$, where μ is the rate of change in the asset price, σ is the volatility of the asset price. The objective is to option price (function value) is

$$J(\mu, \sigma) = \exp(-\mu T)[\bar{S} - K]^+.$$

M/M/1 queue

This problem is taken from Staum (2009). Consider a M/M/1 queue with arrival rate 1 and service rate x. The steady-state time is positive with probability $1/x$ and, given that it is positive, is conditionally exponential with mean $1/(x-1)$. Its means is $y(x) = 1/(x(x-1))$. Each simulation run is initialized in steady-state (which avoids bias from the initial conditions) and simulates a fixed number of costumers. Its output is their average waiting time.

Performance measures

The computational efficiency is critical to define a metamodel's performance. In order to

Table 3.1: Noise definitions in different bed problems

Noise type	Weld beam	Compression	Sinusoidal	Asymmetric
S+Ho	$W \sim \mathcal{N}(0, 2)$	$W \sim \mathcal{N}(0, 0.08)$	$W \sim \mathcal{N}(0, 1)$	$W \sim \mathcal{N}(0, 7)$
S+He	$W \sim x_1 - \mathcal{N}(0, 2)$	$W \sim x_1 - \mathcal{N}(0, 0.08)$	$W \sim x_1 - \mathcal{N}(0, 1)$	$W \sim x_1 - \mathcal{N}(0, 7)$
L+Ho	$W \sim \mathcal{N}(0, 10)$	$W \sim \mathcal{N}(0, 0.4)$	$W \sim \mathcal{N}(0, 5)$	$W \sim \mathcal{N}(0, 35)$
L+He	$W \sim x_1 - \mathcal{N}(0, 10)$	$W \sim x_1 - \mathcal{N}(0, 0.4)$	$W \sim x_1 - \mathcal{N}(0, 5)$	$W \sim x_1 - \mathcal{N}(0, 35)$

compare the two approaches, both stochastic kriging and control variates Metamodeling are applied to each problem with the same computational budget. Each stochastic problem runs with a total computational budget of 100.000 simulation output. 25 replication runs are taken. 100 design points are used in SK, with 1.000 simulation outputs in each. In the CV method, 10 design points are used with 5.000 simulation outputs in each. Moreover, 1.000 low-fidelity points with 50 simulation outputs are used in the second step of the CV method. 1.000 prediction points are used in both methods. The results will formulate a group of quality measures for which we evaluate the accuracy, robustness and efficiency of the methods. The measures were proposed in Li et al. (2010).

The accuracy is intended to reflect the deviation between the metamodel output $\hat{Y}(x)$ and the expectation $\mathrm{E}[Y(x)]$. Both global and local accuracy are considered. The mean squared error (MSE) provides a general evaluation of the overall prediction accuracy, and is given as follows:

$$\mathrm{MSE} = \frac{1}{L} \sum_{l=1}^{L} \left(\hat{Y}(x_l) - \mathrm{E}[Y(x_l)] \right)^2, \tag{3.12}$$

where L is the number of prediction points. Figure 3.1 shows the MSE under the 16 different stochastic problem for both SK and CV metamodels. In 14 of the 16 problems it is clear that CV metamodel achieves the best global accuracy measured by the MSE. The only two exceptions are Large+Homogenous noise in P2 and P3, where the CV accuracy is near the SK one. It can be explained by arguing that the estimated expectation $\hat{\mu}$ in the CV method is direct dependent on the variance of the samples. Because the exceptions happen on a scenario with large noise, the estimator $\hat{\mu}$ delivers a poorer performance compared to the scenarios with small variance. In the scenario with Large+Heterogenous noise, the accuracy of CV method is slightly poorer than in Large+Homogenous. However, the performance of SK method decreases significantly in this particular scenario (L+He). A worse performance of SK at the scenario (L+He) is expected because this is the most

difficult case. It is important to observe that the CV metamodel's efficiency in reducing the variance at prediction points is remarkable, in particular at the latter scenario.

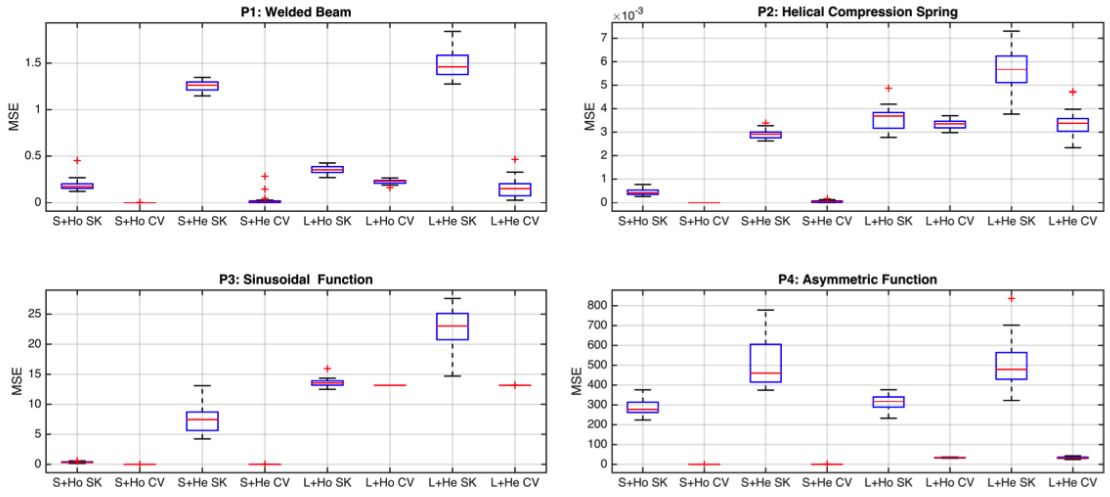

Figure 3.1: Mean Squared Error under different stochastic problems

The average absolute error (AAE) also assess the global accuracy, and is given as follows:

$$\text{AAE} = \frac{1}{L}\sum_{l=1}^{L} |\hat{Y}(x_l) - \text{E}[Y(x_l)]|$$

Figure 3.2 shows the AAE under the different stochastic problem of SK and CV metamodel results. Similarly to MSE measure, the global performance of the CV method measured by the average absolute error is clearly better than the SK performance under different problems and noise type. Two exceptions remain (L+Ho in P2 and P3). It is interesting to note that the MSE and AAE of SK method is more affected by the "topology" of the noise than by the "intensity" of the noise. That is, the MSE and AAE of SK in scenarios with Homogenous noise are small and similar, and the MSE and AAE in the scenarios with Heterogeneous noise are high and similar. On the other hand, the CV method is more affected by the "intensity" of the noise. The MSE and AAE of CV method are small and similar in scenarios with Small variance of noise, and high and similar in the scenarios with a large variance of noise.

The maximum absolute error (MAE) reflects the presence of poor prediction in local

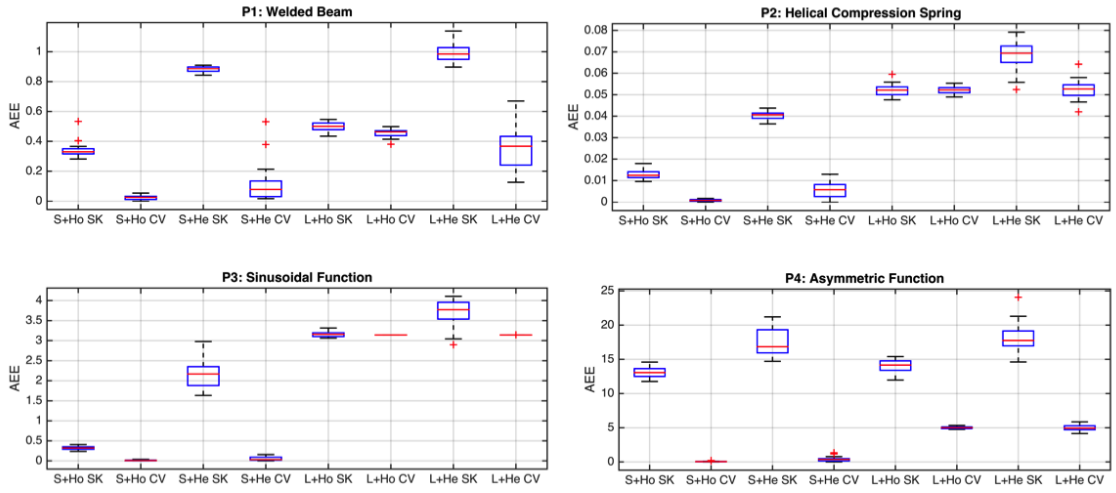

Figure 3.2: Average Absolute Error under different stochastic problems

areas, and is given as follows:

$$\text{MAE} = \max|\hat{Y}(x_l) - \text{E}[Y(x_l)]|$$

Figure 3.3 shows the MAE under the 16 different stochastic problems of SK and CV metamodel results. The worst estimate among prediction points of CV metamodel is significantly better than the SK one at all 16 problems. That is an important measure, and indicates that the CV metamodel has a better ability in consistently estimating the objective function among all prediction points. It is a direct effect of the gain in variance reduction provided by the database CV.

Robustness is another important indicator of performance as it represents each method's ability to consistently achieve similar accuracies at different replications. Robustness is defined as the standard deviation of the mean squared error in (3.12), and its formula is given as follows:

$$robustness = std.[MSE]$$

Figure 3.4 shows the standard deviation of MSE under the 16 different stochastic problems. In 15 of the 16 problems, CV metamodel exhibits a robustness performance better than SK metamodel. In particular, for P3 and P4. As expected, the robustness of CV method decreases with the intensity of noise variance, and the robustness of SK method decreases

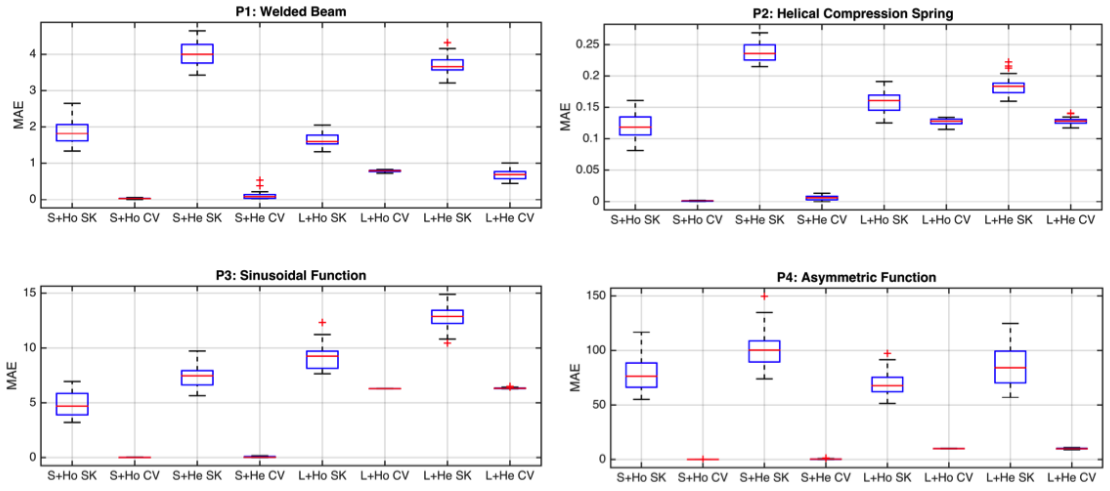

Figure 3.3: Maximum Absolute Error under different stochastic problems

with noise topology. It is worth noting that the robustness of CV is better than SK in L+Ho in P2 and P3, which indicates that CV overall performance in both accuracy and robustness is above SK overall perfomance.

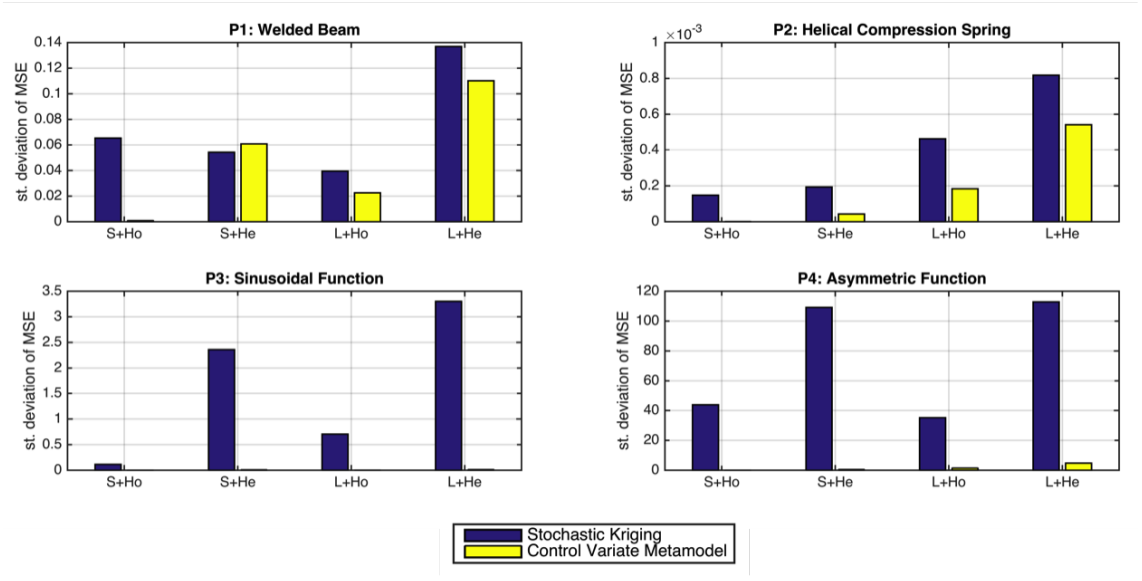

Figure 3.4: standard deviation of mean squared error under different stochastic problems

Path-Dependent Options

We illustrate the application of control variates metamodeling in a classical problem raised from finance engineering, which is the Asian call option problem. Because it is a path-

dependent problem, there is no analytical solution and one must resort to stochastic tools to estimate the option value at each decision variable. Its function value has an interesting characteristic regarding its variance. Decision variables are $x = (\mu, \sigma)$, where μ is the rate of change in the asset price, and σ is its volatility. Thus, the objective function has heterogeneous variance.

The importance of this canonical example is that, differently from previous deterministic template problems, the noise in the objective function is not in an additive form. In the current problem, the standard normal noise is multiplied by a volatility and time-passage term, and is embedded on an exponential function of the asset's price expression. The stochastic structure of this particular function has a direct and negative impact on the strength of correlation between design points and low-fidelity ones. We recall that one of the key elements that determined the efficiency of database CV is the correlation between control (design-points) and variable of interest (low-fidelity points).

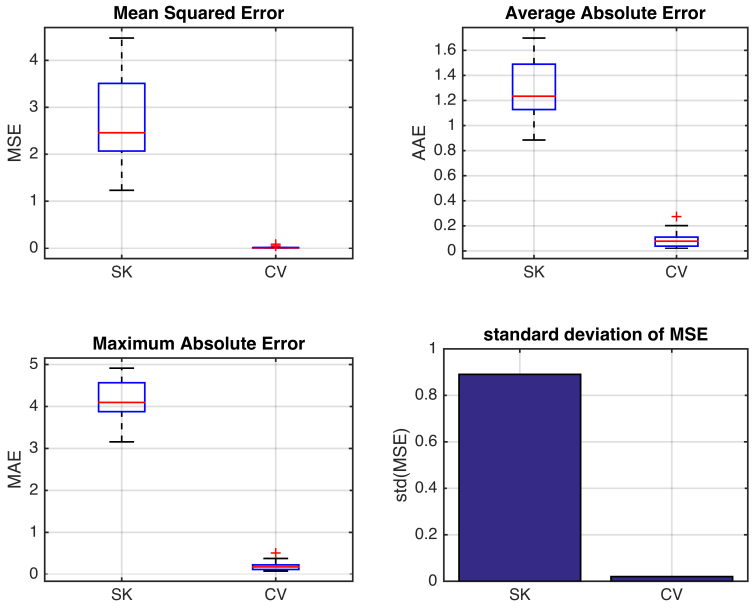

Figure 3.5: Performance measures of stochastic kriging, and control variates metamodeling in the Asian call option problem

Figure 3.5 shows the measure performance of the SK and CV metamodel. In this example, the initial asset price is $S(0) = 100$, the strike price is $K = 100$, $m = 30$ days. The design space of the experiment is $0.1 \leq \mu \leq 0.5$, and $0.05 \leq \sigma \leq 0.25$. For stochastic

kriging, 100 design points are used with 1,000 simulation outputs in each, and 1,000 predication points. For CV metamodel, 10 design points are used with 5,000 simulation outputs in each, and 1,000 low-fidelity points are used with 50 simulation outputs in each. Thus both metamodels have a total simulation budget of 100,000 outputs. 25 replications of each metamodel are used to construct the boxplots of MSE, AAE and MAE, and to compute the standard deviation of MSE.

The global (top panels) and local (bottom left panel) performance of CV metamodel is better than SK method performance. The same is valid for the robustness performance illustrated in the bottom right panel. It is interesting to observe that the MSE of CV metamodel is close to zero. As consequence, the other three measures exhibit the same pattern. That is, they are also close to zero. The overall performance exhibited by the CV metamodel indicates that this method is suitable for this canonical problem. The gain in efficiency by utilizing the CV metamodel can be significant even in a problem where correlation among outputs at different solution point may not be very high.

M/M/1 Queue

Another canonical stochastic class of problem where metamodeling is frequently used is in queue's applications. We choose the M/M/1 simulation example from Staum (2009). It is a simple model, but illustrates some of the key features that are commonly found in simulation models used in operations research, but are non-standard in some relevant fields of statistics: (i) the response surface of variable Y (waiting time in line) is smooth and monotone; (ii) the variance over the surface is heterogeneous; (iii) the variability of expected value $\mathrm{E}[Y]$ is much larger over some parts of the domain than over others.

One interesting aspect of this problem is that there are four sources of randomness when computing the objective function via simulation: one must generate a uniform random variable $\mathcal{U}(0,1)$, and three exponential random variables Exponential (1/(service rate - arrival rate)), Exponential (1/(service rate)), and Exponential (1/(arrival rate)). The implication of multiple source of noise to the control variates metamodeling is that the correlation induced by common random numbers is not so strong as in problems with additive noise or with only one source of noise.

Figure 3.6 shows performance measures of the two algorithms (i) stochastic kriging

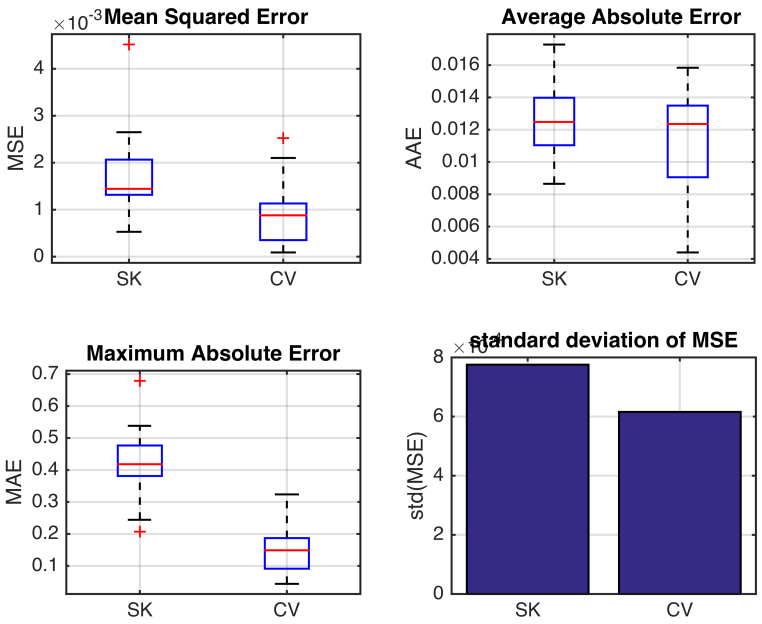

Figure 3.6: Performance measures of stochastic kriging, and control variates metamodeling in the M/M/1 problem

using 100 design points with 1.000 outputs at each design point and, 1.000 prediction points; (ii) and control variates metamodel using 10 design points with 5.000 outputs at each design point, and 1.000 low-fidelity points with 50 outputs at each. A simulation run of 1.000 customers is taken in all algorithms, and each output is their average waiting time. 25 replications of each metamodel is used to construct the boxplots of mean squared error, average absolute error and maximum absolute error, and to compute the standard deviation of mean squared error.

It is interesting to note that, even in problems with multiple sources of noise, all four performance measures of SK are overtaken by the ones of CV metamodeling. Again, we remark that the correlation among outputs in the M/M/1 queue problem is not as high as it was in the first 4 template problems. The reason why correlation is weaker lies on the structure of noise in this problem. There are three sources of noise (defined previously) and the connections between them are not straightforward. Although nature events are equal among solutions because we are using common random numbers, the effects of noise (i.e., nature) on each solution depend on how the random outcomes are interrelated. It is one more example that illustrates the flexibility and efficiency of the CV metamodeling at

problems with different topology of noise.

3.2.3 Analysis of Estimated Coefficient $\hat{\beta}$

The control variates metamodeling performance is directed related to the estimation of coefficient $\hat{\beta}$ in equation (3.11). When the objective function has noise in an additive form, such as in the first four problems previously discussed, the estimated $\hat{\beta}$ has always value 1 because the noise from output distributions of control and of low-fidelity point is the same. Moreover, it is important to note that, in the case of additive noise, the estimated coefficient is also the optimal β^*. Therefore, the control variates metamodeling performance is only affected by the error raised from estimating the control mean $\hat{\mu}(\tilde{x}_k)$ in equation (3.10).

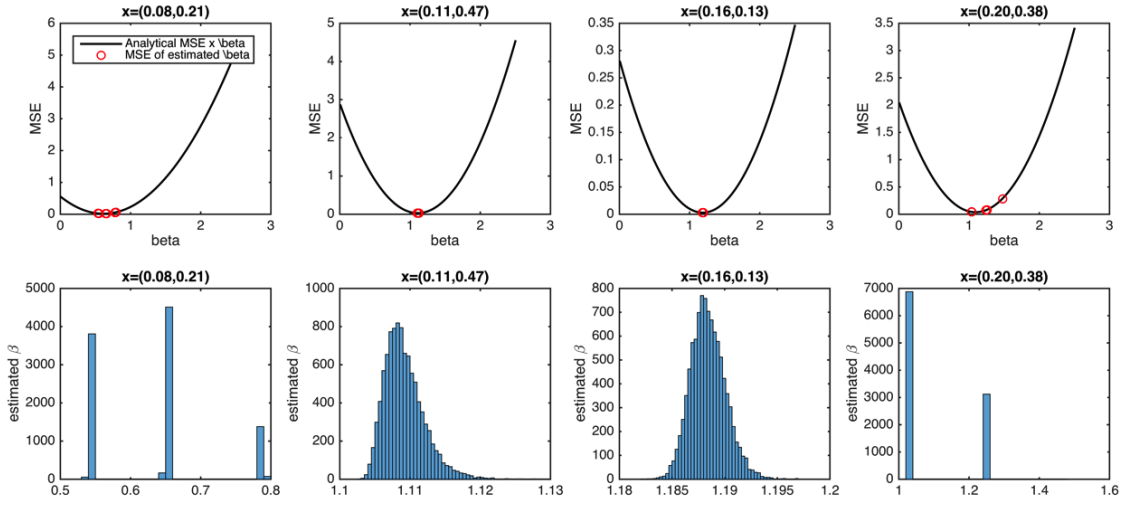

Figure 3.7: MSE for a fixed interval of β, and MSE of estimated $\hat{\beta}$ according to equation (3.11) after 10,000 replications - Asian Call Option example

The top panels in Figure 3.7 show the MSE of a fixed range of β values in the black line, and the MSE of 10,000 estimated $\hat{\beta}$ at different points of the objective function from the Asian Call Option problem in the red bullets. The bottom panels show the corresponding histograms of $\hat{\beta}$. At $x = (0.11, 0.47)$ and $x = (0.16, 0.13)$, the same design point is selected as control at each of the 1,000 replications. Therefore, the underlying histograms show shapes close to a normal distribution with mean centered in the optimal β^*.

On the other hand at $x = (0.08, 0.21)$ and $x = (0.20, 0.36)$, three different design points

are selected as control in different occasions of the 1,000 replications. The left and right histograms show a shape of three normal distributions, where only one of them is centered at the optimal coefficient. It is worth noting that although a fraction of the estimated $\hat{\beta}$ is not centered in the optimal coefficient, their values are sufficiently close to it so that CV Metamodeling can delivery a low MSE.

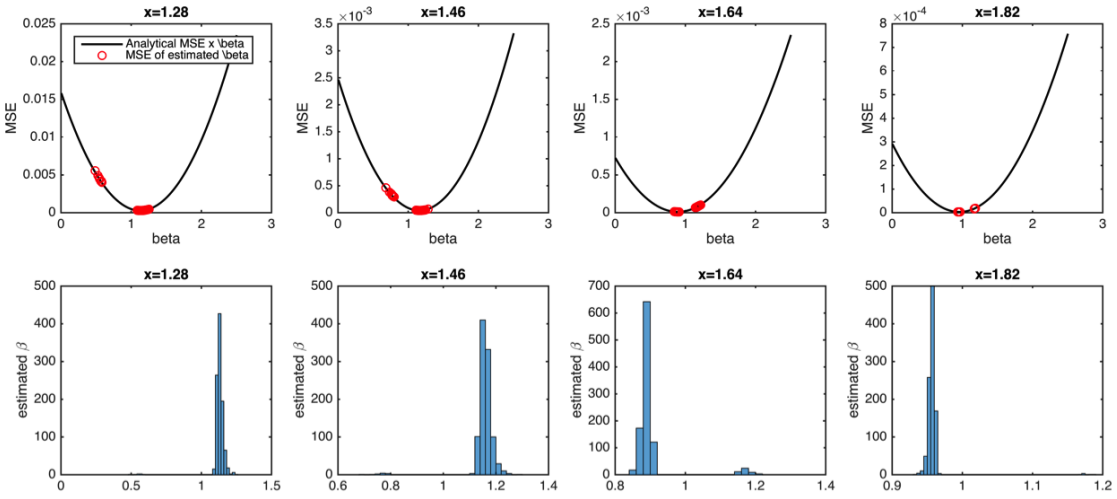

Figure 3.8: MSE for a fixed interval of β, and MSE of estimated $\hat{\beta}$ according to equation (3.11) after 1,000 replications - M/M/1 example

Figure 3.8 shows the same analysis for the M/M/1 example after 1,000 replications. We observe that more than one design point is selected as control at most low-fidelity points among the 1,000 replication. It can be easy explained by the fact that there are two design points with a strong correlation to the underlying points. In the replications, sampled correlation varies according to sampled noise. Therefore, at some replications a particular design point has stronger sampled correlation to the underlying predication point, whereas in other replications the underlying prediction point has stronger sampled correlation to other design point.

In this example, the variance of the objective function is larger at low values of x, and is smaller at high values. As consequence, the fraction of estimated $\hat{\beta}$ which is not centered in the optimal β^* at $x = 1.28$ delivers a higher mean squared error than the ones at $x = 1.82$. However, histograms at the bottom panels show that the frequency of estimated $\hat{\beta}$s for which their distribution are not centered in the optimal β^* is smaller

when compared to the ones which are centered in the optimal coefficient.

3.3 Multiple Controls and Multicollinearity

Here, we extend the multiple CV analysis as in Rosenbaum and Staum (2016). As described in the latter paper, it seems to be attractive to use more than one design point as control variates to the larger set of low-fidelity points in the CV metamodel. However, it may have drawbacks in estimating the coefficient β.

The idea of this section is to analyze CV metamodeling with multiple controls through linear regression. In particular, we want to understand the *multicollinearity* effects that can arise when more than one control is available and are added to the CV estimator model. As reference, we resort to Glasserman (2004) as basis to control variance theory, and to Montgomery and Peck (2001), as basis to linear regression theory.

3.3.1 Connection to Linear Regression

The link between control variates and regression is useful in the statistical analysis of the control variates estimator. There is an alternative form of the simple linear regression model that is occasionally useful and is described in Montgomery and Peck (2001). Suppose that we define the regressor variable $Z(w_i)$ (recall that Z is our control variable in the control variates technique) as the deviation from its own average, say $Z(w_i) - \bar{Z}$ as in (3.8). The regression model then becomes

$$\begin{aligned} Y(w_i) &= \beta_0 + \beta_1 Z(w_i)\epsilon_i \\ &= \beta_0 + \beta_1(Z(w_i) - \bar{Z}) + \beta_1\bar{Z} + \epsilon_i \\ &= (\beta_0 + \beta_1\bar{Z}) + \beta_1(Z(w_i) - \bar{Z}) + \epsilon_i \\ &= \beta_0' + \beta_1(Z(w_i) - \bar{Z}) + \epsilon_i, \end{aligned} \tag{3.13}$$

where β_0 and β_1 are the least-squared estimators, and ϵ_i is a random error component. Note that redefining the regressor variable in (3.13) has shifted the origin of the Z's from zero to $\bar{Z}$. It is easy to show that the least-squares estimator of the transformed intercept

is $\hat{\beta}_0' = \bar{Y}$. The estimator of the slope is unaffected by the transformation. This alternate form of the regression model has some advantages. First, the least-squared estimators

$$\hat{\beta}_0' = \bar{Y}, \qquad \text{and} \qquad \hat{\beta}_1 = -\frac{\text{Cov}[\boldsymbol{Z}, \boldsymbol{Y}]}{\text{Var}[\boldsymbol{Z}]}$$

are uncorrelated. This will help some applications of the model, such as finding confidence intervals on the mean of Y. Therefore, the fitted model is

$$\hat{Y} = \bar{Y} + \hat{\beta}_1(Z - \bar{Z}). \tag{3.14}$$

The Classical CV model (3.1) is very similar to (3.14). The expression in (3.4) is the slope of the least-squares regression throught points $(Y(w_i), Z(w_i))$, $i = 1, ..., n$, which is the least-squares estimator $\hat{\beta}_1$ of the alternate model.

3.3.2 Multiple Controls

Suppose that a simulation produces outputs

$$\boldsymbol{Y} = \begin{pmatrix} Y(w_1) \\ \vdots \\ Y(w_n) \end{pmatrix}, \qquad \boldsymbol{Z} = \begin{pmatrix} Z^{(1)}(w_1) & Z^{(2)}(w_1) & \cdots & Z^{(K)}(w_1) \\ Z^{(1)}(w_2) & Z^{(2)}(w_2) & \cdots & Z^{(K)}(w_2) \\ \vdots & \vdots & & \vdots \\ Z^{(1)}(w_n) & Z^{(2)}(w_n) & \cdots & Z^{(K)}(w_n) \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_d \end{pmatrix},$$

and suppose that the vector of expectations $\text{E}[\boldsymbol{Z}]$ is known or can be somehow estimated. Assume that the pair $(Y(w_i), Z_i)$, $i = 1, ..., n$ are i.i.d with covariance matrix

$$\begin{pmatrix} \Sigma_Z & \Sigma_{ZY} \\ \Sigma_{ZY}^\top & \sigma_Y^2 \end{pmatrix},$$

where Σ_Z is $K \times K$, Σ_{ZY} is $K \times 1$, and the scalar σ_Y^2 is the variance of the $Y(w_i)$. The control variate estimator in this case is given by:

$$\hat{Y}(\beta) = \bar{Y} - \boldsymbol{\beta}^\top(\bar{Z} - \text{E}[\boldsymbol{Z}]),$$

with $\bar{Y}$ and $\bar{Z}$ being the ordinary sample average scalar and vector respectively. The optimal coefficient vector is

$$\boldsymbol{\beta}^* = \Sigma_Z^{-1}\Sigma_{ZY}. \tag{3.15}$$

In practice, the optimal vector of coeffcient $\boldsymbol{\beta}^*$ is unknown but may be estimated. The least-squares normal equation is

$$(\boldsymbol{Z}^\top \boldsymbol{Z})\hat{\boldsymbol{\beta}} = \boldsymbol{Z}^\top \boldsymbol{Y}. \tag{3.16}$$

The number of controls K is ordinarily not very large so size is not an obstacle in inverting $(\boldsymbol{Z}^\top \boldsymbol{Z})$, but if linear combinations of some of the control are highly correlated this matrix may be nearly singular. This should be considered in choosing multiple controls and is the topic of the next Section.

3.3.3 Multicollinearity Effects

If there are near-linear dependencies among the regressors (i.e., they are not orthogonal), the problem of multicollinearity is said to exist. When the method of least squares is applied to non-orthogonal data, very poor estimates of the regression coefficients can be obtained. The variance of the least-squares estimates $\boldsymbol{\beta}$ may be considered inflated, and the length of the vector of least-squares parameter estimates is too long on the average. This implies that the absolute value of the least squares estimates are too large and that they might be unstable. That is, their magnitudes and signs may change considerably given a different sample.

Let us go back to inverting matrix $(\boldsymbol{Z}^\top \boldsymbol{Z})$ in equation (3.16). For simplicity, suppose $K = 2$. We have:

$$\boldsymbol{C} = (\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} = \begin{pmatrix} \frac{1}{1-\rho_{12}^2} & \frac{-\rho_{12}}{1-\rho_{12}^2} \\ \\ \frac{-\rho_{12}}{1-\rho_{12}^2} & \frac{1}{1-\rho_{12}^2} \end{pmatrix}, \tag{3.17}$$

where ρ_{12} is the simple correlation between $Z^{(1)}$ and $Z^{(2)}$. The estimates of the regression coefficients are

$$\hat{\beta}_1 = \frac{\rho_{1Y} - \rho_{12}\rho_{2Y}}{1-\rho_{12}^2}, \qquad \hat{\beta}_2 = \frac{\rho_{2Y} - \rho_{12}\rho_{1Y}}{1-\rho_{12}^2},$$

where ρ_{1Y} and ρ_{2Y} are the respective correlations between Y and $Z^{(1)}$ and $Z^{(2)}$. If there is strong multicollinearity between $Z^{(1)}$ and $Z^{(2)}$, then the correlation coefficient ρ_{12} will be large. From (3.17) we see that as $|\rho_{12}| \to 1$,

$$\text{Var}[\hat{\beta}_j] = C_{jj}\text{Var}[Y] \to \infty, \qquad \text{and} \qquad \text{Cov}[\hat{\beta}_1, \hat{\beta}_2] = C_{12}\text{Var}[Y] \to \pm\infty.$$

Therefore, strong multicollinearity between $Z^{(1)}$ and $Z^{(2)}$ results in large variances and covariance for the least-squares estimators of the regression coefficients. This implies that different samples taken at the same Z levels could lead to widely different estimates of model parameters. Multicollinearity produces similar effects when there are more than two regression variables (i.e., more than two controls used).

Detecting and dealing with multicollinearity

Several techniques have been proposed for detecting multicollinearity. Among them, Montgomery and Peck (2001) introduces examination of correlation matrix, Eigen system analysis and variance inflation factors. Examining the simple correlations ρ_{ij} between the regressors (i.e., controls) is helpful in detecting near-linear dependence between pair of regressors. Unfortunately, when more than two regressors are involved in linear dependences, there is no assurance that any of the pairwise correlations ρ_{ij} will be large.

Here we briefly describe the variance inflation factors method, in which the diagonal elements of $\boldsymbol{C} = (\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}$ matrix in (3.17) are used in detecting multicollinearity. It can be shown that the diagonal elements of the $\boldsymbol{C}$ matrix are

$$C_{jj} = \frac{1}{1 - R_j^2}, \; j = 1, 2, ..., K$$

where R_j^2 is the coefficient of multiple determination from the regression of $Z^{(j)}$ on the remaining $K - 1$ regressor variables. If there is strong multicollinearity between $Z^{(j)}$ and any subset of the other $K - 1$ regressors, then the value of R_j^2 will be close to unit. Therefore, C_{jj} will be large and the variance of its estimated regression coefficient $\text{Var}[\hat{\beta}_j] = C_{jj}\text{Var}[Y]$ will be also very large. Thus, C_{jj} can be viewed as the factor by which the variance of $\hat{\beta}_j$ is increased due to near-linear dependences among regressors. The *variance*

inflator factor (VIF) in then given by:

$$\mathrm{VIF}_j = C_{jj} = \left(1 - R_j^2\right)^{-1}. \tag{3.18}$$

The VIF for each term in the model measures the combined effect of the dependences among regressors on the variance of that term. One or more large VIFs indicates multicollinearity. According to Montgomery and Peck (2001), practical experience indicates that if any of the VIFs exceeds 5 or 10, it is indication that the associated regression coefficients are poorly estimated because of multicollinearity. Furthermore, VIFs can help identifying which regressors are involved in multicollinearity. It is expected that those with large VIFs are associated with near-linear dependences.

3.3.4 CV Metamodel With Multiple Control

When common random numbers are used for sampling the outputs at design points and low-fidelity points, correlation between all sampled points are induced. There is a strong implication of the existence of multicollinearity in the control variates metamodel. If more than one design point are used as controls for a low-fidelity point output, the effects of multicollinearity must be examined.

We examine such an effect in two experiments. We note that the first four experiments discussed in Section 3.2.2 have noise in an additive form in the function of interest J. With common random numbers, the induced correlation between samples is always 1. Thus, only one control can be used because no variance reduction is achieved using more than one. Therefore, in our first experiment, we adapt the stochastic version of the welded beam problem so that noise is inserted not in an additive form.

Then, we analyze the M/M/1 queue problem. As discussed previously, such a problem has three sources of noises, which are interrelated. As consequence, it may be interesting to use the outputs of more than one control to guide the variance reduction of the outputs at a prediction point. That is, there is no control that can fully explain the errors between the mean and observations of prediction point because correlation is not 1.

Experiment 1 - adapted welded beam problem

The adapted the Welded Beam objective function is given by

$$J(X) = 1.10471(x_1 + W_1)^2(x_2 + W_2) + 0.04811(x_3 + W_3)(x_4 + W_4)[14.0 + (x_2 + W_2)],$$

where

$$W_i \sim \mathcal{N}(0, 0.4), \ \forall \ i = 1, ..., 4.$$

Since now there are four independent source of noise, the induced correlation is not 1, although it is still high.

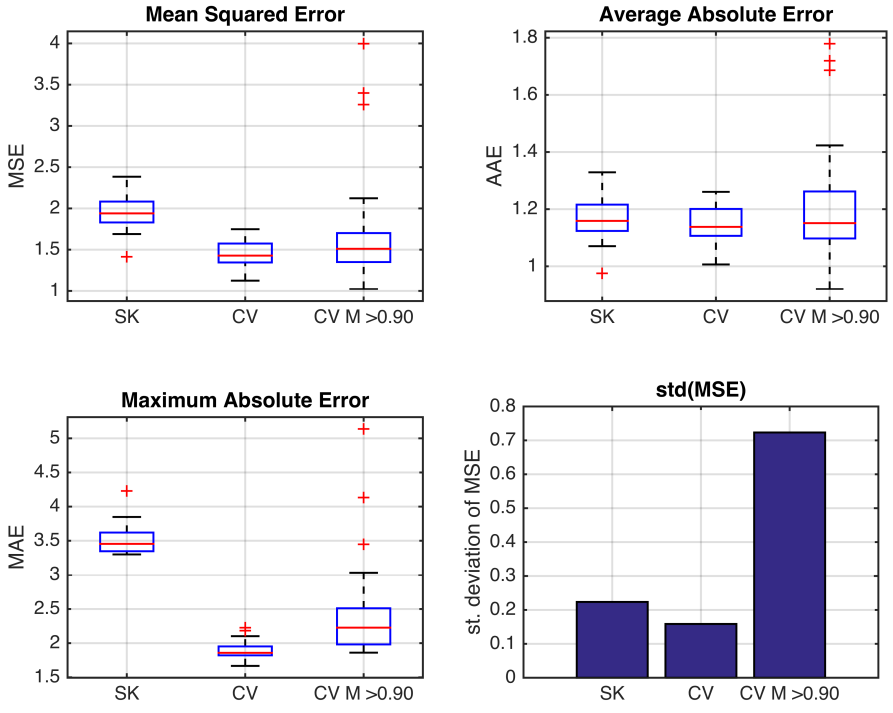

Figure 3.9: Performance measures of stochastic kriging (SK), and control variates metamodel with one (CV) and more (CV M $>$ 0.90) controls in the adapted welded beam problem

Figure 3.9 shows boxplots of mean squared error, average absolute error, maximum absolute error, and standard deviation of mean squared error after 25 replications of three algorithms. The first is stochastic kriging (labeled as SK) using 100 design points with 1.000 outputs at each design point, and 1000 prediction points. The second is control variates metamodel (labeled as CV) using 10 design points with 5000 outputs at each design point, 1000 low-fidelity points with 50 outputs at each, and only one design point

can be used as control. The third is a control variates metamodel (labeled as CV M > 0.90) with the same configuration as the previously algorithm, however design points with correlation higher than 0.90 are qualified to be control, controls are selected according to (3.18), and coefficient $\hat{\beta}$ is computed according to (3.15).

All panels show that, as expected, the CV metamodel with only one control has a better global performance than SK. The top panels show that using more than one control does not improve the CV metamodeling global performance. The bottom left panel shows that the local performance of the CV method can be significantly impacted by multicollinearity effects even if controls are selected according to the method described in Section 3.3.3. Moreover, multicollinearity also has an important negative effect on the robustness performance, which is shown at the bottom right panel.

Experiment 2 - M/M/1 queue system

We complement the illustration of the effects that multicollinearity may cause in the CV metamodeling by examining the M/M/1 queue system of Staum (2009). Since there are four source of noise embedded in this problem, the multicollinearity effects can be evaluated in a scenario where correlation raised by using common random numbers is not as strong as in problems with additive noise or with only one source of noise (e.g., Asian call option).

Figure 3.10 shows boxplots of mean squared error, average absolute error, maximum absolute error, and standard deviation of mean squared error after 25 replications of four algorithms. The first algorithm is stochastic kriging (labeled as SK) using 100 design points with 1000 outputs at each design point and, 1000 prediction points. The second is control variates metamodel (labeled as CV) using 10 design points with 5000 outputs at each design point, 1000 low-fidelity points with 50 outputs at each, and only one design point used as control. The third one is a control variates metamodel (labeled as CV M > 0.80) with the same configuration as previously algorithm, however design points with correlation higher than 0.80 are qualified to be control, controls are selected according to (3.18), and coefficient $\hat{\beta}$ is computed according to (3.15). The fourth algorithm is a control variates metamodel (labeled as CV M > 0.90) with same configuration as previously algorithm, however correlation threshold is 0.90. A simulation run of 1000 customers is taken in all

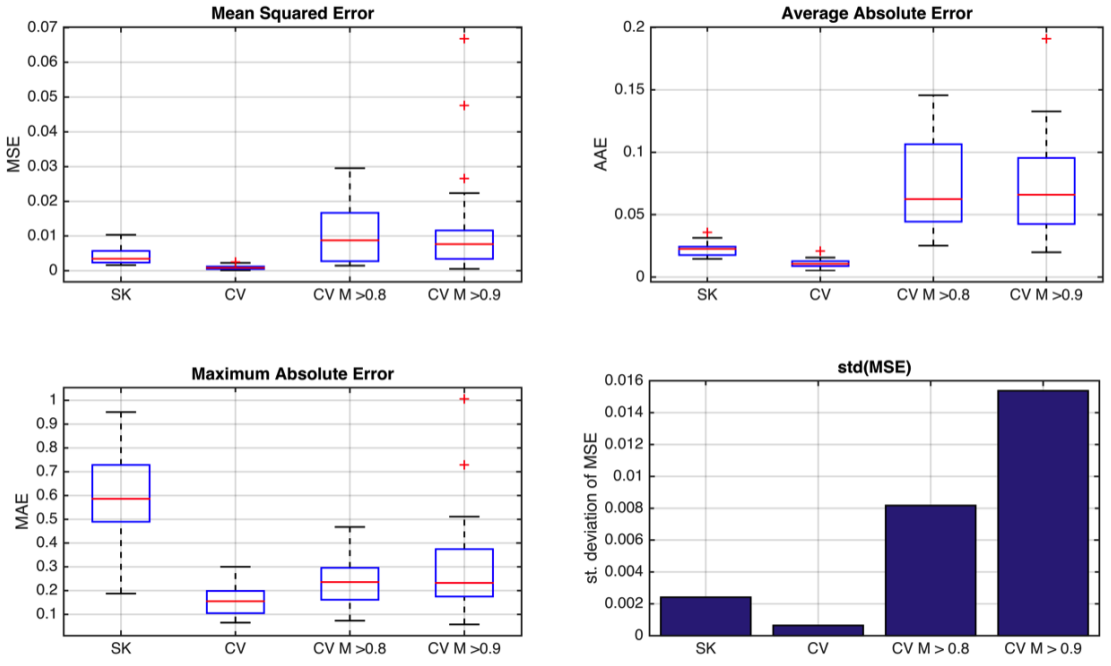

Figure 3.10: Performance measures of stochastic kriging, and control variates metamodel with one and more controls in the M/M/1 problem

algorithms, and each output is the average waiting time of customers.

All panels show that multicollinearity effects raised from induced correlation have a significant impact in estimating coefficient $\hat{\beta}$ when more than one control is used in the CV metamodel. The results are in line with the first experiment (adapted welded beam problem). Top left and right panels show that the global performance of a CV metamodel with multiple controls is worse than the performance of stochastic kriging and CV metamodel with one control. On the other hand, the maximum absolute error panel shows that the local performance of SK is worse than CV metamodel even with multicollinearity effects. The last panel (bottom right) shows that the robustness of CV metamodel is significant affected by multicollinearity.

The experiments corroborates to the argument that the induced correlation via common random numbers hinders the use of multiple controls in the CV metamodel. That is, the performance of CV metamodel may not improve or even worsen in the attempt of increasing accuracy of the estimate by adding more controls.

3.4 Iterative Allocation in the CV Metamodel

One of the opportunities in research on metamodeling, discussed in Chapter 2, is the selection of design points. In this section, we provide an investigation on the choice of design points in the control variates metamodel.

The mean squared error of control variates metamodeling is given by the mean squared error of database control variates in (3.9):

$$\mathrm{MSE}(\hat{Y}_D(\beta)) = \frac{\mathrm{Var}[Y]}{n}\left(1 - \mathrm{Corr}^2[Z,Y]\right) + \frac{\mathrm{Var}[Y]}{N},$$

Equation (3.9) demonstrates that the effectiveness of CV Metamodeling is related to four aspects: (i) the intrinsic variance of underlying interest Y (i.e., $\mathrm{Var}[Y]$); (ii) the correlation between control Z and the quantity of interest Y (i.e., $\mathrm{Corr}[Z,Y]$); (iii) the sampled size n when estimating $\hat{Y}$ at low-fidelity points; and (iv) sampled size N when estimating control mean $\mathrm{E}[Z]$.

Regarding the correlation aspect (ii), the variance reduction factor $(1 - \mathrm{Corr}^2[Z,Y])$ increases very sharply as $|\mathrm{Corr}[Z,Y]|$ approaches 1 and, accordingly, it drops of quickly as $|\mathrm{Corr}[Z,Y]|$ decreases away from 1. Thus, a high degree of correlation is needed for a control variates to yield substantial benefits. As discussed in Section 3.2.2, the use of common random numbers induces the correlation between design points and low-fidelity points. This indicates that the number of design points may be chosen according to its correlation to low-fidelity points. That is, if correlation between a particular design point and a group of low-fidelity points is high enough, there is no need to allocate budget to other design point.

In this Section, we present a procedure to improve simulation budget allocation among design points. The idea is to carefully choose the amount and location of design points so that correlation to all low-fidelity points is always above a high and predefined threshold, and no waste of budget is allocated to 'redundant' design points. By following such a procedure, the expected result is preserving correlation $\mathrm{Corr}^2[Z,Y]$ above a predefined threshold $\boldsymbol{\rho}$ in the same time that the value of N (number of simulation output allocated at each design point) is increased. The main goal is to reduce the MSE in (3.9).

3.4.1 The Procedure

The iterative allocation procedure is based on the two steps described in Section 3.2.1. The main difference is that simulation outputs of low-fidelity points are generated before generating outputs of design points. In this way, an estimated correlation between points can be used to guide the definition of design points. One or more low-fidelity points are iteratively selected and promoted to design point. The steps are as follows:

Defining low-fidelity points

Let L be the number of low-fidelity points, and let B be the total simulation budget. Then, the simulation budget allocated to each low-fidelity point is $n = \lfloor B/(2L) \rfloor$. Note that the simulation budget to be allocated among design point $(B/2)$ is equal to the simulation budget allocated among low-fidelity points. Choose a set of low-fidelity points $X = \{x_1, ..., x_L\}$ in a way that these points are well distributed in the design space. For example, using Latin hypercube technique (see Glasserman (2004) for details).

Simulation at low-fidelity points and definition of design points

Differently from Section 3.2.1, the outputs simulated at low-fidelity points are now stored in a database `Y`. The elements of this database are used to compute the correlation between already selected design points and low-fidelity points. If correlations of a particular low-fidelity point to all already selected design points are under a predefined threshold $\boldsymbol{\rho}$, then this low-fidelity point is selected as design point. The set of design points and its size are defined when the procedure is completed. The scheme version of this step is given below.

Simulation Algorithm at Low-Fidelity Points

---

0 Generate a set $\{w_1, ..., w_n\}$ according to the underlying probability measure.

1 Let `I` be a vector to store the index of low-fidelity points promoted to design-points. Start with `I` $= \{1\}$. Let K be the size of this vector, starting with $K = 1$.

1 For $l = 1, ..., L$

1.1 Perform a simulation effort n at low-fidelity point x_l generating

$$\mathtt{Y}_l = \{Y(x_l, w_1), ..., Y(x_l, w_n)\}$$

and store these outputs in a $n \times L$ database $\mathtt{Y}$ to enable retrieval.

1.2 For $k = 1, ..., K$ compute the correlation between outputs from current low-fidelity point x_l and already selected design points:

$$C(k) = \text{Corr}[\mathtt{Y}_l, \mathtt{Y}_{\mathtt{I(k)}}].$$

1.3 If no element of C is above correlation threshold $\boldsymbol{\rho}$, add index l to set I, and let $K = K + 1$.

Simulation at design points

In this step, the goal is to estimate the control mean of each design point. Let us name the design point set as $\tilde{X} = \{\tilde{x}_1, ..., \tilde{x}_K\}$, where $\tilde{x}_1 = x_1$, $\tilde{x}_2 = x_{\mathtt{I(2)}}$, ... , $\tilde{x}_K = x_{\mathtt{I}(K)}$. The simulation budget allocated to each design point is $N = \lfloor B/(2K) \rfloor$. The control mean is estimated with total budget of $N + n$ because we can use the outputs from database $\mathtt{Y}$ of promoted low-fidelity points. The corresponding scheme version of this step is given below.

Simulation Algorithm at Design Points

---

0 Generate a set $\{w_1, ..., w_N\}$ according to the underlying probability measure.

1 For $k = 1, ..., K$

1.1 Perform a simulation with effort N at design point $\tilde{x}_k$ generating

$$\{Y(\tilde{x}_k, w_1), ..., Y(\tilde{x}_k, w_N)\}.$$

1.2 Compute the estimator

$$\hat{\mu}(\tilde{x}_k) = \frac{1}{N+n}\left[\sum_{i=1}^{n} \mathtt{Y}_{\mathtt{I}(k)}(i) + \sum_{i=1}^{N} Y(\tilde{x}_k, w_i)\right]$$

1.3 Compute the ordinary sample average

$$\bar{\tilde{Y}}_k = \frac{1}{n}\sum_{i=1}^{n} \mathtt{Y}_{\mathtt{I}(k)}(i).$$

Estimation at low-fidelity points

In this step, the objective function value at each low-fidelity point is estimated. A scheme version of this step is given below.

Estimation Algorithm at Low-Fidelity Points

---

1 For $l = 1, ..., L$

1.1 Using the database $\mathtt{Y}$, compute the following sample average

$$\bar{Y}_l = \frac{1}{n}\sum_{i=1}^{n} \mathtt{Y}_l(i).$$

1.2 Set

$$k^* = \arg\max_{k=1,...,K} \mathrm{Corr}[\mathtt{Y}_{\mathtt{I}(k)}, \mathtt{Y}_l].$$

1.3 Compute the CV coefficient:

$$\hat{\beta} = \frac{\mathrm{Cov}[\mathtt{Y}_{\mathtt{I}(k^*)}, \mathtt{Y}_l]}{\mathrm{Var}[\mathtt{Y}_{\mathtt{I}(k^*)}]}.$$

1.4 The estimator of $\mathrm{E}[Y(x_l)]$ is given by

$$\hat{Y}(x_l) = \bar{Y}_l - \hat{\beta}\left(\bar{\tilde{Y}}_{k^*} - \hat{\mu}(\tilde{x}_{k^*})\right).$$

3.4.2 Experimental Results

We illustrate the application of an iterative allocation in CV metamodeling using three of previously discussed problems: the welded beam problem, the Asian call option problem, and the M/M/1 queue problem.

Experiment 1 - welded beam problem

When the stochastic objective function has noise in an additive form, the induced correlation between design points and low-fidelity points in the CV metamodel is 1. Thus, there is no need to have more than one design point as option to be used as control. It indicates that the simulation budget distribution among design points can be improved. That is, if the available budget to all design points is allocated in only one making the value of term N increase, the performance of CV Metamodeling is expected to improve.

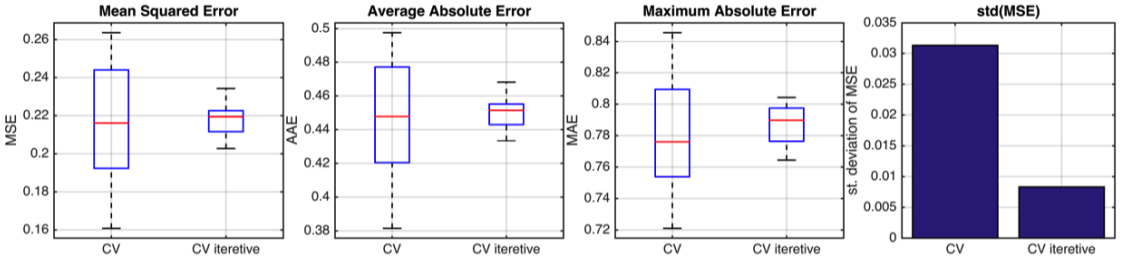

Figure 3.11: Performance measures of standard CV metamodel, and CV metamodel with iterative allocation ($\rho = 0.95$) after 25 replications - welded beam problem with variance scenario 'L+Ho'

Figure 3.11 shows that the main gain applying the iterative CV metamodel in a problem with additive noise is on robustness of the algorithm. It can be explained by noting that the variance of control mean estimate $\hat{\mu}$ has decrease because N increased. That is, we are allocating more simulation budget to the set-up phase of database control variates. Therefore, the resulting variance of estimate at low-fidelity points is also reduced. The effect is a gain in the ability of the CV metamodel in consistently achieving good estimates of the objective function at different replications.

Experiment 2 - Asian call option

The second experiment we use to illustrate the effects of an iterative allocation rule of

simulation budget on the control variates metamodel regards the Asian call option. We conduct 25 replications of the standard procedure of CV metamodel, and the CV metamodel with iterative allocation with correlation threshold of $\rho = 0.95$.

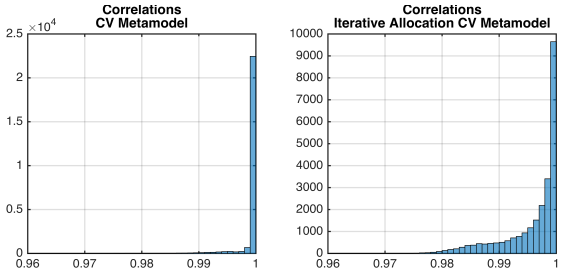

Figure 3.12: Histograms of correlation between design points and low-fidelity points without iterative allocation (right panel), and with iterative allocation (left panel) - Asian Call Option problem

Figure 3.12 shows histograms of correlation between design points and low-fidelity points. We can see that when iterative allocation is applied to CV metamodeling, the frequency of correlations between 0.98 and 1 is increased. As consequence, the frequency of correlations that are close to 1 decreases. It is a direct result of choosing a smaller set of design points. Since the size of control candidates is smaller, the correlations between design and predication points are expected to decrease. However, such a decrease is controlled because there is a lower bound (i.e., a threshold) that guarantees a minimal correlation factor.

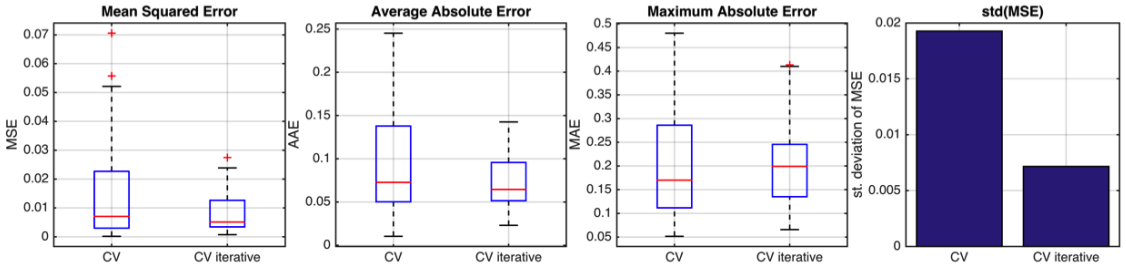

Figure 3.13: Performance measures of CV Metamodeling and CV Metamodeling with iterative allocation ($\rho = 0.95$) - Asian Call Option problem

At the same time, we observe that the results illustrated in Figure 3.13 indicates that the impact of such decrease in induced correlation has no significant effect on local and global performances of the CV metamodel. There is a gain achieved by increasing the sample size N in the set-up phase of database control variates. Such a gain comes with

a cost of selecting a smaller set of design points, and thus decrease correlation factor. However, experimental results show that the gain in robustness compensates the loss in correlation strength. Similarly to the welded beam example, the main gain of iterative allocation in the CV metamodel is on robustness.

Experiment 3 - M/M/1 queue

Although the nature of noise in the M/M/1 queue problem has different characteristics from the noise of the Asian call option problem, their performance results when applying iterative allocation to CV metamodeling are similar (see 3.13 and 3.14).

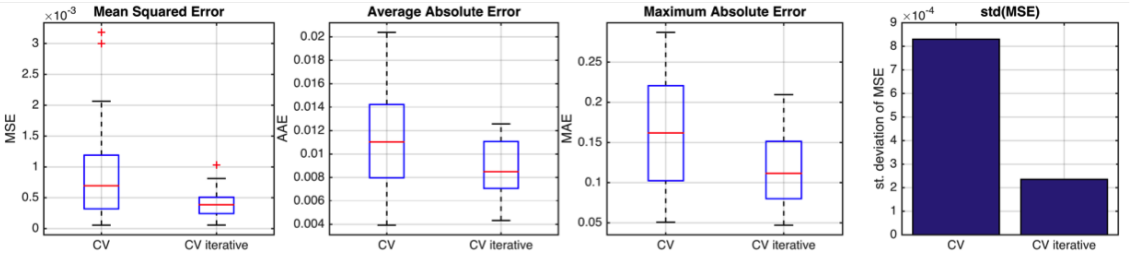

Figure 3.14: Performance measures of CV Metamodeling and CV Metamodeling with iterative allocation ($\rho = 0.95$) - M/M/1 problem

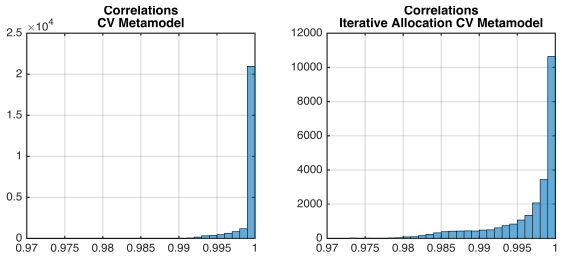

Figure 3.15: Histograms of correlation between design points and low-fidelity points of standard control variates metamodel (right panel), and iterative allocation with $\rho = 0.95$ (left panel) - M/M/1 queue problem

The main difference can be seen at Figure 3.15. The minimum correlation using CV metamodel is close to 0.96, whereas it is close to 0.97 in the iterative allocation. Experimental results show that Iterative allocation CV metamodel can improve not only the choice on set size of design points, but also their locations. In this example, the variance of Y is larger at low values of X, and correlation drops more quickly at this region. Therefore, design points are selected closer to each other in the region with larger variance, and farther in the regions with smaller variances.

The results from the three experiments corroborates to the argument that the performance of CV metamodel can be significantly improved by choosing the set size and location of design points (or controls) based on the correlation among prediction points. In particular, the procedure we propose for choosing the design points aims at increasing the sample size N in the set-up phase of the database control variates by discarding redundant control candidates. Therefore, the main objective is to better estimate control means in (3.10) of the CV metamodel algorithm (i.e., decreasing the underlying error $\mathrm{E}_{\mathtt{W}}[Z]-\mathrm{E}[Z]$ from equation (3.7) of database control variates' algorithm).

We propose a correlation threshold (ρ) that guarantees a minimal strength between correlation of design and prediction points. It ensures a good performance of the CV metamodel while redundant design points are discarded. The main gain of using our procedure for choosing design points is a significant improvement in the metamodel's robustness. That is, the ability on consistently achieving a good estimate of the underlying stochastic function at different replications.

3.5 Final Discussion

In this Chapter, we introduce a novel metamodeling formulation based on the variance reduction technique of control variates. We argue that our control variates metamodel is an efficient method, which requires little input parameter and is flexible enough to be applied to a range of different problems such as queue's and path-dependent options. The indicators of metamodel performance include measures of local and global accuracy, robustness and efficiency. Experimental results show that control variates metamodel outperforms stochastic kriging (which is current the most popular metamodeling framework) in canonical problems raised from real metamodeling applications.

In addition, we extend the multiple control variates analysis of Rosenbaum and Staum (2016). We carefully examined the presence and effects of multicollinearity in the control variates metamodel. Strong multicollinearity, which may arise when common random numbers are used, results in large variances and covariance for least-squares estimators of regression coefficient. We show that the use multiple controls, although it may seems to be attractive, has significant consequences in estimating the CV coefficient $\hat{\beta}$ due to

multicollinearity effects. It has a direct impact on the four performance measures, in particular at robustness.

Moreover, we develop an adaptive scheme that dynamically selects design points to better allocating simulation budget, guided by correlation among low-fidelity points. The selection criterion takes into consideration not only the amount of design points, but also their location so that the correlation between controls and prediction points is always above a predefined threshold. Results show that the main gain of using the iterative allocation rule is in prediction robustness.

As future work, we note that the performance sensibility to the available simulation budget must be evaluated. It is expected a worse performance of CV metamodel in a small simulation budget because the estimation of control means depends on a large enough sample size. Regarding deriving the approximate interval of confidence of the CV metamodel, we believe that such interval is of the nature of the database control variates one, which can be assessed by (3.9).

Regarding the complexity of the CV metamodel, one can find a discussion on the computational cost of database control variates in Borogovac and Vakili (2008). In the latter discussion, the sample size N of setup stage is assumed to be large so that the induced bias raised from (3.7) can be disregarded. In our CV metamodel, one cannot make such assumption because in this case N is just "large enough" so that the bias from setup stage is compensated by the variance reduction rate from estimation stage. Therefore, we believe that the complexity of CV metamodel must be thoroughly investigated.

Another important analysis to be done is the prediction capacity of the control variates metamodel outside the experiment design space. It is known that the performance of response surface methodology may deteriorate significantly at outside points. The need of a function trend is one of the causes of such a behavior, which is not expected in the control variates framework. The latter formulation is not dependent on any auxiliary function, only at the correlation between design points and prediction points. Therefore, the CV metamodel performance at regions outside the experiment design that have high correlations to design points should be good. Preliminary results of ongoing research shows that the prediction capacity of CV metamodel at neighboring points is much better than the one of stochastic kriging and response surface methodology.

Chapter 4

Stochastic Optimization and Control Variates

Variance reduction techniques are frequently used in problems where Monte Carlo simulation is applied. It is worth noting that there are only a few (and very recent) stochastic optimization studies making use/analysis of such methods, although many of them have Monte Carlo simulations embedded in their algorithms. The need of more thorough analysis on the use of variance reduction technique in stochastic optimization is evidenced and discussed in Chapter 2.

Stochastic optimization research applying variance reduction techniques include, but are not limited to: Nelson and Staum (2006) and Tsai and Nelson (2010), where control variates technique is employed to random and selection method; Anderson et al. (2006), applying common random numbers and antithetic variates to scatter search method; and Sen and Bhattacharya (2015), in which subset simulations and importance sampling is employed to genetic algorithms. In Homem-De-Mello and Bayraksan (2015), one can find a review on the use of variance reduction techniques in the stochastic optimization setting.

According to Borogovac and Vakili (2008), a critical barrier to finding effective controls (and consequently applying control variates techniques) is that control means needs to be available to the user (i.e., known, which is the case of Nelson and Staum (2006) and Tsai and Nelson (2010)). In database Monte Carlo (Borogovac and Vakili (2008)), this requirement is no longer needed. The main idea is that "computational effort invested

in one estimation problem may lead to more precise or computationally efficient estimators for related problems". In our view, many stochastic optimization techniques involve estimating the same response surface under different configurations of decision variables. One may use the already existing knowledge of the response on set of decision variables to improve (the accuracy or the efficiency) the estimate of response at other sets by using database Monte Carlo.

The main contribution of this Chapter is to propose a hybrid formulation, resorting to database control variates (a variation of database Monte Carlo proposed in Zhao et al. (2007), Borogovac and Vakili (2008) and Borogovac (2009)) to improve the performance of a random search method named adaptive hyperbox algorithm (AHA) proposed in (Xu et al. (2013)). As remarked in Chapter 2, random search methods have been used as tool to provide an approximate solution (i.e., best solution found as opposite to optimal solution) at high complexity problems such as NP-hard problems.

It is worth mentioning that our hybrid proposal is general, in the sense that it can be employed to a larger set of stochastic optimization methods. That is, our formulation can be easily adapted to many stochastic optimization methods other than random search methods. In the AHA method, likewise in many discrete stochastic optimization methods, some solution points (decision variables) can be revisited many times. That is, there are some points in the design space where more simulation outputs have been sampled than others. As the algorithm runs, more simulations can be taken in a particular solution point. The novelty of our hybrid method is the use of outputs from points with many samples to improve the efficiency (by reducing the variance of Monte Carlo simulations via database control variates) in estimating the function value at other points which have enough correlation associated.

To conduct the analysis, we use the five templates of stochastic optimization problems utilized in Xu et al. (2013). Moreover, we propose three different instances of each template changing the noise configurations. These templates are interesting because they reflex the challenges of many real applications, and embrace different variance structures (homogeneous, heterogeneous, large, and small variances).

The results demonstrate that our hybrid formulation can be beneficial in finding the optimal solution of a stochastic objective function, in particular at the case of large vari-

ance. Moreover, the main gain of combining control variates technique to a random search method is to improve the algorithm's ability at consistently finding good solution at different replications (i.e., at different outcomes of noise). Experiments on high-dimensional problems indicate that our hybrid formulation can scale-up well with the increase of decision variables.

The rest of the Chapter is organized as follows. The AHA and database control variates methods are introduced in Section 4.1. In Section 4.2, we introduce the AHA-DCV formulation and discuss its main features. On going experimental results are given in Section 4.3. Section 4.4 is dedicate to a future research and final discussion.

4.1 Preliminaries

We address the problem of optimizing an unknown response surface of a system modeled by a stochastic simulation. System performance is a random variable $Y(\boldsymbol{x})$ that changes according to D-dimensional decision variables $\boldsymbol{x}$. We assume the design space is $\boldsymbol{\theta} = \Phi \cap \mathbb{I}^D$, where Φ is convex and compact, and $\mathbb{I}^D$ denotes the D-dimensional integer lattice. If the problem is a minimization one, formally we have:

$$\text{minimize } J(\boldsymbol{x}) = \mathrm{E}\left[Y(\boldsymbol{x})\right]$$

with $\boldsymbol{x} \in \boldsymbol{\theta}$.

Next, we introduce the adaptive hyperbox algorithm proposed in Xu et al. (2010b), which is an adaptive random search method under the metaheuristic umbrella. Then, we introduce the database control variates by Zhao et al. (2007), Borogovac and Vakili (2008), and Borogovac (2009), which is a flexible variance reduction technique. These two algorithms constitute the foundations of our hybrid method, which is introduced in 4.2.

4.1.1 Adaptive Hyperbox Algorithm

First, let us start by describing how, in general, an adaptive random search algorithm for discrete stochastic optimization operates. The key element is to construct a most promising area (MPA) at each iteration. That is, defining a set of feasible solutions that are more

likely to be the optimal one. The main difference between algorithms is how this MPA is constructed. Aside the MPA, two main procedures take places: a sampling procedure, and a estimation procedure. The *sampling procedure* consists in, at each iteration, randomly selecting a set of feasible solutions from the MPA. To accomplish such a task, a probability distribution defined on the MPA is used. For example, a uniform distribution, in which all solutions in the MPA are equally likely to be chosen. The size of MPA is expected to decrease as the algorithm evolves. Therefore, it is possible that there are duplicates in the set of selected solutions, and they must be removed. In the *estimation procedure*, the performance of the system is evaluated multiple times. That is, a number of outputs are sampled at each solution in the selected set. These outputs are stored. The performance is estimated by an ordinary average using the total number of outputs that a selected solution has received up to the current iteration.

We observe that our proposal formulation links the estimation procedure of an adaptive random search algorithm to database control variates. The main goal is to reduce the variance (i.e., to improve efficiency) in estimating the system performance at solutions with fewer samples by using the outputs of solutions that have received a sufficient amount of samples. We continue on building the intuitions behind our hybrid formulation in a further discussion at Section 4.1.2, when the database control variates is introduced.

Now, let us introduce AHA, which is an adaptive random search algorithm for solving high-dimensional discrete optimization problem via simulation models proposed in Xu et al. (2013). It constructs a most promising area (MPA) that takes the form of a hyperbox. Let $\boldsymbol{x}$ be a visited solution, with $x^{(d)}$ be its dth coordinate for $d = 1, ..., D$. The current best solution is labeled as $\hat{\boldsymbol{x}}_k^*$. Let $\mathscr{I}(k)$ be the set of unique sampled solutions up to iteration k. Let $\mathscr{L}_k = (l_k^{(1)}, ..., l_k^{(D)})$ and $\mathscr{U}_k = (u_k^{(1)}, ..., u_k^{(D)})$ be respectively the lower and upper dimension bounds of the MPA at iteration $k = 1, ...K$, where

$$l_k^{(d)} = \max_{\boldsymbol{x} \in \mathscr{I}(k),\ \boldsymbol{x} \neq \hat{\boldsymbol{x}}_k^*} \left\{ x^{(d)} : x^{(d)} < \hat{x}_k^{*(d)} \right\} \text{ if it exists; otherwise } l_k^{(d)} = -\infty, \tag{4.1}$$

and similarly

$$u_k^{(d)} = \min_{\boldsymbol{x} \in \mathscr{I}(k),\ \boldsymbol{x} \neq \hat{\boldsymbol{x}}_k^*} \left\{ x^{(d)} : x^{(d)} > \hat{x}_k^{*(d)} \right\} \text{ if it exists; otherwise } u_k^{(d)} = \infty. \tag{4.2}$$

The hyperbox is given by

$$\mathscr{H}_k = \left\{ \boldsymbol{x} : l_k^{(d)} \leq x_{(d)} \leq u_k^{(d)}, d = 1, ..., D \right\}, \tag{4.3}$$

and the MPA at iteration k is defined by $\mathscr{C}_k = \mathscr{H}_k \cap \boldsymbol{\theta}$. The AHA algorithm is described below

Adaptive Hyperbox Algorithm

---

0 Let $\boldsymbol{x}_0$ be a starting solution provided by the user. Set the iteration counter $k = 0$. Let $\mathscr{I}$ be the set of unique sampled solutions on iteration k starting with $\mathscr{I}(0) = \{\boldsymbol{x}_0\}$ and $\hat{\boldsymbol{x}_k}^* = \boldsymbol{x}_0$. Let $\mathscr{E}_k$ be the set of solutions to be evaluated at iteration k, starting with $\mathscr{E}_0 = \{\boldsymbol{x}_0\}$. For simplicity, take s observations from $\boldsymbol{x}_0$. Let $n_k(\boldsymbol{x})$ denote the total number of simulation outputs $\boldsymbol{x}$ has received up to iteration k. Set $n_0(\boldsymbol{x}_0) = s$, and calculate the ordinary average $\bar{Y}_0(\boldsymbol{x}_0)$.

1 Let $k = k+1$. Determine $\mathscr{L}_k$ (lower dimension bounds in 4.1), $\mathscr{U}_k$ (upper dimension bounds in 4.2), $\mathscr{H}_k$ (hyperbox in 4.1) and the MPA $\mathscr{C}_k = \mathscr{H}_k \cap \boldsymbol{\theta}$ (for $k = 1$, $\mathscr{C}_k = \boldsymbol{\theta}$). Let m denote the number of solutions to be randomly chosen from the MPA. Choose $\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{km}$ independently from $\mathscr{C}_k$ using, for simplicity, a uniform distribution. Remove any duplicates from $\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{km}$, and let $\mathscr{I}_k$ be the remaining set. Update the set of unique sampled solutions $\mathscr{I}(k) = \mathscr{I}(k-1) \cup \mathscr{I}_k$.

2 Let $\mathscr{E}_k = \mathscr{I}_k \cup \{\hat{\boldsymbol{x}}_{k-1}^*\}$. For all $\boldsymbol{x} \in \mathscr{E}_k$, take s simulation observations, update total number of samples $n_k(\boldsymbol{x}) = n_{k-1}(\boldsymbol{x}) + s$, and update the ordinary average $\bar{Y}_k(\boldsymbol{x})$ using all samples.

3 Let $\hat{\boldsymbol{x}}_k^* = \arg\ \min_{\boldsymbol{x} \in \mathscr{E}_k} \bar{Y}_k(\boldsymbol{x})$.

4.1.2 Database Control Variates

In Chapter 3, we introduced the classical control variates, which is one of the most effective and broadly applicable variance reduction technique for improving the efficiency of

Monte Carlo simulation (see Glasserman (2004) for details). In section 3.1, we presented a key assumption of control variates technique, which is the requirement of known control means. Usually, the set of control variables Z that are very informative about the variable of interest (Y) typically contains the same complexity as Y. That is, typically $\mathrm{E}[Z]$ is also unknown, which hinders the applicability of control variates technique. One flexible approach that relax this assumption is database control variates, proposed in Zhao et al. (2007), Borogovac and Vakili (2008), and Borogovac (2009).

There are two stages in the database control variates: the set-up stage where control means ($\mathrm{E}[Z]$) are defined, and the estimation stage where the mean of variable of interest ($\mathrm{E}[Y]$) is estimated.

In the *set-up stage*, N outputs of control variable Z are generated via Monte Carlo simulation. Suppose we can model Z as a stochastic function of w, the latter representing the underlying noise. That is, we generate a set $w_1, ..., w_N$, of i.i.d. random variables according to the underlying probability measure P defined on Z. Then, we generate the set of outputs $Z(w_1), ..., Z(w_N)$. The noise samples must be stored into a *database* `W` to enable retrieval in the estimation stage. Outputs of Z may also be stored in a database `Z`, which is advantageous in simulations with a high cost per sample (the case of many simulation models). The control means computed in the set-up stage is given by the following ordinary average:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} Z(w_i). \tag{4.4}$$

The estimation error $\hat{\mu} - \mathrm{E}[Z]$ is fixed, unknown and $\mathrm{E}[\hat{\mu}] = \mathrm{E}[Z]$ (for details, see Borogovac (2009)). As remarked in Borogovac and Vakili (2008), $\hat{\mu}$ can be viewed as the expected value of random variable Z restricted to the probability space `W` (the database containing noise samples) with respect to a uniform measure $P_{\mathtt{W}}$ on this discrete probability space. That is,

$$\hat{\mu} = \mathrm{E}_{\mathtt{W}}[Z],$$

where $\mathrm{E}_{\mathtt{W}}$ denotes expectation with respect to $P_{\mathtt{W}}$.

For this reason, in the *estimation stage*, the underlying probability measure P is re-

placed by the uniform measure $P_{\mathtt{W}}$ with probability space $\mathtt{W}$, where

$$P_{\mathtt{W}}(w) = \frac{1}{N}, \qquad \forall\ w \in \mathtt{W}.$$

Therefore, let us denote a vector of noise $\boldsymbol{w} = (w_1, ..., w_n)^\top$ as the n uniformly selected elements from $\mathtt{W}$. Let us now retrieve from $\mathtt{Z}$ the corresponding vector $\boldsymbol{Z} = (Z(w_1), ..., Z(w_n))^\top$, and compute the ordinary sample average $\bar{Z}$. Accordingly, evaluate the vector $\boldsymbol{Y} = (Y(w_1), ..., Y(w_n))^\top$ and compute the ordinary sample average $\bar{Y}$. The estimated coefficient of database control variates is the classical control variates one:

$$\hat{\beta} = \frac{\mathrm{Cov}[\boldsymbol{Z}, \boldsymbol{Y}]}{\mathrm{Var}[\boldsymbol{Z}]}. \tag{4.5}$$

The database control variates estimator is given by:

$$\hat{Y}(\hat{\beta}) = \bar{Y} - \hat{\beta}(\bar{Z} - \hat{\mu}).$$

The mean squared error of the database control variates estimator using the classical CV coefficient in (4.5) is given by:

$$\mathrm{MSE}(\hat{Y}(\beta)) = \frac{\mathrm{Var}[Y]}{n}\left(1 - \mathrm{Corr}^2[Z, Y]\right) + \frac{\mathrm{Var}[Y]}{N}, \tag{4.6}$$

which is the variance of the classical control variates estimator plus a term $(\mathrm{Var}[Y]/N)$ that arise from the error $\mathrm{E}_{\mathtt{W}}[Z] - \mathrm{E}[Z]$.

In the previous section, an adaptive random search was introduced. In this algorithm, the total number of samples is different between solutions. That is, as the algorithm evolves, some solutions receive more simulation outputs than others. In our proposal formulation, which is described in the next section, we label solutions that have a total simulation output higher than a sample size threshold as control candidates. That is, we take solutions with a large sampled size as candidates to play the role of control variables Z. Then, if the correlation between a solution to be sampled at current iteration and a control candidate is above a correlation threshold, we apply the database control variates technique to reduce variance of estimated system performance at the solution to

be sampled. That is, we see solutions to be sampled as potential variable of interest Y in the database control variates context.

4.2 The AHA-DCV Fomulation

In this section, we propose our hybrid formulation (named AHA-DCV) for adaptive random search algorithms using database control variates for improving the estimation of system performance. We present a scheme of the AHA-DCV algorithm bellow, and after we discuss its main features.

Adaptive Hyperbox Algorithm with Database Control Variates (AHA-DCV)

---

0 Set a correlation threshold ϱ. Let $\boldsymbol{x}_0$ be a starting solution provided by the user. Set iteration counter $k = 0$. Make the set of unique sampled solutions as $\mathscr{I}(0) = \{\boldsymbol{x}_0\}$ and $\hat{\boldsymbol{x}}_k^* = \boldsymbol{x}_0$. Let the set of solutions to be evaluated at current iteration start as $\mathscr{E}_0 = \{\boldsymbol{x}_0\}$. Let p be the number of noise sources. Generate η p-dimensional vectors of noise $\{\boldsymbol{w}_1, ..., \boldsymbol{w}_\eta\}$ according to the underlying probability measure P defined on Y. Store the noise vectors into a database `W` to enable retieval. Take η observations $\{Y(\boldsymbol{x}_0, \boldsymbol{w}_1), ..., Y(\boldsymbol{x}_0, \boldsymbol{w}_\eta)\}$, also store these simulation outputs at `Y`. Set $n_0(\boldsymbol{x}_0) = \eta$, and calculate the ordinary average $\bar{Y}_0(\boldsymbol{x}_0)$.

1 Let $k = k + 1$. Determine $\mathscr{L}_k$ (lower dimension bounds of MPA in 4.1), $\mathscr{U}_k$ (upper dimension bounds of MPA in 4.2), $\mathscr{H}_k$ (hyperbox in 4.3) and MPA $\mathscr{C}_k = \mathscr{H}_k \cap \boldsymbol{\theta}$ (for $k = 1$, $\mathscr{C}_k = \boldsymbol{\theta}$). Choose $\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{km}$ independently from $\mathscr{C}_k$ using, for simplicity, an uniform distribution. Remove any duplicates from $\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{km}$, and let $\mathscr{I}_k$ be the remaining set. Update the set of unique sampled solutions $\mathscr{I}(k) = \mathscr{I}(k-1) \cup \mathscr{I}$.

2 Let the ith control candidate $\boldsymbol{x}_{CV}^{(i)}$ be the ith visited solution listed in $\mathscr{I}(k-1)$ with $n_{k-1} \geq \eta$, $i = 1, ..., I$, I being the number of visited solution that have been sampled more than η times. Let $\mathscr{E}_k = \mathscr{I}_k \cup \{\hat{\boldsymbol{x}}_{k-1}^*\}$. For all $\boldsymbol{x} \in \mathscr{E}_k$:

2.1 Take s simulation observations at $\boldsymbol{x}$ using as noise the corresponding vector of random variables $\boldsymbol{w}$ at `W`. That is, recover from `W` the elements at positions $a =$

$n_{k-1}(\boldsymbol{x})+1$ to $b = n_{k-1}(\boldsymbol{x})+s$ so that common random number are used among all sampled solutions. Compute and store the vector of observations $\mathtt{Y}_{ab}(\boldsymbol{x}) = (Y(\boldsymbol{x}, \boldsymbol{w}_a), ..., Y(\boldsymbol{x}, \boldsymbol{w}_b))^\top$. Update total number of observations $n_k(\boldsymbol{x}) = b$, and update the ordinary average $\bar{Y}_k(\boldsymbol{x})$ using all b observations.

2.2 If $\boldsymbol{x} = \boldsymbol{x}_{CV}^{(i)}$ for any i, go to the next element of $\mathscr{E}_k$. Else, compute

$$\rho^{(i)} = \text{Corr}[\mathtt{Y}_{1b}(\boldsymbol{x}), \mathtt{Y}_{1b}(\boldsymbol{x}_{CV}^{(i)})] \qquad \forall\ i = 1, ..., I.$$

Let $i^* = \arg\max_{i=1,...,I} \rho^{(i)}$. If $\rho^{(i^*)} < \varrho$, go to the next element of $\mathscr{E}_k$. Else $\boldsymbol{x} \neq \boldsymbol{x}_{CV}^{(i)}$ for all $i = 1..., I$, and $\rho^{(i^*)} \geq \varrho$. In this case, update the ordinary average $\bar{Y}_k(\boldsymbol{x}) = \hat{Y}(\boldsymbol{x}, \hat{\beta})$ using the following database control variates estimator:

$$\hat{Y}(\boldsymbol{x}, \hat{\beta}) = \bar{\mathtt{Y}}_{1b}(\boldsymbol{x}) - \hat{\beta}\left[\bar{\mathtt{Y}}_{1b}(\boldsymbol{x}_{CV}^{(i^*)}) - \bar{Y}_{k-1}(\boldsymbol{x}_{CV}^{(i^*)})\right]$$

where

$$\hat{\beta} = \frac{\text{Cov}[\mathtt{Y}_{1b}(\boldsymbol{x}), \mathtt{Y}_{1b}(\boldsymbol{x}_{CV}^{(i^*)})]}{\text{Var}[\mathtt{Y}_{1b}(\boldsymbol{x}_{CV}^{(i^*)})]}.$$

3 Let $\hat{\boldsymbol{x}}_k^* = \arg\ \min_{\boldsymbol{x}\in\mathscr{E}_k} \bar{Y}_k(\boldsymbol{x})$.

The algorithm starts with a user choice on two parameters. First ϱ is chosen, a correlation threshold between control and variable of interest. Because control variates efficiency depend on a high correlation factor (see (4.6)) and is very sensitive to it, usually $\varrho > 0.9$ is an acceptable measure. Taking a very high threshold may discard "good" controls. On the other hand, taking a low correlation threshold may lead to poor control variates estimations (see Glasserman (2004) for details).

The second parameter is η, the minimal sample size of control candidates. Parameter η is direct related to the sample size N at the set-up stage of database control variates. The larger η is, the more accurate is the control mean (see (4.4)). Let us label the total number of outputs taken at any control candidate $i = 1, ..., I$ as $N^{(i)} = n_k(\boldsymbol{x}_{CV}^{(i)})$. Since the sample size of all control candidates must be equal or larger than η (i.e., $N^{(i)} \geq \eta$ for all $i = 1, .., I$), this parameter has relevant impact on algorithm's performance (see (4.6)).

Similarly, a small η may also lead to poor control variate estimations because it can increase the error $\mathrm{E}_{\mathtt{W}}[Y(\boldsymbol{x}_{CV}^{(i)})] - \mathrm{E}[Y(\boldsymbol{x}_{CV}^{(i)})]$, for all $i = 1, ..., I$ (see 4.6). It is important to note that the sample size of control candidates $N^{(i)}$ can increase with iterations. Depending on response surface (system performance function) characteristics, a solution may be visited many times, which can improve the AHA-DCV performance by better estimating control means.

The AHA-DCV formulation requires storing noise vectors in a database for two reasons. First, to enable the problem transformation from probability measure P to $P_{\mathtt{W}}$ as required by database control variates. Secondly, to enable the use of common random numbers among simulations, inducing higher correlation among the outputs of controls ($Y(\boldsymbol{x}_{CV}^{(i)}, \boldsymbol{w})$, $\boldsymbol{w} \in \mathtt{W}$) and variables of interest ($Y(\boldsymbol{x}, \boldsymbol{w})$, $\boldsymbol{x} \in \mathscr{E}_k$ and $\boldsymbol{w} \in \mathtt{W}$). However, storing $\boldsymbol{w}$ requires memory space, which grows linearly with total simulation budget. It is important to note that although storing outputs $Y(\boldsymbol{x}, \boldsymbol{w})$ on the database $\mathtt{Y}$ can be advantageous in a simulation with a high cost per sample, it is not an algorithm's requirement.

We observe that it is a choice to start the algorithm by taking η observations at initial point $\boldsymbol{x}_0$. Depending on the response surface, the algorithm can start by taking only $s << \eta$ samples at initial point and wait until a solution is visited more than η times to have the first control candidate. However, there is no guarantee that any solution will receive that many observations in this case. That is, there is no guarantee that a control candidate will be found until is termination if $n_0(\boldsymbol{x}_0) = s$. As consequence, the performance of our hybrid algorithm AHA-DCV would be exactly the same as stand alone AHA one because there is no available control.

Up to this moment, no special stooping rule has been discussed. We assume the algorithm stops when total simulation budget is consumed. We can add a random jump to avoid local minimal at any solution $\boldsymbol{x} \in \boldsymbol{\theta}$ if no performance improvement was experienced in the last iterations. There are many other more sophisticated strategies to avoid getting trapped at local minima. Because it is not the focus of this research to evaluate different strategies that avoid local optimum, we use this simple one without loss of generality.

4.3 Numerical Experiments

In order to provide a proper analysis of an adaptive random search method combined to a variance reduction technique, we use as template the test problems utilized in Xu and Nelson (2013) and Xu et al. (2010b) (all discrete problems from stochastic optimization). We give a description of these problems and real application examples that can be illustrated by these templates. For each problem, we tested three instances to observe the method behavior at different variance magnitudes and topology. The default for sample size threshold is $\eta = 100$, and for correlation threshold is $\varrho = 0.9$. The default for sample size is $s = 10$, and for size of selected solutions from most promising area is $m = 10$.

Since computational effort is critical to evaluate the method's efficiency, we run both stand alone AHA and AHA-DCV under a range of simulation budged. A total of 100 replications are taken for each tested problem under each simulation budget. Four performance measures are used to conduct the analysis. They are adapted from Li et al. (2010). The first one is the *average among replications of best solution found*, which reflects the global accuracy in the search of optimum. The second one is the *standard deviation among replications of best solution found*, which reflects the model robustness or ability to consistently achieve similar accuracies at different replications. The third one is *best solution found among replication*, and the fourth one is the *worst solution found among replication.* They reflect the presence of good/poor solutions and its range.

4.3.1 Inventory Problem

We start with the *inventory* problem, which delivered the best AHA-DCV performance compared to stand alone AHA among all canonical examples. It is an inventory management problem with dynamic consumer distribution adapted from Mahajan and van Ryzin (2001). In this problem, the analytic optimal solution is not available, which is the most common case in real stochastic optimization applications. It is a one-shot inventory stocking decision taken by a retailer for D product variants at the beginning of a season. Let x_d for $d = 1, ..., D$ be the initial inventory level. It is assumed that replenishment is not allowed, and there is no salvage product value. Pricing is assumed to be an exogenous decision. The unit price of each variant is labeled as p_d, and its cost is c_d. The objective is

to *maximize* the expected proft (total revenue minus total cost). The number of customers follows a Poisson distribution. Each customer $t = 1, ..., T$ chooses from the variant that are in-stock when he/she arrives and there is a no-purchase option. Customer's choice is modeled by a multinomial logit model $U_{td} = a_d - p_d + \xi_{td}$, where a_d is a variant quality index, and ξ_{td} follows a Gumbel distribution. Thus, there are two sources of noise in this system: the total number of customers; and a noise that is added to the customer's utility in an additive form.

In our tests we have: $D = 6$, $p_d = 6$, $c_d = 3$, $a_0 = 4$ (variat quality index of no-purchase option), and $a_d = 12.25 - 0.5(d - 1)$ for $d = 1, ..., D$. There are three instances:

(i) $0 \leq x_d \leq 100$, $T \sim$ Poisson(100), and $\xi_{td} \sim$ Gumbel$(1, 1)$.

(ii) $0 \leq x_d \leq 100$, $T \sim$ Poisson(100), and $\xi_{td} \sim$ Gumbel$(0.5, 0.5)$.

(iii) $0 \leq x_d \leq 1000$, $T \sim$ Poisson(1000), and $\xi_{td} \sim$ Gumbel$(0.5, 0.5)$.

Because no optimal analytic solution is known, an estimate of "true" value is taken based on 10,000 replications for each algorithm run (i.e., for each best solution found).

Figure 4.1 shows performance plots for the three instances of the inventory management problem. The first row of panels shows that AHA-DCV converges to a better solution faster than stand-alone AHA for all three problem's instances. It is interesting to note that even in the range of small budget ($B \leq 10,000$), control variates method delivered a better average best solution. That is, the gain in efficiency by using database control variates is significant even in a scenario where the total number of outputs sampled at control candidate points may not be very large and therefore there are relatively small number of outputs to compute control means.

The AHA-DCV accuracy performance is, on average at all range of simulation budget, 4.9%, 6.5%, and 6.3% better than stand-alone AHA respectively for each instance. The AHA-DCV accuracy performance is, at the best, 9%, 12%, and 8% better than stand-alone AHA respectively for each instance. It occurred when the simulation budget is 4000, 4000 and 7000 respectively for each instance. That is, within relatively small simulation budgets. As budget increases, the stand alone AHA performance approximates AHA-DCV. Because there are more simulations available, eventually stand alone AHA ends up finding a solution that is as good as the one in AHA-DCV.

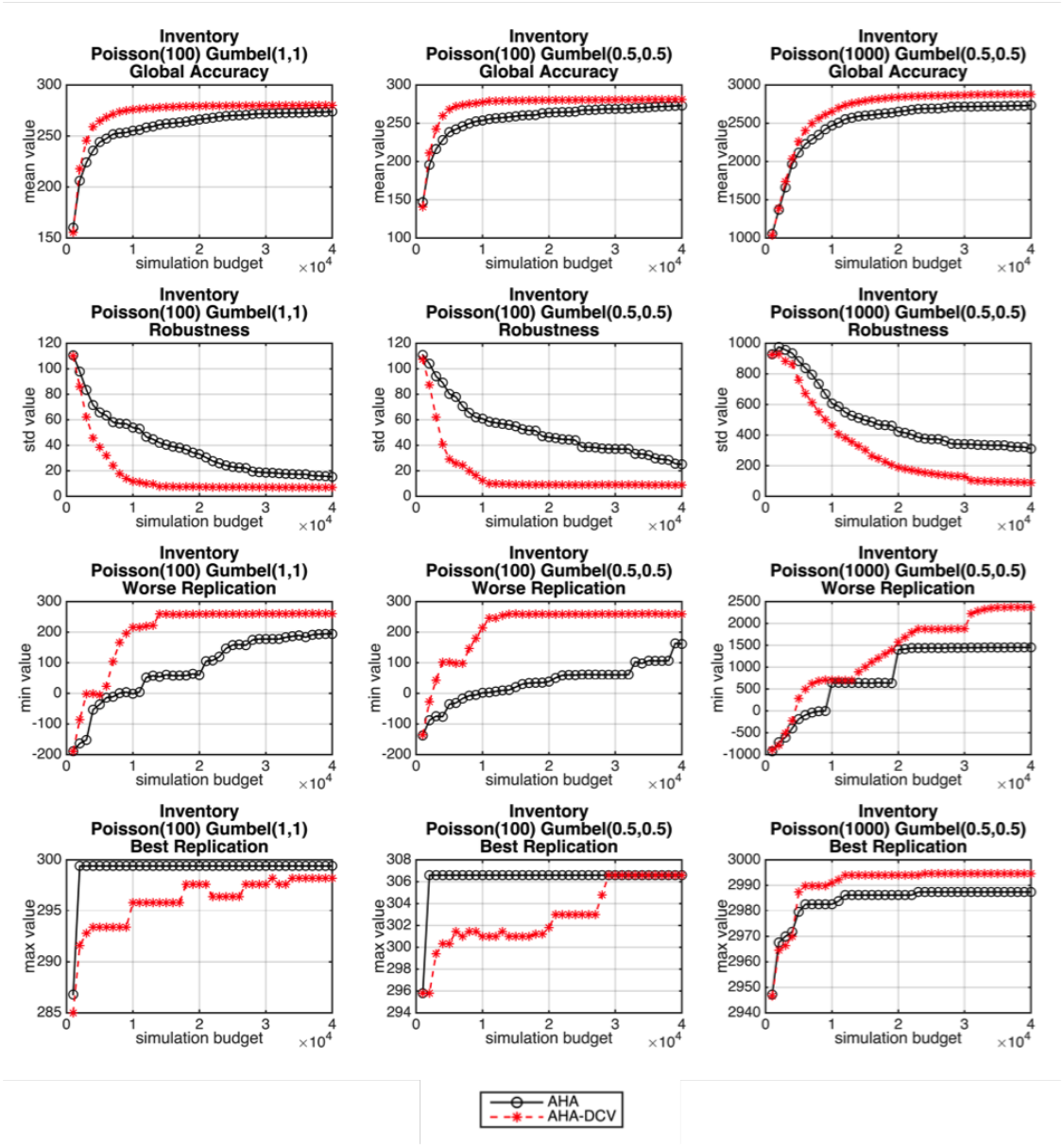

Figure 4.1: Performance measures for the Inventory Management problem

Moreover, there is a significant difference in the robustness performance, as showed in the second row in Figure 4.1. We recall that the robustness performance is computed as the standard deviation of the objective function at best solution found. In the first and second instances, the major difference occurred within an intermediary budget level ($5,000 \leq B \leq 20,000$). On the other hand, the major difference between method's robustness in the third instance is within the largest budget, with 40,000 outputs. We observe that the third instance is the one with lower variance. Further, we make a discussion on both method behaviors at the different instances.

This significant gain in robustness achieved by the AHA-DCV is directly reflected in the worst replication measures, showed in the fourth row at Figure 4.1. On average, the performance of AHA-DCV in robustness is 56%, 66% and 38% better that stand alone AHA. While best replication (fifth row at Figure 4.1) differences between the two methods are not significant (AHA is 1.0%, 1.1%, and 0.2% better on average), the performance of AHA-DCV in the worst replication draws attention (AHA-DCV is 148%, 501%, and 59% better on average respectively).

That is a notable gain in robustness. It means that, in our experiment with 100 replications, the AHA-DCV has consistently found good solutions at all replications. As opposite to it, in some replications, the solution found in stand alone AHA were very poor in comparison to AHA-DCV. It is important to note that the noise vectors randomly generated in each replication are the same for both stand alone AHA and AHA-DCV, which guarantees a fair comparison of methods within a replication.

It is also worth noting that the overall performance of stand-alone AHA has improved compared to the AHA-DCV in the third instance. Because the expected number of customers is higher at this particular instance, it is expected that the problem variance decrease. That is, a lower variance is expected in simulation experiments where the run length is larger. This fact indicates a better performance of AHA-DCV in environments with large magnitude of variance.

After evaluating the performance measures of stand alone AHA and AHA-DCV at the inventory management problem, we state that adaptive random search methods have a potential to take advantage of database control variates. We observe an improvement not only in the global efficiency at searching the optimum, but also a significantly improvement in its robustness by consistently finding a good solution at different replications.

4.3.2 Multimodal Problem

The second problem is a *multimodal function*. A multimodal optimization deals with searching multiple local and global optimal of many functions, in opposite to finding a single optimal solution. A nice survey on stochastic optimization techniques for solving multimodal optimization can be found in Dasa et al. (2011). According to this survey, it may be interesting in real application problems to switch among local and global solutions

without significantly perturbing the system performance (objective function).

> "*As practical example, consider the problem of locating the resonance points in a mechanical or electrical system (Back et al. (1997)). If the fitness function gives a resonant amplitude of the system under particular conditions, one may be interested in detecting all resonant frequencies with amplitudes above a particular threshold and not simply the frequency of greatest resonance. Frequencies of large resonance need to be identified because the designer generally wishes to minimize or maximize all such resonances, depending on the application.*"

We test the following two-dimensional multimodal function utilized in Xu et al. (2013), which is an adaptation of the multimodal function $F2$ proposed by Deb and Goldberg (1989). The function is given by:

$$g(x_1, x_2) = -\left(F(x_1) + F(x_2)\right),$$

where

$$F(x) = \frac{\sin^6(0.05\pi x)}{2^{2((x-10)/80)^2}},$$

$$x_1, x_2 \in \theta = (0, 100),\ x \text{ integer}.$$

There are 25 local optimum and the global optimal function value is -2. We propose and test three problem instances, which are:

(i) $\min_x \mathrm{E}[g(x_1, x_2) + \mathcal{N}(0, 0.3)]$, with a threshold of control mean sample size of $\eta = 100$.

(ii) $\min_x \mathrm{E}[g(x_1, x_2) + \mathcal{N}(0, 3)]$, with a threshold of control mean sample size of $\eta = 100$.

(iii) $\min_x \mathrm{E}[g(x_1, x_2) + \mathcal{N}(0, 3)]$, with a threshold of control mean sample size of $\eta = 500$.

The first row in Figure 4.2 shows the performance on global accuracy, which is accessed by the average of objective value at best solution found among replications for each of the instances. In the first instance, the global accuracy in finding a good solution of AHA-DCV is, on average, 0.9% better than stand alone AHA. Similarly to the inventory problem, both methods performed close in the instance with lower variance (first instance). This

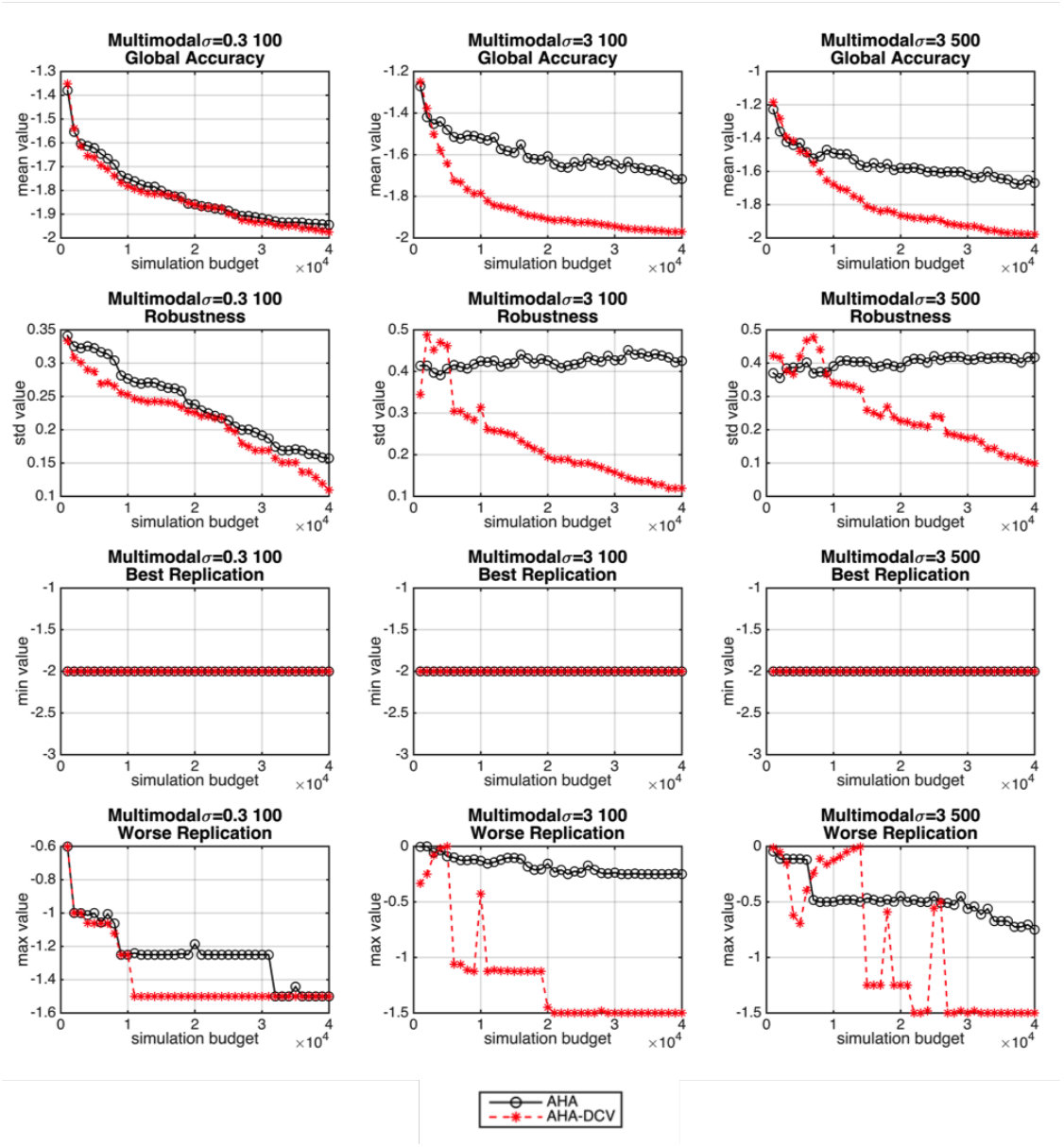

Figure 4.2: Performance measures for the multimodal problem

behavior indicates that, in objective functions with small variance, the AHA ability at finding a good solution approximates the one at AHA-DCV. In a environment with small variance, there is no space to improve efficiency in the estimation of function value via database control variates.

In the instances with larger variances (second and third), the AHA-DCV performance on global accuracy is significant better than the stand alone AHA. On average, the AHA-DCV is 15.8% and 14.4% better than stand alone AHA in the second and third instances respectively. The AHA-DCV is, at best, 17.7% and 18.0% better than stand alone AHA

in the second and third instances respectively. It occurred when the simulation budget is 12,000 and 33,000 respectively for each of these instances. It indicates that, for the multimodal problem with the specified variance of second and third instances, the stand alone AHA requires a larger simulation budget to approximate the performance of AHA-DCV in comparison to the inventory management problem.

We observe that in the experiments with smaller simulation budget, both stand alone AHA and AHA-DCV performance in global accuracy was poor. It was expected because the algorithms were not able to "walk" much. The interesting part is that, in spite of a small samples in the set-up stage, (i.e., small total number of samples at control candidate's solutions) the error raised in computing control means was not large enough to impact the AHA-DCV performance in comparison to the AHA performance. On average, AHA-CV performance on global accuracy in the first instance is 0.9% better than stand alone AHA.

Regarding robustness performance, the AHA-DCV method performed on average 9.7%, 46.0% and 40.0% better than stand alone AHA in each instance respectively. It is interesting to note that, even in the instance with lower variance, the robustness performance of AHA-DCV is 9% better. It is a gain in robustness that cannot be neglected. On the other hand, the gain in robustness in the second and third instances (which exhibits larger variance) is remarkable. Again, this pattern is similar to the ones observed in the inventory management problem. The use of database control variates to exploit the deviations on outputs of most sampled solutions and guide a variance reduction in solutions with fewer samples has a direct and positive impact on the variance of best solution found among replications.

The third row in Figure 4.2 shows the best solution found among replications for each simulation budget. In all cases, the best solution found is the optimal one. The fourth low in Figure 4.2 shows the worst solution found among replications for each simulation budget. In this measure, the performance of AHA-DCV is on average 11.2%, 569% and 96% better than stand alone AHA in each instance respectively. It indicates that the interval of function value at solution found in each simulation budget is much tighter in AHA-DCV than in stand alone AHA. The implications of these measures are very similar to the implications of the robustness measure.

We note that in third instance, we increase the required total number of samples

a solution has received to be selected as control candidate. The variance remains the same as in the second instance. Therefore in third instance, a solution is classified as control candidate if it has been observed at least 500 times. The standard threshold is $\eta = 100$. The motivation to increase this sampled size is to observe how the AHA-DCV performance is affected in the multimodal problem. The overall performance of AHA-DCV in this instance is similar to, but a little worse than the second instance. The effects are more evident within the interval of 3,000 to 15,000 of simulation budget. This behavior indicates that establishing a high threshold on minimal number of samples a solution must receive to become candidate may discard some good potential candidates. Moreover, it also indicates that the gain in accuracy when computing control means is not sufficient to overcome the loss of discarding potential control candidates.

The analysis of the multimodal problem corroborates with the final discussion on the inventory management problem. For these two canonical examples, the performance of our hybrid method overcame the one of a stand one adaptive random search method. A considerable gain in both global accuracy and robustness is observed for simulation budget larger than 4,000 outputs. Moreover, the AHA-DCV performance on multimodal problems with larger variance in comparison to the one of stand alone AHA is noteworthy.

4.3.3 Powell Singular Function

The third tested problem is a four dimensional *Powell singular function*, introduced by Powell (1962) as an unconstrained optimization problem. It is a classical test function for global optimization techniques and is considered a difficult case (for details, see Steihaug and Suleiman (2013)). The function is in the form of:

$$g(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 + 1.$$

In the discrete case, this function has two local minima, and the global optimal function value is 1. We assume that decision variable x_i ranges from -100 to 100 for $i = 1, ..., 4$. The three tested instances are:

(i) $\min_x \mathrm{E}[g(x_1, x_2, x_3, x_4) + \mathcal{N}(0, 30)]$.

(ii) $\min_x \mathrm{E}[g(x_1, x_2, x_3, x_4) + \mathcal{N}(0, 3)]$.

(iii) $\min_x \mathrm{E}[g(x_1, x_2, x_3, x_4) + \mathcal{N}(0, 3000)]$.

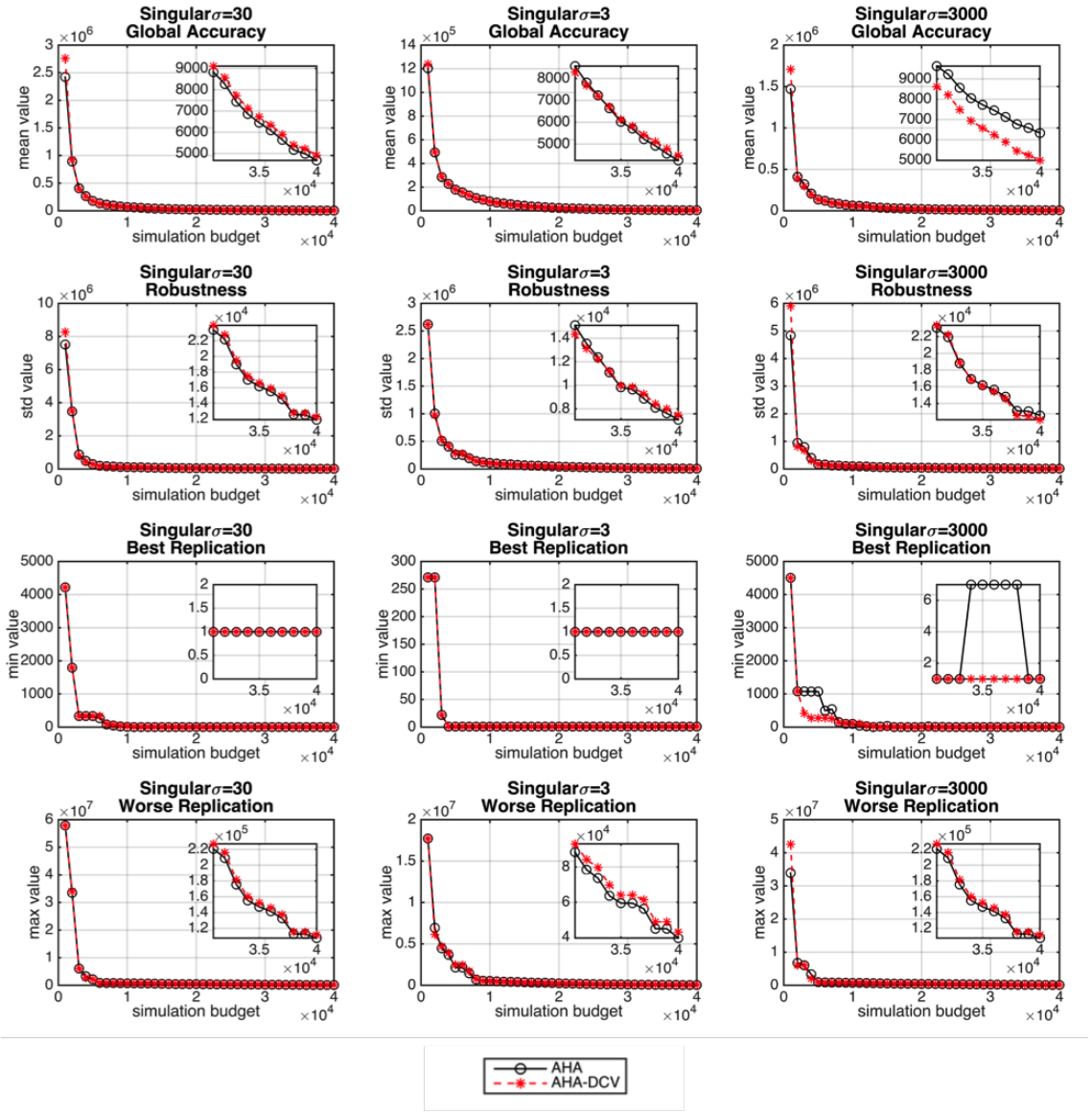

Figure 4.3: Performance measures for the Powell Singular Function

The first row in Figure 4.3 shows the plots of global accuracy for each instance. As expected, the performances of both algorithms in this problem are very similar at instances with small variances (first and second). On average, stand alone AHA performance on global accuracy is 6.4% and 0.5% better than AHA-DCV at these two instances respectively. Within a simulation budget of 1,000 (lowest) and 40,000 (highest) respectively, the stand alone AHA achieved its maximum performance in comparison to AHA-CH (12.3%

and 5.1% better respectively for each instance). On the other hand, AHA-DCV achieved its maximum performance within a simulation budget of 8,000 and 24,000 outputs respectively (6.7% and 7.5% better respectively for each instance).

It is worth to note a poorer AHA-DCV performance in global accuracy at the smallest simulation budget (1,000 outputs) for all instances in the three test problems evaluated up to this moment (inventory management, multimodal problem, and Powell singular function). This behavior is a consequence of the choice on the sample size of initial solution $n_0(\boldsymbol{x}_0) = \eta$. In a scenario with very small simulation budget, the outputs allocated to initial solution in order to guarantee at least one control candidate are worthless. That is, in the case of very low budget, is better to make $n_0(\boldsymbol{x}_0) = s$, which is the procedure in stand alone AHA. The outputs saved by taking fewer samples in initial solution are reallocated to the random search.

In the instance with higher variance (third instance), the AHA-DCV performance in global accuracy overcame the stand alone one. If we consider a range of simulation budget between 2,000 and 40,000 outputs (disregarding the lowest one of 1,000 outputs), the AHA-DCV performance is 4.3% better on average. We observe that within simulation budget of 1,000 outputs, stand alone AHA performed 16.2% better. Within a range of simulation budget of 2,000 to 10,000 outputs, AHA-DCV performed 3.8% better on average. Within a range of simulation budget of 11,000 to 20,000 outputs, AHA-DCV performed 2.8% better on average. Within a range of simulation budget of 21,000 to 30,000 outputs, AHA-DCV performed 6.8% better. Finally, within a range of simulation budget of 31,000 to 40,000 outputs, AHA-DCV performed 15.3% better on average. Therefore, AHA-DCV performance is increasing as budget increases. It is expected that, similarly to the inventory problem, the differences between both method's performance increase from a very low budget to an intermediary one. Then, from an intermediary budget to a very large one, the differences between performances are expected to decrease.

The second row in Figure 4.3 shows that the performance in robustness is very similar for both methods at all instances. On average, stand alone AHA is 0.4% better in the first instance. In the second and third instances, AHA-DCV is on average 1.5% and 0.3% better respectively at these two instances. Directly related to this behavior is the range of function value found among replications, illustrated in the third (best replication)

and fourth (worst replication) rows. Regarding the best replication, stand alone AHA performance is on average better in the first instance (2.5% better). In the second instance (lowest variance), the performances on best replication are equal (both methods were able to find the optimal solution in all range of simulation budget). In the third instance (largest variance), AHA-DCV performance is 26.1% on average better.

Regarding the worst replications, stand alone AHA performed better on average at all instances (0.2%, 1.4% and 10.7% respectively). It is interesting to note that a similar behavior is not observed in the inventory management problem and in the multimodal problem. For these latter test problems, the gains by utilizing AHA-DCV formulation in robustness are remarkable. We observe that the magnitude of variance in all instances in comparison to the function value is relatively small. A fact that corroborates with is argument is that the performance of AHA-CV is better in the instance with higher variance (third one), and also increases as function value at best solution found decreases.

4.3.4 High-Dimensional Problem

The fifth problem is a high-dimensional functions to compare stand-alone AHA and AHA-DCV performance in high-dimensional space. The function is proposed in Xu et al. (2010a), and it is designed as a test function to evaluate the impact of increasing dimension in a optimization algorithm. The high-dimensional function is of the form:

$$g(x_1, ..., x_D) = -\alpha \exp\left[-\gamma \sum_{d=1}^{D} (x_d - \lambda)^2\right].$$

The response surface has the shape of an inveted multivariate normal density function with a single optimal function value of $-\alpha$. The feasible region is the hyperbox defined by

$$x_d \in \left\{-\frac{\omega^{1/D}}{2}, \frac{\omega^{1/D}}{2}\right\}.$$

In our tests, we have: $D = 20$, $\omega = 10^{20}$, $\alpha = 10,000$, $\gamma = 0.001$ and $\lambda = 0$. The three instances are:

(i) $\min_x \mathrm{E}[g(\boldsymbol{x}) + \mathcal{N}(0, 100)]$.

(ii) $\min_x \mathrm{E}[g(\boldsymbol{x}) + \mathcal{N}(0, 1000)]$.

(iii) $\min_x \mathrm{E}[g(\boldsymbol{x}) + \mathcal{N}(0, 10000)]$.

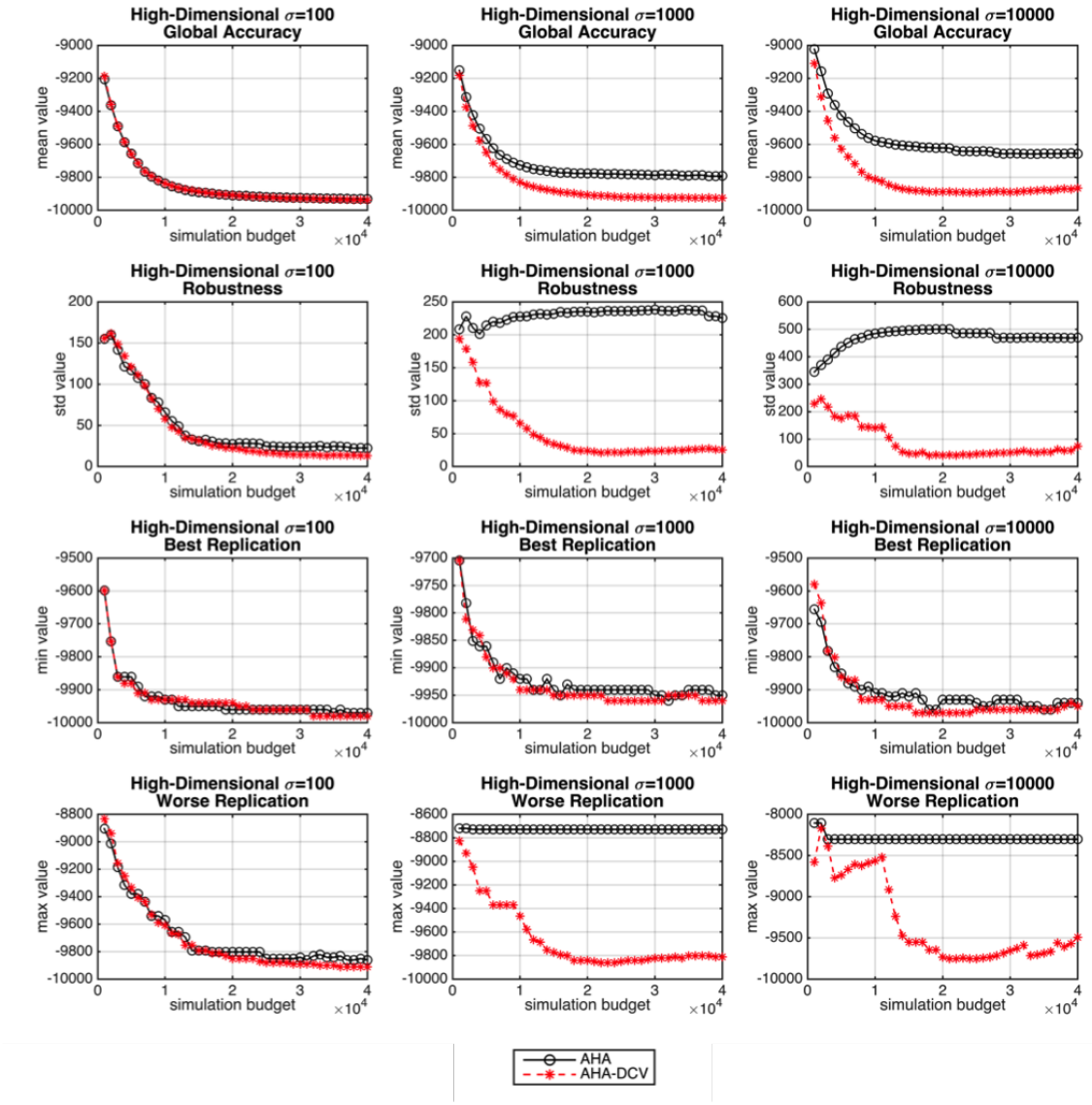

Figure 4.4: Performance measures for the High Dimensional with 20 dimensions

The results illustrated in Figure 4.4 indicates that the performance of our hybrid method can scales-up well with a high-dimensional problem. In line with previously tests, the AHA-DCV method works better than stand alone AHA in the instances with larger variance (second and third one). Similarly, both algorithms performance are very close at an instance with small variance (first one). Moreover, the main gain exhibit by AHA-CV in the high-dimensional experiment is the consistence at finding good solutions among replications (i.e., in robustness).

The first row in Figure 4.4 shows the plots of global accuracy. AHA-DCV performance

in this measure is 0.0%, 1.2% and 2.4% on average better than stand alone AHA respectively for each instance. Although AHA-DCV method generates is a consistent gain in global accuracy, the differences between both methods are not significant. However, it is worth to mention that AHA-DCV performance at very low simulation budget is better than stand alone AHA for the instances with higher variances. Such a behavior is not observed in previously experiments (inventory management, multimodal problem and Powell singular function). Therefore, it can indicate that AHA-DCV may be more suitable than stand alone AHA for high-dimensional problems with large variances even in situations with very small simulation budget.

In spite of a similar performance in global accuracy, the gain in robustness promoted by AHA-DCV is significant. The second row in Figure 4.4 shows the results of each instances at the latter measure. On average, AHA-DCV is 11.0%, 78.0% and 81.2% better than stand alone AHA respectively for each instance. While the range of solution found by stand alone AHA remains similar with the increase of simulation budget, the range of AHA-DCV solutions consistently decreases. That is, the variance reduction provided by the database control variates combined with the random search algorithm is notable even in high-dimensional problems.

A direct effect of the gain in variance reduction is illustrated by the forth row in Figure 4.4. The second and third panels show that the worse solution found by AHA-DCV among all replications approximates the optimal solution as budget increases. On the other hand, the worse solution found by stand alone AHA remains the same as budget increases. As illustrated in the third row of Figure 4.4, the best solution found by the both methods among replications are very close within all range of simulation budget.

4.3.5 High-Dimensional Multimodal Problem

In the last template problem, algorithm performances are tested on the following high-dimensional multimodal function:

$$g(x_1, ..., x_D) = -\sum_{d+1}^{D} \left\{ \alpha_1 \exp\left[-\gamma_1 \left(x_d - \lambda_1\right)^2\right] + \alpha_2 \exp\left[-\gamma_2 \left(x_d - \lambda_2\right)^2\right] \right\}.$$

In our tests, we have: $\alpha_1 = 300$, $\alpha_2 = 500$, $\gamma_1 = 0.001$, $\gamma_2 = 0.005$, $\lambda_1 = -38$ and $\lambda_2 = 56$. The three instances are:

(i) $g(\boldsymbol{x}) + \mathcal{N}(0, 1)$.

(ii) $g(\boldsymbol{x}) + \mathcal{N}(0, 1000)$.

(iii) $g(\boldsymbol{x}) + \mathcal{N}(0, 0.1)$.

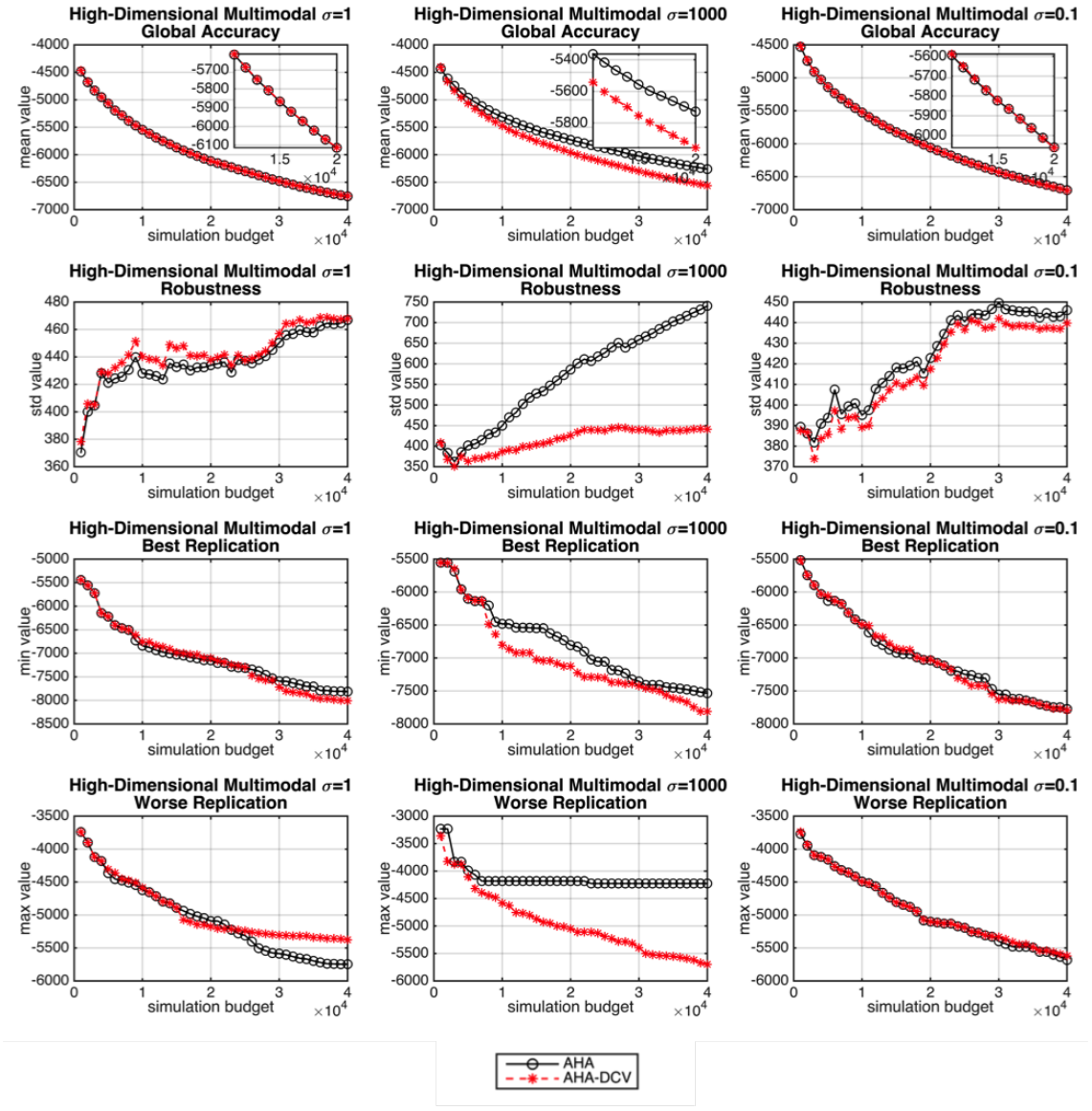

Figure 4.5: Performance measures for the High Dimensional Multimodal with 20 dimensions

The first row in 4.5 shows that the performance of both methods in global accurary are very similar among all three instances. The AHA-DCV perforamance is 3.8% on average

better than the one in stand alone AHA in the second instance, which ehibits a variance much larger than the first and thrid instances. The differences in performance at the first and third instances are lower that 0.001% on average. AHA-CV is, at best, 0.05%, 4.8% and 0.01% better than stand alone within simulation budgets of 30,000, 40,000 and 35,000 respectively for each instances. AHA is at best 0.4%, 0.6% and 0.03% better than stand alone within simulation budgets of 1,000, 1,000 and 1,000 respectively for each instances.

Likewise in global accuracy measure, both methods performance in robustness at the instances with lower variances are very similar. Stand alone AHA performance in robustness is 1.6% on average better than AHA-DCV in the first instance, and AHA-DCV performance is 1.5% on average better than stand alone AHA in the thrid instance. In the instance with higher variance (second instance), AHA-DCV performance in robustness is 27% better on average. Although the gain is robustes in this example is not as large such as in previously examples, the AHA-DCV ability at consistently finding a good solution among replication is significative in the instance with larger variance.

Following the trend on all examples, the gain in robustness is a consquecence of AHA-DCV capacity to find good solutions among all replications. We observe that, as reported in the plots on third and fourth rows in Figure 4.5, the best solution found among replications by both algorithms are very similar. The main difference between methods is on the worst solution found among replications at the instance with large variance (second one). In this instance, the worst solution found by the AHA-DCV method is, on average, 20% better than the one found by the stand alone AHA.

All observations on the current test problems are in line with the other canonical examples. We remark that AHA-DCV overall performance in the high-dimensional multimodal problem overcame stand alone AHA. It indicates that AHA-DCV formulation can be benefitial even in a high-dimensional multimodal problem with considerable variance.

4.4 Final Discussion

In this Chapter, we present a novel formulation for improving the efficiency of a random search method in solving stochastic optimization problems. We provide a framework that allows the interconnection between a random search method (adaptive hyperbox algorithm

- AHA) and a flexible variance reduction technique (database control variates). The result is a hybrid method labeled as AHA-DCV. The essence of our approach lies in the use of outputs from already sampled solutions to better estimate the response surface at solutions to be sampled. The use of such available information, which usually is discarded in most common techniques, has proven to be very useful for reduction the variance of best solution found by the algorithm.

There is a consistent evidence throughout the canonical experiments discussed at Section 4.3 that our hybrid method is more suitable than stand alone AHA for solving stochastic optimization problems with large variances. The AHA-CV performance on global accuracy (i.e., the ability of algorithm at finding good solutions on average) overcame the stand alone AHA is the vast majority instances for all tests, in particular at those with larger variances. Moreover, the gains in robustness (i.e., the ability of algorithm at consistently find a good solution among replication) are notable. It indicates that the range of solutions found by the AHA-CH is significantly tighter than the one in stand alone AHA, in particular for the worst replication.

We remark that the hybrid formulation introduced in this Chapter can be easily adapted to a vast of other stochastic optimization tools. However, its procedure may vary depending on the underlying method. Key elements for adapting our formulation to other stochastic optimization tool are: the sample procedure, the mechanism of optimal search, and if the method addresses continuous or discrete. On going research include adapting the AHA-DCV to simultaneous perturbation stochastic approximation, response surface methodology and finite-difference stochastic approximation.

One possible limitation of our hybrid approach to stochastic optimization methods is the requirement of using common random numbers among noisy sources. Depending on problem complexity and nature, it may not be possible to use it in all sources of noise, which can have a serious impact on the correlation magnitude between sampled solutions. Another limiting aspect is the computational cost of storing the vectors of noise to enable retrieval at each algorithm iteration. It may slow down computational time and may require a large memory capacity, again depending on problem complexity and nature. It may be that storing just initial seed of random generation might be enough to allow the use of common random numbers.

Although the discussed experiments provide rich insights, future research include the analytical analysis, which is needed to deeper understand the behavior of a stochastic optimization method embedded with a variance reduction technique. The development of analytical properties at not only AHA-DCV, but also other stochastic optimization methods embedded with database control variates must be investigated. We suggest evaluating the hybrid formulation at a very simple objective function, such as a quadratic one, to observe fundamental behaviors.

Future theoretical analysis regarding the proposed hybrid formulation is twofold. First, it is of great interest to derive the interval of confidence of the estimated function value at sampled solutions. It is expected that such interval can be assessed by the mean squared error of database control variates in (4.6). However, in the case of adaptive random search algorithms, the sample size N in setup stage of database control variates can dynamically change as the algorithm evolves. That is, N is not constant and can increase with iterations because solutions can be revisited. Therefore, we must investigate how such interval of confidence is affected as the random search algorithm evolves.

Second, the complexity of the hybrid algorithm is to be derived. The AHA complexity is $O(|\mathcal{L}(k)| \log(|\mathcal{L}(k)|))$ (see Xu et al. (2013) for details). Therefore, it is associated to the size of the set of unique sampled solutions. On the other hand, the complexity of the database control variates algorithm is associated with the number of controls (denoted by I), sample size of setup stage (denoted by $n_k(\boldsymbol{x}_C V^{(i)})$), and - likewise AHA - with the cost of computing an output $Y(\boldsymbol{x}, \boldsymbol{w})$. We must investigate the connections between the two algorithms to derive the order of complexity of the hybrid one.

We now make a discussion on specific challenges of the AHA-DCV formulation. First, one must investigate if is there a more intelligent manner to choose initial point than a pure random choice. The initial solution may the unique control candidate available if solutions are not often revisited. Therefore, it would be good if initial solution were chosen taking into consideration the correlation within the feasible space. Secondly, it may be beneficial to investigate the choice on input parameters (initial sample size $n_0(\boldsymbol{x}_0)$, sample size threshold η and correlation threshold ϱ). For example, it may be interesting to let η and ϱ be dynamically chosen through iterations. Finally, a straightforward improvement in AHA-DCV is to update current estimates in all/some solutions if a control candidate

receive more samples. When more information about controls is available, the accuracy on control means is improved. Therefore, the variance on estimates of the function value is expected to decreases.

Chapter 5

Optimization in High-Dimensional Spaces

Let us begin by considering deterministic optimization algorithms. For example, consider the basic *steepest descent* algorithm and a sample minimization problem, specified as follows: Find $\boldsymbol{x}^* \in \mathbb{R}^d$

$$J(\boldsymbol{x}^*) = \min_{\boldsymbol{x} \in \mathbb{R}^d} J(\boldsymbol{x}).$$

To simplify the discussion, assume such a global minimum exists and is unique. The analysis of how well the steepest descent algorithm (and other minimization algorithms) performs is often expressed in terms of the *rate of convergence* of the algorithm to the minimum. In essence, this rate of convergence tells us how the algorithm behaves once it has reached the vicinity of the optimum. The rate of convergence criterion does not give us an idea of how the algorithm moves from an arbitrary starting point to somewhere close to the optimum. One may argue that this latter criterion is an important measure of the performance of the algorithm, perhaps more important than its rate of convergence in practical contexts of specific applications. Let us refer to this second measure of performance as *finite time* measure in contrast to the *asymptotic* convergence rate measure. Furthermore, we would be interested in understanding how the dimension of the search space, namely d, influences such a finite time measure.

Note that one possible reason for the selection of the rate of convergence criterion to

assess the performance of optimization algorithms may be that this rate depends only on the behavior of the cost function *in the vicinity of the optimum* while a finite time measure depends on the global features of the cost function that are harder to characterize and analyze.

Now let us turn to stochastic optimization problems and consider the counterpart of the steepest descent algorithm in this context, namely the *stochastic approximation* algorithm. In this case the gradient of the cost function can only be estimated and it always involves some estimation noise. The setting, similar to the one we considered in the deterministic case, can be described as follows: We are looking for $\boldsymbol{x}^*$ in the search space $\mathbb{R}^d$ such that

$$J(\boldsymbol{x}^*) = \min_{\boldsymbol{x} \in \mathbb{R}^d} E[L(\boldsymbol{x}, W)],$$

where J, the cost function, is sufficiently well-behaved and the expectation is with respect to the stochastic vector W. We assume that at any $\boldsymbol{x}$ we can estimate the gradient $\nabla J(\boldsymbol{x})$ of the cost function. Let $H(\boldsymbol{x}, W)$ be this estimator. Then, the stochastic optimization algorithm is defined as follows:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \epsilon_n H(\boldsymbol{x}_n, W_{n+1}),$$

where, to ensure convergence, the sequence $\{\epsilon_n; n \geq 1\}$ is required to satisfy (i) $\epsilon_n \geq 0$, (ii) $\sum_{n=1}^{\infty} \epsilon_n = \infty$, and (iii) $\sum_{n=1}^{\infty} \epsilon_n^2 < \infty$.

Note that condition (iii) implies that $\lim_{n \to \infty} \epsilon_n = 0$. Therefore, the principle method of dealing with estimation noise of the gradient in this approach is to gradually diminish the effect of estimation noise using coefficients ϵ_n. Condition (ii) suggests that we should not be reducing ϵ_n too fast so as to allow for the signal in the gradient to get us to the optimum. Condition (iii) states that we should not be doing it too slowly so as to ensure it does not lead us astray.

The key motivation in the work in this Chapter is to gain some seminal understanding of the effect of the dimension of the search space on the finite time behavior of gradient-based optimization algorithms. Given that in recent years in many application domains

ever more high dimensional problems are being considered, we believe such a study can be beneficial. Our study is seminal in nature and to that end we consider the simplest possible optimization problems that nonetheless captures some key elements of the influence of the dimension of the search space on the performance of the algorithm.

The Chapter is organized as follows. In Section 5.1, we present the related literature on optimization in high dimensions. Two elementary algorithms are introduced in Section 5.2, and the main analyzes in the effects of dimension on their performances are derived. A final discussion and directions for future research are provided in Section 5.3.

5.1 Related Literature

In Shan and Wang (2010), a nice survey on modeling and optimization strategies to solve high-dimensional problems can be found. This survey pointed out that research on this topic is scarce and sporadic, partially due to its difficulty on the problem itself.

Strategies to tackle high-dimensionality include parallel computing (e.g., Wang et al. (2013)), increasing computer power, reducing design space, screening significant variables (e.g.,Chu et al. (2011b)), decomposing design problems into subproblems (e.g., Gardeux et al. (2011) and Regis and Shoemaker (2013), mapping (e.g., Xu et al. (2013), (Jeff Hong and Nelson (2006) and Hong et al. (2010)), and visualizing the variable/design space.

Other work on high-dimensional problems woth citing are: the robust optimization study of Bandi and Bertsimas (2012); a differential evolution approach based on opposition-based learning of Wang et al. (2011); and the swarm optimization approaches in Chu et al. (2011a), Jia et al. (2011) and Imanian et al. (2014). Theses strategies tackle from different angles the difficulties caused by the high-dimensionality.

According to Shan and Wang (2010), a deeper understanding of the high-dimensional space is felt needed to develop more robust models. The limits of our imagination to conceive more than 3-dimensional spaces hinders the development of intuitive sampling approaches, and also hinders our understanding of such a vast space. In the latter survey, two important questions are raised: (i) are there other properties and/or knowledge about a high-dimensional space?; and (ii) and how to design sampling and modeling techniques to take advantage of such a property?

We aim to investigate the research gap highlighted in Shan and Wang (2010), that a more in-depth theoretical study of high-dimensional space properties can guide the development of more generic/sophisticate optimization techniques.

5.2 Elementary Algorithms

Consider the following simple deterministic optimization problem:

$$\min J(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}, \tag{5.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{Q}$ is a positive definite matrix. Problem (5.1) is an unconstrained deterministic quadratic problem with optimal solution at $\boldsymbol{x}^* = \boldsymbol{0}$. Assume that in this case we use finite-difference estimates of the gradient to be more in line with those more general cases where computing the gradient of the cost function analytically is not feasible and finite difference estimates are used. Note that *the cost of estimating the gradient* corresponds to $d + 1$ function evaluations that increases linearly with d. We begin with an algorithm that requires a minimum number of function evaluations, namely one, for determining a potential direction for minimizing the cost function.

5.2.1 Algorithm 1 - Single Sample

We begin with the following simple random sampling algorithm:

Algorithm 1 - Single Sample

Step 0. Set iteration count $k = 0$. Let $\boldsymbol{x}_0$ be a starting point. Set $J_0 = J(\boldsymbol{x}_0)$.

Step 1. Randomly generate $\boldsymbol{z}$, a jointly Gaussian random variable with mean $\boldsymbol{0} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{I}$ (identity $d \times d$ matrix). Then, the vector $\boldsymbol{z}/||\boldsymbol{z}||$ is uniformly distributed over the surface of a unit d-ball.

Step 2. Let α be the step size of the algorithm. Set

$$J_k^+ = J\left(\boldsymbol{x}_k + \alpha \frac{\boldsymbol{z}}{||\boldsymbol{z}||}\right).$$

2.1. If $J_k^+ < J_k$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{z}/||\boldsymbol{z}||$, and $J_{k+1} = J_k^+$.

2.2. Else, set

$$J_k^- = J\left(\boldsymbol{x}_k - \alpha \frac{\boldsymbol{z}}{||\boldsymbol{z}||}\right).$$

2.3. If $J_k^- < J_k$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \boldsymbol{z}/||\boldsymbol{z}||$, and $J_{k+1} = J_k^-$.

2.4. If $J_k^+, J_k^- \geq J_k$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$, and $J_{k+1} = J_k$.

Step 3. Set $k = k + 1$ and go to Step 1.

Thus, the algorithm moves towards the optimal point by generating and testing random points on the surface of a d-dimensional ball of radius α centered at the current point $\boldsymbol{x}_k$. If the sampled new point is not better than the current one, the opposite direction of movement is also tested. If the opposite point also shows no improvement, it is said that the algorithm does not move and a new direction of movement $\boldsymbol{z}/||\boldsymbol{z}||$ is generated.

Probability of Movement And Its Length

We now evaluate the probability of moving to a better solution at an iteration of the algorithm. Assume a current point $\boldsymbol{x} \in \mathbb{R}^d$, let $\boldsymbol{z}$ a d-vector of i.i.d. standard normal random variables, and $\boldsymbol{w} = \frac{\boldsymbol{z}}{||\boldsymbol{z}||}$. Let us call the probability of finding a better solution at the current point as $P(moving)$. This probability is given by:

$$\begin{aligned} P(moving) &= P\left(J\left(\boldsymbol{x} + \alpha \boldsymbol{w}\right) < J(\boldsymbol{x})\right) + P\left(J\left(\boldsymbol{x} - \alpha \boldsymbol{w}\right) < J(\boldsymbol{x})\right) \\ &= P\left(\boldsymbol{x}^\top Q \boldsymbol{w} < -\frac{\alpha}{2}\right) + P\left(\boldsymbol{x}^\top Q \boldsymbol{w} > \frac{\alpha}{2}\right) \\ &= P\left(\frac{\boldsymbol{x}}{||\boldsymbol{x}||}^\top Q \boldsymbol{w} < -\frac{\alpha}{2||\boldsymbol{x}||}\right) + P\left(\frac{\boldsymbol{x}}{||\boldsymbol{x}||}^\top Q \boldsymbol{w} > \frac{\alpha}{2||\boldsymbol{x}||}\right). \end{aligned}$$

Consider the simple case of $Q = I$. In this case $\frac{\boldsymbol{x}}{||\boldsymbol{x}||}^\top Q \boldsymbol{w}$ is the cosine of the angle between vectors $\boldsymbol{x}$ and $\boldsymbol{w}$. Let θ denote this angle. Note that in the more general case of a positive definite matrix Q, with a change of coordinate system, we can interpret $\frac{\boldsymbol{x}}{||\boldsymbol{x}||}^\top Q \boldsymbol{w}$ as the cosine of the angle between two similarly relevant vectors. In the simple case of $Q = I$,

the above derivation implies that the probability of not moving at an iteration is given by

$$-\frac{\alpha}{2||\boldsymbol{x}||} \leq \cos\theta \leq \frac{\alpha}{2||\boldsymbol{x}||}. \tag{5.2}$$

Clearly, both α and $||\boldsymbol{x}||$ do not depend on the problem dimension. Therefore, the key element to why this optimization algorithm may be very inefficient in high-dimensional problems may be the effect of dimension on $\cos\theta$.

Furthermore, in the simple case of $Q = I$, the amount of movement, once a better solution is identified, is given by

$$h = \alpha \cos\theta. \tag{5.3}$$

Experimental Results

The first experiment we conduct intends to illustrate the effects of increasing dimension in $\cos\theta$. We make a total of 21 experiments combining different instances of problem dimension ($d = 2, 3, 4, 5, 10, 100, 1000$) and starting points ($||\boldsymbol{x}_k|| = 1, 100, 10000$). Without loss of generality, we assume $Q = I$. For each combination of dimension and starting point, we run 10,000 replications of a single iteration of Algorithm 1. That is, a replication consists in starting at $\boldsymbol{x}_k$ and computing $\boldsymbol{x}_{k+1}$. Specifically, the algorithm generates a random direction of movement $\boldsymbol{w}_k$. The goal of each experiment is to compute the probabilistic distribution of the cosine of the angle θ between vector $\boldsymbol{x}$ and $\boldsymbol{w}$.

Figure 5.1 shows the histograms of $\cos\theta$ for each experiment. The dimension d of experiments increases with rows, and the distance between current and optimal solution ($||\boldsymbol{x}_k||$) increases with columns. The first behavior worth mentioning is that, clearly, $\cos\theta$ does not depend on the distance between the current solution and the optimal one. As can be seen, the probabilistic distribution of $\cos\theta$ remains the same as we change $||\boldsymbol{x}_k||$ (columns). It implies that the amount of movement the algorithm can achieve in a single iteration does not depend on how far from optimum the current solution is. We make a thorough discussion on this topic in further experiments.

On the other hand, the distributions reported in the histograms change significantly as problem dimension increases (rows). In the case of $d = 2$, we observe a larger frequency of $|\cos\theta|$ close to 1. Such a frequency is consistently decreasing as dimension increases.

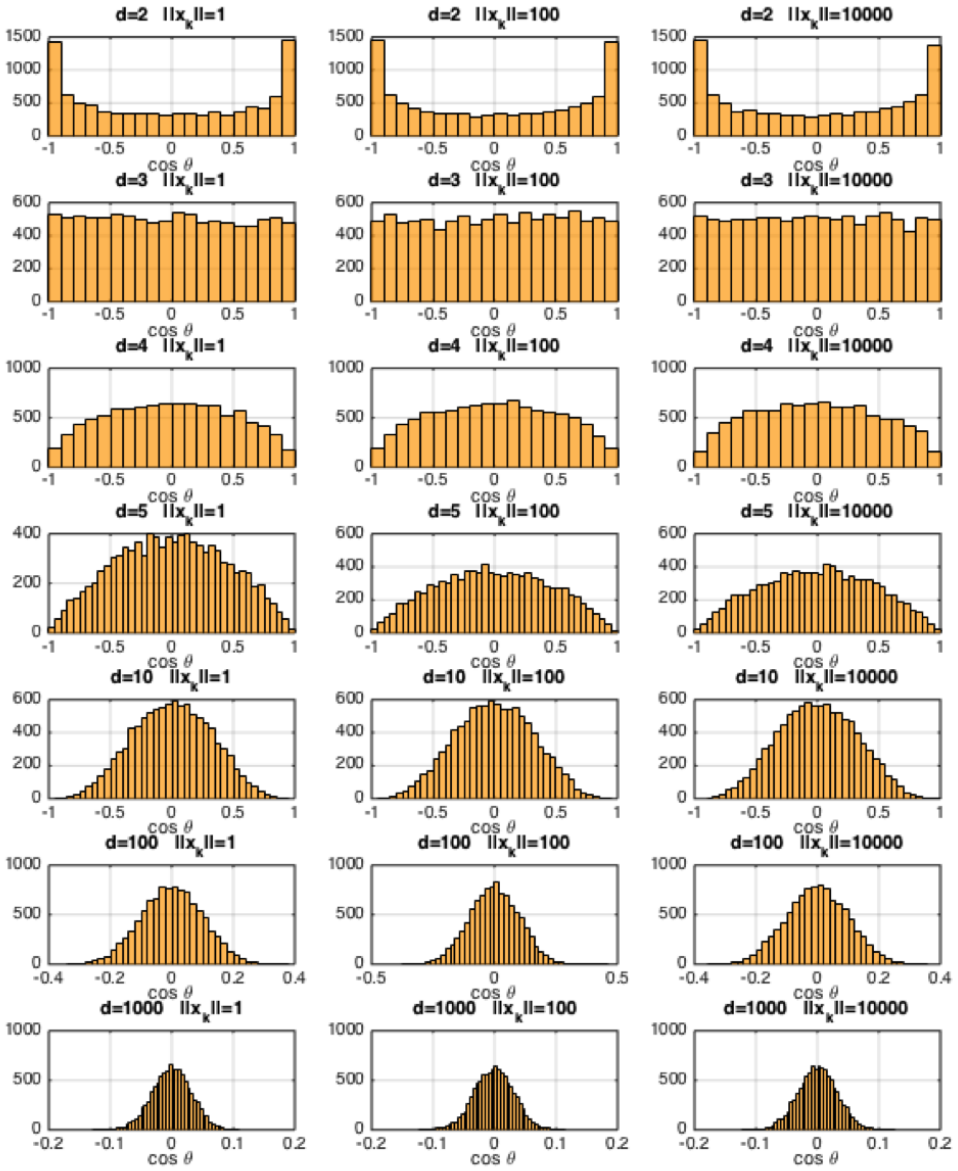

Figure 5.1: $\cos\theta$ at different dimensions (d) and at different distance from optimal point ($||\boldsymbol{x}_k||$)

Between $d = 2$ and $d = 4$, there is a change in the shape of histograms. It can be explained by considering the area on the surface of a d-dimensional ball centered at $\boldsymbol{x}_k$ for which condition (5.2) is satisfied. When the dimension is increased, the proportion of the area where the algorithm does not find a better solution over the area it does find increases. As a consequence, the higher frequency of $|\cos\theta|$ shifts from 1 towards 0 as dimension increases.

This behavior has two important implications to the performance of algorithms in

problems with higher dimensions. First, the amount of movement that an algorithm can achieve at a single iteration, which is given by equation (5.3), decreases as dimension increases. That is, the algorithm becomes slower in high-dimensional spaces. Second, the probability of moving, which is given by equation (5.2), also decreases as dimension increases. That is, the ability of the algorithm in finding better solution becomes less efficient as dimension increases.

	$\|\|x_k\|\| = 1$	$\|\|x_k\|\| = 100$	$\|\|x_k\|\| = 10,000$
$d = 2$	33.4%	0.4%	0.0%
$d = 3$	49.8%	0.5%	0.0%
$d = 4$	60.2%	0.7%	0.0%
$d = 5$	68.6%	0.7%	0.0%
$d = 10$	88.2%	1.2%	0.02%
$d = 100$	100.0%	4.0%	0.02%
$d = 1000$	100.0%	12.9%	0.08%

Table 5.1: Probability of not moving at different dimensions and points after 10,000 replications

Table 5.2.1 shows the probability of not moving, computed by simulation in each experiment. As expected, the probability of not moving (i.e., the probability of not finding a better solution) increases as the dimension increases. We also note that the increase in the probability of not moving is inversely proportional to the current distance from optimum ($||\boldsymbol{x}_k||$), which is a direct result from equation (5.2).

We observe that in the particular cases where $d = 100$ and $d = 1000$, the probability of moving in the vicinity of the optimum is so small that the algorithm was not able to improve the current solution after the 10,000 replications. In the next experiment, we discuss more on the reasons why the probability of not moving increases as the algorithm gets in the vicinity of the optimum.

Figure 5.2 shows the probabilistic distribution of the amount of movement of each experiment, which can be accessed by equation (5.3). In the experiments, we consider $\alpha = 1$ for simplicity. It is interesting to note that, when the current solution is not in the vicinity of the optimal, the length $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k+1}||$ has a direct connection to $\cos\theta$. It is the projection of the direction of movement $\boldsymbol{w}_k$ onto vector $\boldsymbol{x}_k$.

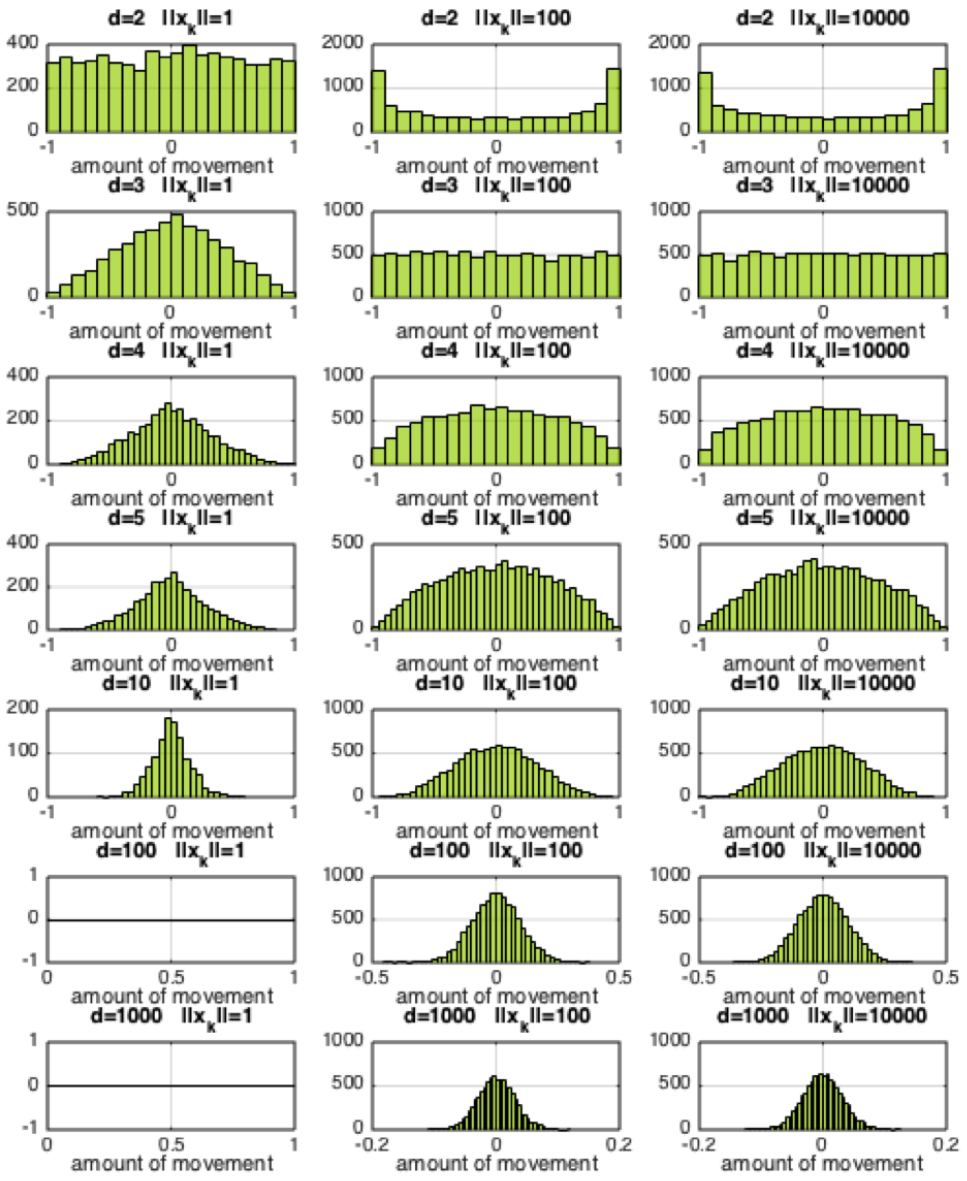

Figure 5.2: The amount of movement given by equation (5.3) at different dimensions (d) and at different distance from optimal point ($||\boldsymbol{x}_k||$). Negative values are with respect to negative $\cos\theta$. Movements equal to zero (i.e., when the algorithm does not move) are disregarded in these panels.

However, as the algorithm approaches the vicinity of the optimum, the distribution of observed amount of movement changes. This can be easily explained by the fact that h is large enough so that the d-dimensional ball with radius h centered at $\boldsymbol{x}_k$ includes the optimal point $\boldsymbol{x}^*$. Therefore, if the distance between current point and the optimal one is smaller than the radius of the d-dimensional ball, both the probability of moving and the amount of movement decrease. In fact, it is easy to show that if the distance from current

point to the optimal one is less than $h/2$, (i.e., $||\boldsymbol{x}_k|| \leq h/2$ for $Q = I$), then the algorithm is not able to move.

Figure 5.3 shows the probabilistic distribution of the resulting movement torward the optimum achieved by the algorithm in a single iteration (i.e., $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k+1}||$). We start by observing that such a movement is equal to the absolute value of equation (5.3). It is a direct consequence of testing the opposite direction of movement (i.e., checking if $\boldsymbol{x}_k - \boldsymbol{w}_k$ is a better solution than $\boldsymbol{x}_k$). Moreover, we remark that the results illustrated in this figure include the cases in which the algorithm does not move. That is, when $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$.

In each experiment, we provide the average (μ) and standard deviation σ of $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k+1}||$. As one can observe, the resulting movement achieved by the algorithm does not change when the current solution is not in the vicinity of the optimum. On the other hand, the probability of moving and amount of movement when closer to the optimal point are small.

Now, we draw attention to the patterns in the second and third columns, when the current point is far away from the optimum ($||\boldsymbol{x}_k = 100$ and $\boldsymbol{x}_k|| = 10,000$). We observe that when the problem dimension is $d = 2$, the probability that the algorithm achieves a larger step (i.e., $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k+1}||$) towards the optimum is considerably larger than that of achieving a small step. However, as the dimension gets higher, the range of the resulting movement becomes tighter. Furthermore, as dimension increases, the chance of achieving a small step becomes larger. Finally, there is an increasing frequency of cases in which the algorithm does not move (i.e., $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k+1}|| = 0$) as the problem dimension increases.

As a consequence, the algorithm becomes slower by taking small steps towards the optimum (although the step size is constant, $\alpha = 1$), and less efficient in finding a better solution as dimension increases.

5.2.2 Algorithm 2 - Multiple Samples

In the second algorithm, at each iteration, we consider evaluating the function J at a finite number of randomly selected points on the surface of a unit sphere in $\mathbb{R}^d$ and selecting the best direction for movement based on the collected information. The best direction corresponds to the steepest descent direction of a linear approximation to J given the collected information.

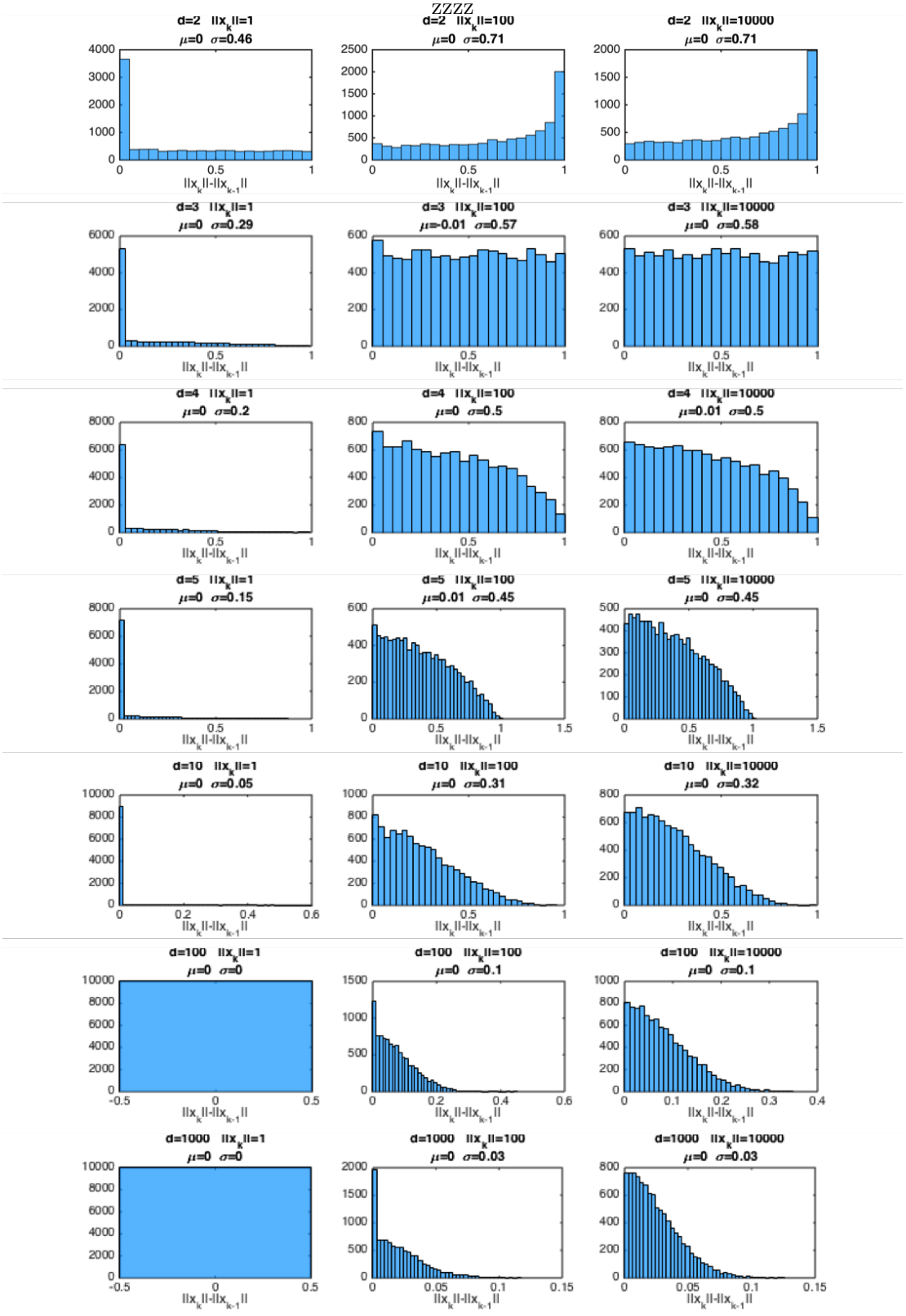

Figure 5.3: The amount of movement at different dimensions (d) and at different distance from optimal point ($||\boldsymbol{x}_k||$). It is the version of above figure with movements equal to zero.

More specifically, let $\boldsymbol{x}$ be a point in $\mathbb{R}^d$. Let $J = J(\boldsymbol{x})$. Assume m random points are selected on the unit sphere in $\mathbb{R}^d$ around $\boldsymbol{x}$. Let $\boldsymbol{w}_i = \frac{\boldsymbol{z_i}}{||\boldsymbol{z_i}||}$ for $i = 1, \cdots, m$, where $\boldsymbol{z_i},\ i = 1, \cdots, m$ are i.i.d. d-dimensional jointly normal random variables with zero mean and covariance equal to identity matrix. Let

$$J^{(i)} = J(\boldsymbol{x} + \boldsymbol{w}_i), \quad \text{and} \quad a_i = J^{(i)} - J, \quad i = 1, \cdots, m.$$

We consider points in $\mathbb{R}^d$ that are of the form

$$\boldsymbol{x} + \sum_{i=1}^{m} c_i \boldsymbol{w}_i,\ i = 1, \cdots, m \quad \text{and} \quad \| \sum_{i=1}^{m} c_i \boldsymbol{w}_i \| = 1.$$

The linear/affine approximation to the function J on the above points is given by

$$L(\boldsymbol{x} + \sum_{i=1}^{m} c_i \boldsymbol{w}_i) = J + \sum_{i=1}^{m} c_i (J^{(i)} - J) = J + \sum_{i=1}^{m} c_i a_i.$$

Note that

$$L(\boldsymbol{x}) = J(\boldsymbol{x}), \quad \text{and} \quad L(\boldsymbol{x} + \boldsymbol{w}_i) = J(\boldsymbol{x} + \boldsymbol{w}_i),\ i = 1, \cdots, m.$$

We now look for a feasible direction that provides the smallest value of the function J. Given that we do not know the values of function J in all directions that can be spanned by $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m$, we use the direction that minimizes the linear approximation, namely, we consider the following minimization problem:

$$\min_{c_1, \cdots, c_m} \{ J^{(0)} + \sum_{i=1}^{m} c_i a_i; \quad \| \sum_{i=1}^{m} c_i \boldsymbol{w}_i \| = 1 \}.$$

The constraint $\| \sum_{i=1}^{m} c_i \boldsymbol{w}_i \| = 1$ can be written as

$$\| \sum_{i,j=1}^{m} c_i c_j \langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle = 1 \|.$$

Define an $m \times m$ matrix M by $M_{ij} = \langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle$ and m-dimensional vectors $\boldsymbol{c} =$

$(c_1, \cdots, c_m)^T$ and $\boldsymbol{a} = (a_1, \cdots, a_m)^T$. To find the best direction, then, we need to solve

$$\min_{\boldsymbol{c}} \{\boldsymbol{c}^T \boldsymbol{a}; \quad \boldsymbol{c}^T M \boldsymbol{c} = 1\}.$$

Using a Lagrange multiplier λ we can transform the above constrained optimization to an unconstrained one given by

$$\min_{\boldsymbol{c}} \{\boldsymbol{c}^T \boldsymbol{a} + \lambda(\boldsymbol{c}^T M \boldsymbol{c} - 1)\}.$$

Differentiating with respect to $\boldsymbol{c}$, we have

$$\boldsymbol{a} + 2\lambda M \boldsymbol{c} = 0.$$

Therefore,

$$\boldsymbol{c}^* = \alpha M^{-1} \boldsymbol{a},$$

for a constant α. To find the constant, note that we have

$$\boldsymbol{c}^{*\boldsymbol{T}} M \boldsymbol{c}^* = 1 \quad \Rightarrow \quad \alpha^2 \boldsymbol{a}^T M^{-1} M M^{-1} \boldsymbol{a} = 1 \quad \Rightarrow \quad \alpha^2 \boldsymbol{a}^T M^{-1} \boldsymbol{a} = 1.$$

In the above derivation, we have used the fact that M^{-1} is a positive definite matrix and hence its transpose is equal to itself, i.e., M^{-1}. Therefore, we have

$$\boldsymbol{c}^* = -\frac{1}{(\boldsymbol{a}^T M^{-1} \boldsymbol{a})^{1/2}} M^{-1} \boldsymbol{a}.$$

The vector $\boldsymbol{c}^*$ identifies the direction of steepest descent of the linear approximation function L and will be used in *Algorithm 2* given below.

Algorithm 2 - Multiple Samples

Step 0. Set iteration counter $k = 0$. Let $\boldsymbol{x}_0$ be a starting point provided by the user. Compute $J^{(0)} = J(\boldsymbol{x}_0)$.

Step 1. Randomly generate m d-vectors $\boldsymbol{w}_i = \boldsymbol{z}_i / ||\boldsymbol{z}_i||$ for $i =, \cdots, m$, where $\boldsymbol{z}_i$ is also a

d-dimensional vector of jointly normal random variables with mean $\mathbf{0} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{I}$ (identity $d \times d$ matrix).

Step 2. Take m samples at each direction $\boldsymbol{w}_i$. That is, compute

$$J^{(i)} = J(\boldsymbol{x}_k + \boldsymbol{w}_i), \qquad i = 1, \cdots, m.$$

Step 3. 3.1. Compute $a_i = J^{(i)} - J^{(0)}$, and $M_{ij} = \langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle$, $i, j = 1, \cdots, m$.

3.2. Compute c^* as follows.

$$\boldsymbol{c^*} = -\frac{1}{(\boldsymbol{a}^T M^{-1} \boldsymbol{a})^{1/2}} M^{-1} \boldsymbol{a}.$$

3.3. Let the best point function value found by the linear approximation be:

$$\boldsymbol{x}_k^* = \boldsymbol{x}_k + \sum_{i=1}^{m} c_i^* \boldsymbol{w}_i.$$

Step 4. Compute $J^* = J(\boldsymbol{x}_k^*)$. If we have found a better solution, that is, if $J^* - J^{(0)} < 0$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k^*$, $J^{(0)} = J^*$. Else, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$. Update $k = k + 1$, and go to step 1.

Experimental Results

To illustrate the effects of dimension d and sample size m on Algorithm 2, we conduct very similar experiments to the ones regarding Algorithm 1. We make a total of 16 experiments combining different problem dimensions ($d = 5, 10, 50, 100$) and different sample sizes (m) according to the dimension. We assume $Q = I$, and run 10,000 replications of a single iteration. The starting point is constant for all experiments, with $||\boldsymbol{x}_k|| = 10$. The goal of each experiment is to compute the resulting distance $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k-1}||$ achieved by the algorithm towards the optimal point $\boldsymbol{x}^* = \mathbf{0}$.

We remark that our motivation to conduct a comparison between the two algorithms lies on a deeper understanding of the search for the optimum in high-dimensional spaces when only noisy estimates of the best direction of movement is available. Essentially,

this is similar to the setting of stochastic optimization, in which the gradient can only be estimated and it always involves some estimation noise. One of our key objectives here is to provide some insight on why the performance of stochastic optimization algorithms deteriorate in high dimensions.

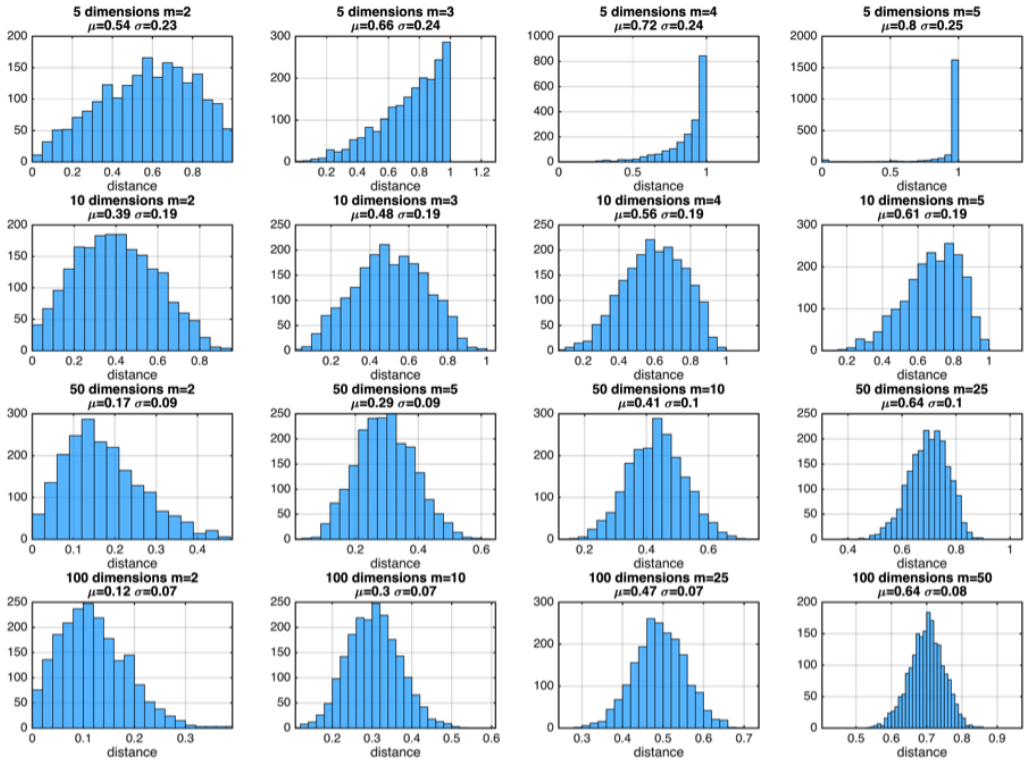

Figure 5.4: Amount of movement at different dimensions d and different sample size (m) of algorithm 2 after 10,000 replications.$Q = I$ and $||\boldsymbol{x}_k|| = 10$

In the first row of Figure 5.4, we have the histograms of probabilistic distribution on the distance $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k-1}||$ when problem dimension is $d = 5$. The first plot reports a higher frequency of movement close to 0.6, and lower frequencies at the edges (0 and 1). We observe that this is a different behavior in comparison to Algorithm 1. That is, the estimate of the gradient computed by using a linear approximation with 2 sampled points is closer to the true gradient in comparison to Algorithm 1, where a single partial derivative is computed.

As we increase the sample size m used in the linear model, the error in estimating the gradient decreases. That is, if more observations (i.e., more information) of the objective function are available, Algorithm 2 improves its estimate of the true gradient computed

by the linear model. As a consequence, the frequency of resulting distance achieved by such an algorithm shifts toward 1 as sample size m approaches the problem dimension d (columns).

On the other hand, as dimension increases, the quality of estimated gradient deteriorates within the same sample size. For example, in the first column, the sample size is fixed ($m = 2$) and dimension is increased in each row. When dimension is $d = 2$, the resulting distance $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k-1}||$ ranges from 0 to 1 with a higher probability of occurring between 0.6 and 0.7. As dimension increases, the range gets tighter and closer to 0. In other words, the algorithm losses its ability of finding a good direction of movement (or a good estimate of the gradient) as dimension increases within a fixed sample size.

We can interpret such behaviors as the cost of computing a good estimate of the gradient in a high-dimensional space. As dimension increases, more samples/observations of the underlying objective function are required in order to keep-up the quality of gradient estimate. The fundamental reasons that explain the poor performance of Algorithm 2 in high-dimensional spaces remain the same of Algorithm 1. The analysis regarding the angle between $\boldsymbol{x}$ and direction of movement are preserved.

In Algorithm 1, $\cos\theta$ is given by angle between current point $\boldsymbol{x}$ and direction of movement $\boldsymbol{w} = \boldsymbol{z}/||\boldsymbol{z}||$. On the other hand in Algorithm 2, $\cos\theta$ is given by the angle between $\boldsymbol{x}$ and $\sum_{i=1}^{m} c_i^* \boldsymbol{w}_i$. In the latter case, the intrinsic knowledge in c^* guides the direction o movement closer to the true gradient, which is the direction $\boldsymbol{x}$. However, it comes with a cost of m samples.

In Figure 5.5 we present the same experiments as in Figure 5.4 changing the matrix Q. We set the main diagonal of Q as a d-dimensional vector of i.i.d. Uniform random variables between 1 and 10. The elements outside the main diagonal are all zero. The motivation is to understand the behavior of Algorithm 2 in different response surfaces, maintaining as basis a quadratic function with unique optimal point.

The histograms in Figure 5.5 show a similar probabilistic distribution of $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k+1}||$. That is, the analysis we make in the case where $Q = I$ are also valid for a larger set of quadratic and convex functions. We argue that, in general, one can make satisfactory local approximations of more complex nonlinear function by using a quadratic function. That is, the analyzes we conduct in this Chapter regarding the behavior of optimization algorithms

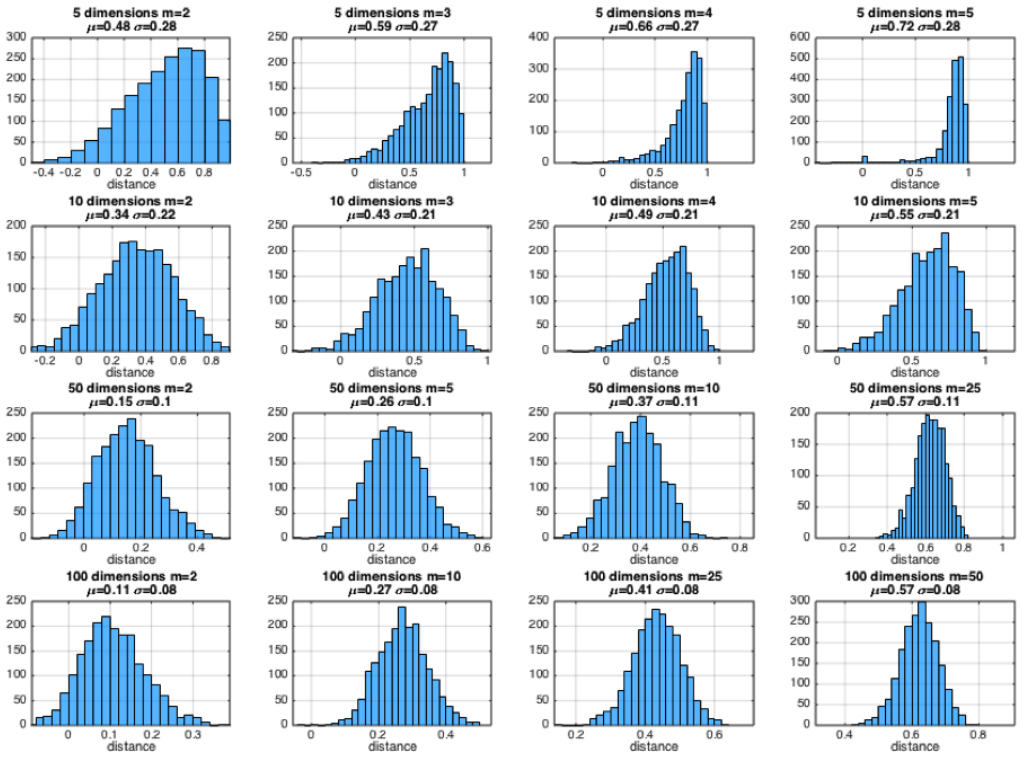

Figure 5.5: Amount of movement at different dimensions d and different sample size (m) of algorithm 2 after 10,000 replications. Matrix Q has a random main diagonal and $||\boldsymbol{x}_k|| = 10$

based on gradient in high-dimensional space are also valid for local interpretations on more complex nonlinear function than a quadratic function.

In the final experiment, we let the algorithms run for a number of iteration, as opposed to previously experiments were we considered one iteration only. That is, instead of computing the resulting distance in a single iteration (i.e., $||\boldsymbol{x}_k|| - ||\boldsymbol{x}_{k-1}||$), we set a simulation budget and let the algorithms move until the budget is consumed. There is a total of 6 experiments, combining different dimensions ($d = 100, 1000$) and different simulation budgets (50, 100 and 1000 outputs). We take 10,000 runs of Algorithm 1, Algorithm 2 with $m = 2$, and Algorithm 2 with $m = 10$. We start the algorithm by setting $||\boldsymbol{x}_0|| = 100$ and compute the final distance from optimal. That is, we are interested in observing how far from optimal point each algorithm has stopped. We set Q as in the last experiment (i.e., its main diagonal is composed by i.i.d. uniform random variables between 1 and 10).

In Figure 5.6 show the boxplot for each experiment. In the first column, we have

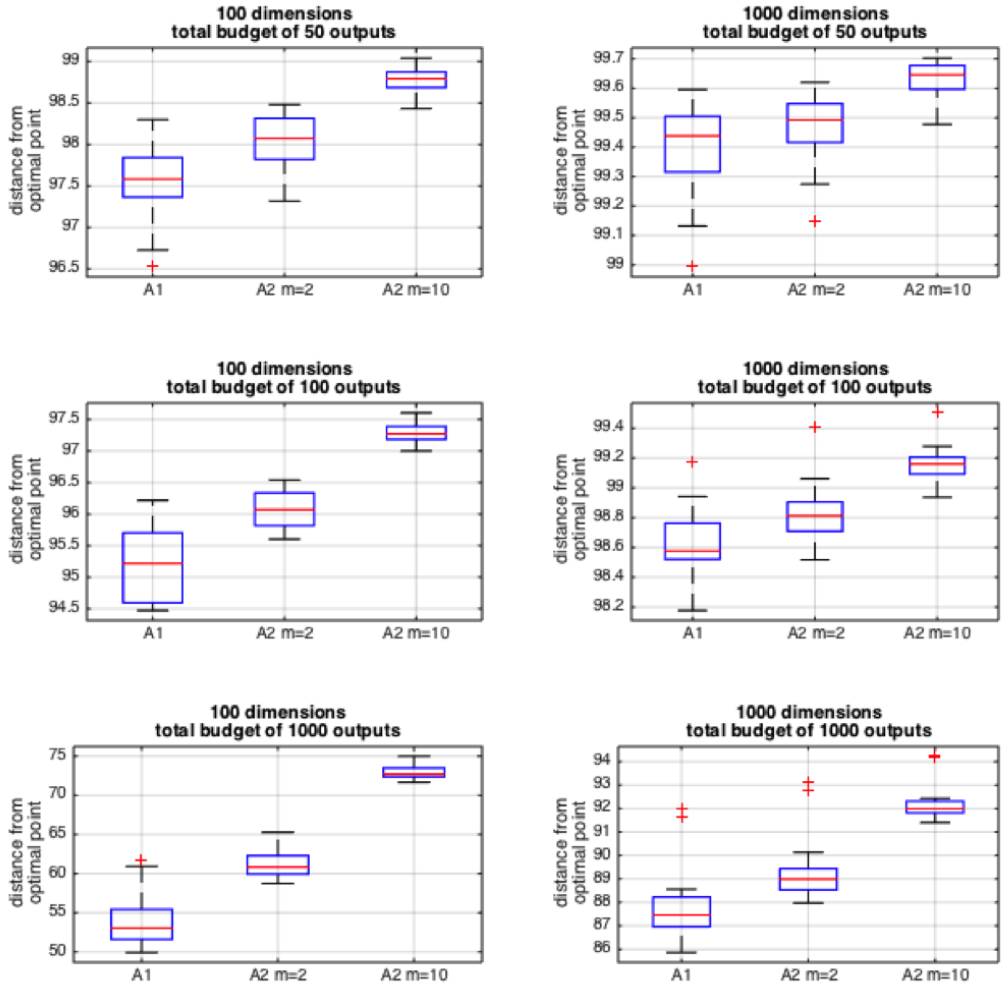

Figure 5.6: Performance of Algorithm 1 and Algorithm 2 (with $m = 2$ and $m = 10$ in high dimensional spaces)

$d = 100$, and $d = 1000$ in the second one. Simulation budget increases with the rows. We observe a similar pattern in all experiments. The Algorithm 1 goes to a larger distance, in comparison to Algorithm 2, however, it reports a larger variations on the distance from optimal point. These results suggest that while Algorithm 1 takes a relatively smaller step in each iteration, it requires fewer samples to estimate a direction of movement. Therefore, it is able to get further because it is less costly. Moreover, Algorithm 1 presents a large variance on the best solution found because the estimate of the gradient is less accurate. It indicates that the algorithm's path towards the optimum is less precise.

It is clear that the accuracy of the gradient estimate of Algorithm 2 and 10 are better than Algorithm 1. However, it comes at the expense of m outputs. We observe that the

Algorithm 2 with $m = 10$ has more information available to estimate the gradient. The resulting distance achieved by this algorithm within a single iteration is larger in comparison to the other two algorithms. However, because it consumes 10 samples per iteration, it is not able to move much before the simulation budget is consumed. Therefore, the total distance towards the optimum achieved is shorter in comparison to other algorithms.

5.3 Final Discussion

As stated early in this Chapter, our main purpose has been to gain some seminal understanding of the behavior of optimization algorithms in high dimensions. To this end we have selected the simplest possible cost functions that nonetheless capture some of the key features of the behavior of optimization algorithms in high dimension. The random sampling component of the algorithms we considered, we believe, capture a basic aspect of stochastic optimization algorithms, namely the fact that only partial information about the gradient is available at each iteration and obtaining more information corresponds to expenditure of additional computational resources.

Our efforts in this Chapter have mostly focused on analyzing the behavior of the algorithms in each iteration, namely the local behavior of the algorithms. Note that in this case, our assumption of a quadratic cost function is not overly restrictive and we expect that our observations would be valid in more general settings.

We noted that there seems to be a tradeoff between (i) obtaining more accurate estimates of the gradient which provides a longer step towards the optimum, and (ii) moving based on more noisy estimates of the gradient that requires smaller expenditure of computational resources. Such a tradeoff is generally not captured in asymptotic rates of convergence.

One possible direction for further research is to provide an analysis of the finite horizon behavior of the algorithms in high dimension. In ongoing research, we are analyzing the behavior of such elementary algorithms by using a Markov chain model. We believe it would be useful to derive the expected first passage time to the vicinity of the optimum as a global measure of the performance of the algorithm. Although we believe that the simplest cost functions capture the key features of high-dimensional space, it is important

for the completeness of theoretical analyzes to considered a more general set of surfaces.

Chapter 6

Conclusions

The survey on stochastic optimization provided groundwork to this thesis. We discoursed on the main characteristics of the most used stochastic optimizations methods, together with their benefits, limitations and especially research trends pointed out in the references herein. We positioned our thesis among the very recent developments, linking each of our main results to open gaps in the literature.

In this thesis, we make contributions to the development of more sophisticated/efficient stochastic optimization methods. More specifically, our initial goal was to design a bridge between such methods and variance reduction techniques. The research question to be answered was: What can the algorithm learn from the current available outputs to better estimate the function at other sets of decision variables? The idea was to explore as much as possible all available information that is collected from simulation output at already sampled decision variables.

The variance reduction technique of control variates is used to guide the knowledge transfer from already sampled simulation outputs at a particular decision variable to others. Because classical control variates request known control means, we resort to a variant named database control variates. The importance of latter technique is that expectation of control means no longer needs to be known exactly, and can be estimated in a setup phase. To be effective, database control variates requires that the computational effort allocated to the setup phase must be large enough so that estimation error raised in estimating control means does not affect control variates performance. We explored a lacuna

in stochastic optimization literature regarding the use of variance reduction techniques to improve efficiency of methods. In particular, the use of control variates was restricted to a few studies where the classical version is applied.

Moreover, we analyse the effects of dimension in gradient-based optimization algorithms.

6.1 Main Results and Intuition

A Metamodeling Based on Control Variates

We observed that generally in metamodeling, a set of design points is chosen and from these the function value is approximated at prediction points. Our intuition after understand more about the most used metamodeling methods (response surface methodology, radial basis function and stochastic kriging) was that we could use the outputs from design points to more efficiently and with more accuracy estimate function value at prediction points. Therefore, we started by developing a metamodel framework with database control variates as its foundations.

We demonstrated that our control variates metamodel is very flexible. It less dependent on initial parameters, and does not require a basis function as trend such as in current metamodeling tools. To guide our analysis, we used as background performance measures and template problems at Li et al. (2010). In this latter research, a systematic comparison of five of the most popular metamodeling techniques for stochastic optimization is conceived. The motivation behind this choice was to provide a fair comparison between our proposed control variates metamodel and stochastic kriging, which is the technique that has been receiving more attention in the past few years. Results show that the performance of control variates metamodel overcame stochastic kriging in all measures (global and local accuracy, robustness and efficiency) at all template problems. In addition, we analyze the performance of both methods in two practical applications of human activities to illustrate its potentials: path-dependent options, and M/M/1 queue problem. Again, control variates metamodel performed significantly better than stochastic kriging.

Next, we evaluated the use of multiple controls in our control variates metamodeling.

Because there is more than one candidate to be control (more than one design point), it is natural to consider the possibility of using more controls to better estimate the function. It is known that if there is linear dependence (correlations) among controls, adding more control does not bring new information. Thus, there are no expectations of gains in efficiency if controls are highly correlated.

However, we take this analysis further and demonstrate that using multiple controls in the particular case of the control variates metamodel can be dangerous. We use control variates connections with linear regression to conduct analyzes. We found that induced correlation raised from common random numbers can cause multicollinearity when multiple controls are used. The main drawback of multicollinearity in our case is the inflation of control variates estimator's variance, which can have a direct impact on the method's performance.

Once the control variates metamodeling framework was constructed, we noticed that there were redundant design points in the method. In other words, we realize that correlation among design points and prediction points could be maintained in a high level using fewer design points. Then, we derived a procedure to select location of design points in order to eliminate redundant ones. With such a produce, more simulation budged was allocated to the setup phase of database control variates. Results show a significant increase in the robustness performance of the algorithm, which is the ability of consistently make good predictions at different replications.

A Hybrid Fomulation of Random Search Method and Control Variates

In the next step of research, the objective was to develop a more generic procedure based on database control variates that could be applied to a larger set of stochastic optimization methods. We proposed a hybrid procedure to combine the efficiency benefits of control variates in Monte Carlo simulations to a recent developed random search method. In random search methods, a solution point may be revisited (i.e., points that are more likely to be the optimal have more chances to receive a larger simulation budget). Our intuition was to select points with more number of simulation outputs (more visited) to be use as control to better estimate the function to be optimized at points with smaller simulation outputs.

It is important to remark that simulation allocation among visited solution points is determined by the stochastic optimization method, and not by the database control variates technique. The estimated control means are the ordinary average itself of simulation outputs at most visited solution points. Therefore, there is no additional effort allocated to the setup phase of database control variates. We believe this is the main benefit of our proposed procedure. In other worlds, our framework does not require more simulation budget than the original random search method for using database control variates embedded in the Monte Carlo simulation.

Preliminary results corroborate to the argument that benefits of using a variance reduction procedure based on database control variates overcome the harm that small simulation allocation to the setup phase can cause. We analyze the performance of AHA-DCV procedure at five different stochastic optimization problems with three variants for each one of them. In majority of them, the global accuracy performance of AHA-DCV was better than stand-alone AHA or at least as good as. Only in particular cases, it was not the case. Moreover, there was observed a significant improvement in the robustness of the random search method. That is, the ability of consistently finding good solutions at different replications.

A Finite Time Analysis of High-Dimensional Search Space

The final contribution of this thesis to stochastic optimization methods is to derive a better understanding of high-dimension optimization problem. We bring light in the possible causes of poor performances of gradient-based stochastic optimization methods in finding good solutions when dimension of problems increases.

In particular, we first analyze the behavior of an elementary algorithm with a single sample based on finite-difference. We derive the probability of moving and the length of resulting movement for such an algorithm within a single iteration. We have found that the key element to why this optimization algorithm may be very inefficient in high-dimensional space lies in the cosine of the angle between the random direction of movement and the gradient direction. Furthermore, we identify that the amount of movement in a single iteration is also directly connect with such an angle. We conduct experiments to illustrate important implications of these findings.

We demonstrate that, as dimension increases, the algorithm becomes slower by taking smaller steps toward the optimum. Similarly, it becomes less efficient in finding a better solution. What happens is that the area in the surface of a d-dimensional ball for which the condition of moving is satisfied decrease as dimension increases. The experimental results corroborates with this argument.

Than we analyze an elementary algorithm based on linear approximation with multiple samples. The intuitions on the effects of dimension raised in the first algorithm remain when analyzing the second one. In the latter case, the experimental results corroborates to the argument that as dimension increases, more samples are required in order to keep-up the quality of the gradient estimate. We demonstrate that the fundamental reasons of such behavior are also connected to the cosine of the angle between direction of movement and the true gradient.

We end the Chapter by stating that there is a tradeoff between obtaining more accurate estimates of the gradient which provides a longer step towards the optimum; and moving based on more noisy estimates of the gradient that requires smaller expenditure of computational resources. We remark that such tradeoff is generally not captured in the asymptotic rates of convergence.

6.2 Possible Future Considerations

The next step of research on the control variates metamodel introduced in Chapter 3 is to evaluate the prediction ability of metamodeling methods at outside regions of the experiment design space. It is known that the performance of response surface methodology, stochastic kriging and radial basis function is expected to deteriorates rapidly at outside points. We argue that such a behavior may not be true for the control variates metamodel mainly because there is no model dependence on trend functions. Therefore, the control variates performance at outside regions that have high correlation to any design point is expected to be good. Our partial results on ongoing research corroborates with this argument.

In addition, also pretend to investigate the sensitivity of control variates metamodel to the available simulation budget. A very low simulation budget direct impacts our method

in two moments: (i) in the setup phase - estimating function value at design points - where control means are estimated. Low simulation allocation to this phase can lead to poor control mean estimation increasing the variance of function value at prediction points. Also (ii) a low simulation budget can inflate the variance of control variates coefficient in the estimation phase. It is natural to expect that low simulation budget reflects negatively on the performance of current Metamodel. An examination of the performance of control variates metamodel under low simulation budget is felt needed.

There is a wide potential directions of research concerning our hybrid approach for efficient stochastic optimization methods introduced in Chapter 4. Firstly, we plan to analyze and derive more formal results to understand the effects on convergence proofs of an embedded control variates technique on the random search algorithm. By accomplishing this task, it may bring light to the type of stochastic optimization problems in which our framework can be more beneficial.

Further, we want to adapt our hybrid approach to a larger set of stochastic optimization methods, such as stochastic approximation, metamodeling, sampled average approximation and ranking and selection. The main challenge is to decide locations and budget allocations of candidates for control in order to guarantee efficiency gains. For discrete stochastic optimization methods, repeated visits to solution points favor our approach. For continuous stochastic optimization methods, adaptation is less straightforward because there is no revisited to a same solution point. In this case, it is necessary to allocate additional simulation budget to a set of solution points that are candidates to play the role of control.

Regarding the theoretical analysis of the high-dimensional space in stochastic optimization problems, there are two immediate directions on future research. First, is to provide an analysis of the finite horizon behavior of the algorithms in high dimension. In ongoing research, we are analyzing the behavior of such elementary algorithms by using a Markov chain model. We believe it would be useful to derive the expected first passage time to the vicinity of the optimum as a global measure of the performance of algorithm in finite time. Second, it is important for the completeness of theoretical analyzes to considered a more general set of surfaces. Our expectations are that our findings up to this moment are valid for larger settings of stochastic problems.

References

Acar, E. (2015). Effect of error metrics on optimum weight factor selection for ensemble of metamodels. *Expert Systems with Applications*, 42:2703–2709.

Amaran, S., Sahinidis, N. V., Sharda, B., and Burry, S. J. (2014). Simulation optimization: a review of algorithms and applications. *Quarterly Journal of Operational Research*, 12:301–333.

Anderson, N., Evans, G., and Biles, W. (2006). Simulation optimization of logistics systems through the use of variance reduction techniques and criterion models. *Engineering Optimization*, 38(4):441–460.

Andradottir, S. (2014). A Review of Random Seach Methods. *Handbook of Simulation and Optimization*, pages 277–292.

Andradottir, S. and Kim, S. H. (2010). Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics*, 57(5):403–421.

Arora, J. (1989). *Introduction to Optimum Design*. McGraw-Hill, New York.

Aydemir-Karadag, A. and Turkbey, O. (2013). Multi-objective optimization of stochastic disassembly line balancing with station paralleling. *Computers and Industrial Engineering*, 65(3):413–425.

Aydin, N. and Murat, A. (2013). A swarm intelligence based sample average approximation algorithm for the capacitated reliable facility location problem. *International Journal of Production Economics*, 145(1):173–183.

Back, T., Fogel, D., and Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*. Oxford Univ. Press.

Bandi, C. and Bertsimas, D. (2012). Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134(1):23–70.

Banos, R., Ortega, J., Gil, C., Fernandez, A., and De Toro, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, 40(5):1696–1707.

Barrientos, A. H., Cortes, M. E., and Mota, I. F. (2014). Analysis of scientific collaboration patterns in the co-authorship network of Simulation–Optimization of supply chains. *Simulation Modelling Practice and Theory*, 46:135–148.

Barut, E. and Powell, W. B. (2014). Optimal learning for sequential sampling with non-parametric beliefs. *Journal of Global Optimization*, 58:517–543.

Bastani, M., Thanos, A. E., Celik, N., and Chen, C. H. (2014). Efficient Design Selection on Microgrid Simulations. *Proceedings of the 2014 Winter Simulation Conference*, pages 2762–2773.

Batur, D. and Kim, S. H. (2010). Finding feasible systems in the presence of constraints on multiple performance measures. *ACM Transactions on Modeling and Computer Simulation*, 20(3):13:26.

Bechhofer, R. E. (1954). A single-sample multiple decision procedure for ranking means of normal populations with known variances. *The Annals of Mathematical Statistics*, 25(1):16–39.

Benyoucef, L., Xie, X., and Tanonkou, G. (2013). Supply chain network design with unreliable suppliers: A lagrangian relaxation-based approach. *International Journal of Production Research*, 51(21):6435–6454.

Bhatnagar, S. and Prashanth (2015). Simultaneous Perturbation Newton Algorithms for Simulation Optimization. *Journal of Optimization Theory and Applications*, 164(2):621–643.

Borogovac, T. (2009). Constructive and generic control variate for monte carlo estimation.

Borogovac, T. and Vakili, P. (2008). Control variate technique: A constructive approach. *Proceedings - Winter Simulation Conference*, pages 320–327.

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society, Series B*, 13(1):1–38.

Brantley, M. W., Lee, L. H., Chen, C. H., and Xu, J. (2014). An efficient simulation budget allocation method incorporating regression for partitioned domains. *Automatica*, 50:1391–1400.

Buhmann, M. D. (2003). *Radial Basis Function*. Cambridge University Press, Cambridge, UK.

Chandwani, V., Agrawal, V., and Nagar, R. (2014). Modeling slump of ready mix concrete using genetic algorithms assisted training of artificial neural networks. *Expert Systems with Applications*, 42(2):885–893.

Chang, K. H. (2015). Improving the Efficiency and Efficacy of Stochastic Trust-Region Response-Surface Method for Simulation Optimization. *IEEE Transactions on Automatic Control*, 60(5):1235–1243.

Chang, K. H., Hong, L. J., and Wan, H. (2013). Stochastic Trust-Region Response-Surface Method (STRONG)—A New Response-Surface Framework for Simulation Optimization. *INFORMS Journal on Computing*, 25(2):230–243.

Chang, K. H., Li, M. K., and Wan, H. (2014). Combining STRONG with screening designs for large-scale simulation optimization. *IIE Transactions*, 46(4):357–373.

Chau, M. and Fu, M. C. (2014). An Overview of Stochastic Approximation. In M. C. Fu editor. *Handbook of Simulation and Optimization*, pages 149–178.

Chau, M., Fu, M. C., Qu, H., and Ryzhov, I. O. (2014). Simulation optimization: A tutorial overview and recent developments in gradient-based methods. *Proceedings of the 2014 Winter Simulation Conference*, pages 21–35.

Chau, M., Qu, H., Fu, M. C., and Ryzhov, I. O. (2013). An empirical sensitivity analysis

of the Kiefer-Wolfowitz algorithm and its variants. *Proceedings of the 2013 Winter Simulation Conference*, pages 945–956.

Chen, C. H., Chick, S. E., Lee, L. H., and Pujowidianton, N. A. (2014). Ranking and Selection: Efficient Simulation Budget Allocation. In M. C. Fu editor. *Handbook of Simulation and Optimization*, 216(1):45–80.

Chen, K. and Yu, J. (2014). Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach. *Applied Energy*, 113:690–705.

Chen, W., Kucukyazici, B., Verter, V., and Jesus Saenz, M. (2015). Supply chain design for unlocking the value of remanufacturing under uncertainty. *European Journal of Operational Research*, 247(3):804–819.

Chen, X. and Kim, K. K. (2014). Stochastic Kriging with Biased Sample Estimates. *ACM Transactions on Modeling and Computer Simulation*, 24(2):8:23.

Cheng, B., Jamshidi, A., and Powell, W. B. (2015). Optimal learning with a local parametric belief model. *Journal of Global Optimization*, 63:401–425.

Chu, W., Gao, X., and Sorooshian, S. (2011a). Handling boundary constraints for particle swarm optimization in high-dimensional search space. *Information Sciences*, 181(20):4569–4581.

Chu, W., Gao, X., and Sorooshian, S. (2011b). A new evolutionary search strategy for global optimization of high-dimensional problems. *Information Sciences*, 181(22):4909–4927.

Colosimo, B. M., Pagani, L., and Strano, M. (2015). Reduction of calibration effort in FEM-based optimization via numerical and experimental data fusion. *Structural and Multidisciplinary Optimization*, 51:463–478.

Couckuty, I., Dhaene, T., and Demeester, P. (2014). ooDACE Toolbox: A Flexible Object-Oriented Kriging Implementation. *Journal of Machine Learning Research*, 15:3138–3186.

Dasa, S., Maity, S., Qu, B.-Y., and Suganthan, P. (2011). Real-parameter evolutionary multimodal optimization-a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1(2):71–78.

Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. *Proceedings of the Thrid International Conference in Genetic Algorithms*, pages 42–50.

Delyon, B. and Juditsky, A. (1993). Accelerated stochastic approximation. *SIAM Journal on Control and Optimization*, 3(4):868–881.

Diabat, A. (2014). Hybrid algorithm for a vendor managed inventory system in a two-echelon supply chain. *European Journal of Operational Research*, 238(1):114–121.

Diaz, R., Bailey, M. P., and Kumar, S. (2016). Analyzing a lost-sale stochastic inventory model withMarkov-modulated demands: A simulation-based optimization study. *Journal of Manufacturing Systems*, 38:1–12.

Elsayed, K. and Lacor, C. (2014). Robust parameter design optimization using Kriging, RBF and RBFNN with gradient-based and evolutionary optimization techniques. *Applied Mathematics and Computation*, 236:325–344.

Evers, L., Glorie, K., Van Der Ster, S., Barros, A., and Monsuur, H. (2014). A two-stage approach to the orienteering problem with stochastic weights. *Computers and Operations Research*, 43:248–260.

Frazier, P. I. (2014). A Fully Sequential Elimination Procedure for Indifference-Zone Ranking and Selection with Tight Bounds on Probability of Correct Selection. *Operations Research*, 62(4):926–942.

Frazier, P. I., Powell, W. B., and Dayanik, S. (2008). A Knowledge-Gradient Policy for Sequential Information Collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals Statistics*, 19:1–141.

Fu, M., Bayraksan, G., Henderson, S., Nelson, B., Powell, W., Ryzhov, I., and Thengvall, B. (2015). Simulation optimization: A panel on the state of the art in research and practice. *Proceedings - Winter Simulation Conference*, 2015-January:3696–3706.

Fu, M. C. (2002). Feature Article: Optimization for simulation: Theory vs. Practice. *INFORMS Journal on Computing*, 14(3):192–215.

Fu, M. C. (2014). *Handbook of Simulation Optimization*. Springer, New York.

Gardeux, V., Chelouah, R., Siarry, P., and Glover, F. (2011). Em323: A line search based algorithm for solving high-dimensional continuous non-linear optimization problems. *Soft Computing*, 15(11):2275–2285.

George, A. P. and Powell, W. B. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, 65(1):167–198.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glynn, P. W. and Juneja, S. (2004). A large deviations perspective on ordinal optimization. *Proceedings of the 2004 Winter Simulation Conference*, pages 577–585.

Granichin, O. N. (2015). Stochastic Approximation Search Algorithms with Randomization at the Input. *Automation and Remote Control*, 76(5):762–775.

Gupta, S. S. and Miescke, K. J. (1996). Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54(2):229–244.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13:311–329.

Hannah, L. A., Powell, W. B., and Dunson, D. B. (2014). Semiconvex regression for metamodeling-based optimization. *Structural and Multidisciplinary Optimization*, 24(2):573–597.

He, D., Chick, S. E., and Chen, C. H. (2007). Opportunity cost and OCBA selection procedures in ordinal optimization for a fixed number of alternative systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(5):951–961.

Healey, C. M., Andradottir, S., and K, S. H. (2013). Efficient comparison of constrained systems using dormancy. *European Journal of Operational Research*, 224:340–352.

Healey, C. M., Andradottir, S., and Kim, S. H. (2014). Selection Procedures for Simulations with Multiple Constraints under Independent and Correlated Sampling. *ACM Transactions on Modeling and Computer Simulation*, 24(3):14:25.

Healey, C. M., Andradottir, S., and Kim, S. H. (2015). A minimal switching procedure for constrained ranking and selection under independent or common random numbers. *IIE Transactions*, 47(11):1170–1184.

Heim, B., Ronnow, T., Isakov, S., and Troyer, M. (2015). Quantum versus classical annealing of ising spin glasses. *Science*, 348(6231):215–217.

Hernandez, A. F. and Grover, M. A. (2013). Error estimation properties of Gaussian process models in stochastic simulations. *European Journal of Operational Research*, 228:131–140.

Homem-de Mello, T. and Bayraksan, G. (2014). Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85.

Homem-De-Mello, T. and Bayraksan, G. (2015). Stochastic constraints and variance reduction techniques. *International Series in Operations Research and Management Science*, 216:245–276.

Hong, L., Nelson, B., and Xu, J. (2010). Speeding up compass for high-dimensional discrete optimization via simulation. *Operations Research Letters*, 38(6):550–555.

Hong, L. J., Luo, J., and Nelson, B. L. (2015). Chance Constrained Selection of the Best. *INFORMS Journal on Computing*, 27(2):317–334.

Hsieh, L. Y., Chang, K. H., and Chein, C. F. (2014). Efficient development of cycle time response surfaces using progressive simulation metamodeling. *International Journal of Production Research*, 52(10):3097–3109.

Huajun, Z., Jin, Z., and Hui, L. (2016). A method combining genetic algorithm with simultaneous perturbation stochastic approximation for linearly constrained stochastic optimization problems. *Journal of Combinatorial Optimization*, 31(3):979–995.

Huan, X. and Marzouk, Y. (2014). Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6):479–510.

Hunter, S. R. and Pasupathy, R. (2013). Optimal Sampling Laws for Stochastically Constrained Simulation Optimization on Finite Sets. *INFORMS Journal on Computing*, 25(3):527–542.

Hussain, M., Barton, R., and Joshi, S. (2002). Metamodeling: Radial basis functions, versus polynomials. *European Journal of Operational Research*, 138(1):142–154.

Imanian, N., Shiri, M., and Moradi, P. (2014). Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems. *Engineering Applications of Artificial Intelligence*, 36:148–163.

Jeff Hong, L. and Nelson, B. (2006). Discrete optimization via simulation using compass. *Operations Research*, 54(1):115–129.

Jia, D., Zheng, G., Qu, B., and Khan, M. (2011). A hybrid particle swarm optimization algorithm for high-dimensional problems. *Computers and Industrial Engineering*, 61(4):1117–1122.

Jia, G. and Taflanidis, A. A. (2013). Kriging metamodeling for approximation of high-dimensional wave and surge responses in real-time storm/hurricane risk assessment. *Computer Methods in Applied Mechanics and Engineering*, 262:24–38.

Jin, R., Chen, W., and Simpson, T. W. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23:1–13.

Kaminski, B. (2015). Refined knowledge-gradient policy for learning probabilities. *Operations Research Letters*, 43:143–147.

Kaminski, B. and Szufel, P. (2014). Asynchronous Knowledge Gradient Policy for Ranking and Selection. *Proceedings of the 2014 Winter Simulation Conference*, pages 3785–3796.

Kazem, A., Sharifi, E., Hussain, F., Saberi, M., and Hussain, O. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing Journal*, 13(2):947–958.

Kersaudy, P., Sudret, B., Varsier, N., Picon, O., and Wiart, J. (2015). A new surrogate modeling technique combining Kriging and polynomial chaos expansions – Application to uncertainty analysis in computational dosimetry. *Journal of Computational Physics*, 286:103–117.

Kesten, H. (1958). Accelerated Stochastic Approximation. *The Annals of Mathematical Statistics*, 29(1):41–59.

Khong, S. Z., Tan, Y., Manzie, C., and Nesic, D. (2015). Extremum seeking of dynamical systems via gradient descent and stochastic approximation methods. *Automatica*, 59:44–52.

Kiefer, K. and Wolfowitz, J. (1952). Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3):462–466.

Kim, S., Pasupaty, R., and Henderson, S. G. (2014). A Guide to Sample Average Approximation. In M. C. Fu editor. *Handbook of Simulation and Optimization*, pages 207–243.

Kim, S. H. and Nelson, B. L. (2006). Selectiong the best system. In S. G. Henderson, B. L. Nelson editors. *Handbook in Operations Research and Management Science: Simulation*, 216(1):501–534.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Kleijnen, J. P. C. (2014). Response Surface Methodology. In M. C. Fu editor. *Handbook of Simulation and Optimization*, pages 81–104.

Kushner, H. J. and Yin, G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, 2nd ed. New York.

Kusiak, J., Sztangret, L., and Pietrzyk, M. (2015). Effective strategies of metamodelling of industrial metallurgical processes. *Advances in Engineering Software*, 89:90–97.

Lavenberg, S. and Welch, P. (1981). Perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science*, 27(3):322–335.

Lee, S. and Nelson, B. L. (2014). Bootstrap Ranking and Selection Revisited. *Proceedings of the 2014 Winter Simulation Conference*, pages 3857–3868.

Lee, S. and Wright, S. J. (2013). Stochastic Subgradient Estimation Training for Support Vector Machines. *Springer Proceedings in Mathematics and Statistics*, 30:67–82.

Li, H. and Demeulemeester, E. (2016). A genetic algorithm for the robust resource leveling problem. *Journal of Scheduling*, 19(1):43–60.

Li, L., Jafarpour, B., and Khaninezhad, M. R. M. (2013). A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Computational Geosciences*, 17:167–188.

Li, R. and Reveliotis, S. (2015). Performance optimization for a class of generalized stochastic Petri nets. *Dicrete Event Dynamic Systems Theory and Applications*, 25:387–417.

Li, Y., Ng, S., Xie, M., and Goh, T. (2010). A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Applied Soft Computing Journal*, 10(4):1257–1273.

Long, Y., Chew, E., and Lee, L. (2015). Sample average approximation under non-i.i.d. sampling for stochastic empty container repositioning problem. *OR Spectrum*, 37(2):389–405.

Lu, L., Xu, Y., Antoniou, C., and Akiva, M. B. (2015). An enhanced SPSA algorithm for the calibration of Dynamic Traffic Assignment models. *Transportation Research Part C*, 51:149–166.

Luo, J., Hong, L. J., Nelson, B. L., and Wu, Y. (2015). Fully Sequential Procedures for Large-Scale Ranking-and-Selection Problems in Parallel Computing Environments. *Operations Research*, 63(5):1177–1194.

Mahajan, S. and van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 43(6):334–351.

Martinez, D. L., Shih, D. T., Chen, V. C. P., and Kim, S. B. (2015). A convex version of multivariate adaptive regression splines. *Computational Statistics and Data Analysis*, 81:89–106.

Maryak, J. L. and Chin, D. C. (2008). Global random optimization by simultaneous perturbation stochastic approximation. *IEEE Transaction on Automatic Control*, 53:780–783.

Mattila, V. and Virtanen, K. (2015). Ranking and selection for multiple performance measures using incomplete preference information. *European Journal of Operational Research*, 242:568–579.

McGill, J. A., Ogunnaike, B. A., and Vlachos, D. G. (2015). A robust and efficient triangulation-based optimization algorithm forstochastic black-box systems. *Computers and Chemical Engineering*, 60:143– 153.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A. (1986). Convergence and finite-time behavior of simulated annealing. *Advances in Applied Probability*, 18:747–771.

Montgomery, D. C. and Peck, E. (2001). *Introduction to linear regression analysis*. Wiley-Interscience, New York, 3 edition.

Nedic, A. and Lee, S. (2014). On stochastic subgradient mirror-descent algorithm with weighted averaging. *SIAM Journal on Optimization*, 24(1):84–107.

Nelson, B. (1990). Control variate remedies. *Operations Research*, 38(6):974–992.

Nelson, B. and Staum, J. (2006). Control variates for screening, selection, and estimation of the best. *ACM Transactions on Modeling and Computer Simulation*, 16(1):52–75.

Ni, E. C., Henderson, S. G., and Hunter, S. R. (2014). A Comparison of Two Parallel Ranking and Selection Procedures. *Proceedings of the 2014 Winter Simulation Conference*, pages 3761–3772.

Ni, E. C., Hunter, S. R., and Henderson, S. G. (2013). Ranking and Selection in a High Performance Computing Environment. *Proceedings of the 2013 Winter Simulation Conference*, pages 833–845.

Nicolai, R. and Dekker, R. (2009). Automated response surface methodology for simulation optimization models with unknown variance. *Quality Technology and Quantitative Management*, 6(3):1–28.

Nogueira, B., Maciel, P., Tavares, E., Silva, R., and Andrade, E. (2016). Multi-objective optimization of multimedia embedded systems using genetic algorithms and stochastic simulation. *Soft Computing*, pages 1–18.

Okobiah, O., Mohanty, S., and Kougianos, E. (2014). Fast Design Optimization Through Simple Kriging Metamodeling: A Sense Amplifier Case Study. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(4):932–937.

Osmani, A. and Zhang, J. (2014). Economic and environmental optimization of a large scale sustainable dual feedstock lignocellulosic-based bioethanol supply chain in a stochastic environment. *Applied Energy*, 114:572–587.

Osorio, C. and Bierlaire, M. (2013). A Simulation-Based Optimization Framework for Urban Transportation Problems. *Operations Research*, 61(6):1333–1345.

Ozdemir, D., Yucesan, E., and Herer, Y. (2013). Multi-location transshipment problem

with capacitated production. *European Journal of Operational Research*, 226(3):425–435.

Pasupathy, R., Hunter, S. R., Pujowidianto, N. A., Lee, L. H., and Chen, C. H. (2014). Selection Procedures for Simulations with Multiple Constraints under Independent and Correlated Sampling. *ACM Transactions on Modeling and Computer Simulation*, 25(1):1:26.

Pasupathy, R., Schmeiser, B., Taaffe, M., and Wang, J. (2012). Control-variate estimation using estimated control means. *IIE Transactions (Institute of Industrial Engineers)*, 44(5):381–385.

Powell, M. J. D. (1962). An iterative method for finding stationary values of a function of several variables. *The Computer Journal*, 5(2):147–151.

Qu, H. and Fu, M. C. (2012). On Direct Gradient Enhanced Simulation Metamodels. *Proceedings of the 2012 Winter Simulation Conference*.

Quan, N., Yin, J., Ng, S. H., and Lee, L. H. (2013). Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints. *IIE Transactions*, 45(7):763–780.

Rafiq, M. Y., Bugmann, G., and Easterbrook, D. J. (2001). Neural networks design for engineering applications. *Computers and Structurtes*, 79:1541–1552.

Rajabi-Bahaabadi, M., Shariat-Mohaymany, A., Babaei, M., and Ahn, C. (2015). Multi-objective path finding in stochastic time-dependent road networks using non-dominated sorting genetic algorithm. *Expert Systems with Applications*, 42(12):5056–5064.

Rao, S. (1996). *Engineering Optimization*.

Reeves, C. R. (1997). Genetic algorithms for the operations researcher. *INFORMS Journal on Computing*, 9(3):231–250.

Regis, R. and Shoemaker, C. (2013). Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization*, 45(5):529–555.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *Computer Vision and Image Understanding*, 22:400–407.

Rosenbaum, I. and Staum, J. (2016). Database monte carlo for simulation on demand. *Proceedings - Winter Simulation Conference*, 2016-February:679–688.

Royset, J. and Szechtman, R. (2013). Optimal budget allocation for sample average approximation. *Operations Research*, 61(3):762–776.

Samadi, P., Rad, H. M., Wong, V. W. S., and Schober, R. (2014). Real-Time Pricing for Demand Response Based on Stochastic Approximation. *IEEE Transaction on Smart Grid*, 5(2):789–798.

Schmeiser, B., Taaffe, M., and Wang, J. (2001). Biased control-variate estimation. *IIE Transactions (Institute of Industrial Engineers)*, 33(3):219–228.

Schueller, G. I. and Jensen, H. A. (2008). Computational methods in optimization considering uncertainties – An overview. *Computer Methods in Applied Mechanics and Engineering*, 198:2–13.

Sen, D. and Bhattacharya, B. (2015). On the pareto optimality of variance reduction simulation techniques in structural reliability. *Structural Safety*, 53:57–74.

Shan, S. and Wang, G. (2010). Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241.

Shapiro, A. (2003). Monte Carlo sampling methods. In A. Ruszczynski and A. Shapiro, editors. *Stochastic Programming, Handbooks in Operations Research and Management Science*.

Shapiro, A. (2013). Sample average approximation. In S. I. Gass and M. C. Fu, editors. *Encyclopedia of Operations Research and Management Science*, page 1350–1355.

Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2014). Lectures on Stochastic Programming: Modeling and Theory. *MPS-SIAM Series on Optimization*, pages 207–243.

Shapiro, A., Tekaya, W., Da Costa, J., and Soares, M. (2013). Risk neutral and risk averse stochastic dual dynamic programming method. *European Journal of Operational Research*, 224(2):375–391.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transaction on Automatic Control*, 37(3):332–334.

Spall, J. C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transaction on Automatic Control*, 45:1839–1853.

Spall, J. C. (2003). *Introduction to stochastic search and optimization: Estimation,simulation, and control*. Wiley-Interscience, Hoboken, NJ.

Staum, J. (2009). Better Simulation Metamodeling: The way, what, and how of Stochastic Kringing. *Proceedings of the 2009 Winter Simulation Conference*, pages 119–133.

Steihaug, T. and Suleiman, S. (2013). Global convergence and the powell singular function. *Journal of Global Optimization*, 56(3):845–853.

Sun, G., Song, X., and Baek, S. (2014). Robust optimization of foam-filled thin-walled structure based on sequential Kriging metamodel. *Structural and Multidisciplinary Optimization*, 49:897–913.

Swersky, K., Chen, B., Marlin, B., and De Freitas, N. (2010). A tutorial on stochastic approximation algorithms for training restricted boltzmann machines and deep belief nets. *2010 Information Theory and Applications Workshop, ITA 2010 - Conference Proceedings*, pages 80–89.

Tabatabaei, M., Hakanen, J., Hartikainen, M., Miettnen, K., and Sindhaya, K. (2015). A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization*, 52:1–25.

Tsai, S. and Nelson, B. (2010). Fully sequential selection procedures with control variates. *IIE Transactions (Institute of Industrial Engineers)*, 42(1):71–82.

Tsai, S. C. and Zheng, Y. X. (2013). A simulation optimization approach for atwo-echelon inventory system with service level constraints. *European Journal of Operational Research*, 229:364–374.

Tsoukalas, I., Kossieris, P., Efstratiadis, A., and Makropoulos, C. (2016). Surrogate-enhanced evolutionary annealing simplex algorithm for effective and efficient optimization of water resources problems on a budget. *Environmental Modelling and Software*, 77:122–142.

Tympakianaki, A., Koutsopoulos, H. N., and Jenelius, E. (2015). c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation. *Transportation Research Part C*, 55:231–245.

Viana, F. A. C., Simpson, T. W., Balabanov, V., and Toropov, V. (2014). Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come? *AIAA Journal*, 52(4):670–690.

Wang, G. and Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design, Transactions of the ASME*, 129(4):370–380.

Wang, H. and Kim, S. H. (2013). Reducing the Conservativeness of Fully Sequential Indifference-Zone Procedures. *IEEE Transactions on Automatic Control*, 58(6):1613–1619.

Wang, H., Rahnamayan, S., and Wu, Z. (2013). Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems. *Journal of Parallel and Distributed Computing*, 73(1):62–73.

Wang, H., Wu, Z., and Rahnamayan, S. (2011). Enhanced opposition-based differential

evolution for solving high-dimensional continuous optimization problems. *Soft Computing*, 15(11):2127–2140.

Wang, L.-F. and Shi, L.-Y. (2013). Simulation optimization: a review on theory and applications. *Zidonghua Xuebao/Acta Automatica Sinica*, 39(11):1957–1968.

Wang, Q. (2015). Analysis of practical step size selection in stochastic approximation algorithms. *Annals of Operations Research*, 229:759–769.

Wang, Q. and Spall, J. C. (2013). Rate of Convergence Analysis of Discrete Simultaneous Perturbation Stochastic Approximation Algorithm. *American Control Conference*, 7:4771 – 4776.

Wang, Z., Mu, J., and Miao, Y. (2015). Some convergence theorems for RM algorithm. *Statistics and Probability Letters*, 99:54–60.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85.

Xiao, H. and Lee, L. H. (2014). Simulation optimization using genetic algorithms with optimal computing budget allocation. *Simulation*, 90(10):1146–1157.

Xiao, H., Lee, L. H., and Chen, C. H. (2015). Optimal Budget Allocation Rule for Simulation Optimization Using Quadratic Regression in Partitioned Domains. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 45(7):1047–1062.

Xiao, H., Lee, L. H., and Ng, K. M. (2014). Optimal Computing Budget Allocation for Complete Ranking. *IEEE Transactions on Automation Science and Engineering*, 11(2):516–524.

Xie, J. and Frazier, P. I. (2013). Upper Bounds on the Bayes-Optimal Procedure for Ranking and Selection with Independent Normal Priors. *Proceedings of the 2013 Winter Simulation Conference*, pages 877–887.

Xu, J., Nelson, B., and Hong, J. (2010a). Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 20(1).

Xu, J., Nelson, B., and Hong, L. (2013). An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS Journal on Computing*, 25(1):133–146.

Xu, J., Nelson, B. L., and Hong, J. L. (2010b). Industrial strengh compass: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 20(1):1–29.

Xu, W. and Nelson, B. (2013). Empirical stochastic branch-and-bound for optimization via simulation. *IIE Transactions (Institute of Industrial Engineers)*, 45(7):685–698.

Xu, Z. and Dai, Y.-H. (2012). New stochastic approximation algorithms with adaptive step sizes. *Optimization Letters*, 6(8):1831–1846.

Yan, X. and Reynolds, A. C. (2014). Optimization algorithms based on combining FD approximations and stochastic gradients compared with methods based only on a stochastic gradient. *SPE Journal*, 19(5):873–890.

Yang, G., Wu, S., Jin, Q., and Xu, J. (2016). A hybrid approach based on stochastic competitive hopfield neural network and efficient genetic algorithm for frequency assignment problem. *Applied Soft Computing Journal*, 39:104–116.

Yang, X.-S. (2014). *Nature-Inspired Optimization Algorithms*.

Yin, G., Wang, L., Sun, Y., Casbeer, D., Holsapple, R., and Kingston, D. (2013). Asymptotic optimality for consensus-type stochastic approximation algorithms using iterate averaging. *Journal of Control Theory and Applications*, 11(1):1–9.

Yuan, J., Ng, S. H., and Tsui, K. L. (2013). Calibration of Stochastic Computer Models Using Stochastic Approximation Methods. *IEEE Transactions on Automation Science and Engineering*, 10(1):171–186.

Zadeh, P. M., Toropov, V. V., and Wood, A. S. (2009). Metamodel-based collaborative optimization framework. *Structural and Multidisciplinary Optimization*, 38:103–115.

Zhang, M., Yang, L., Xu, C., and Du, X. (2016). An efficient code to optimize the heliostat

field and comparisons between the biomimetic spiral and staggered layout. *Renewable Energy*, 87:720–730.

Zhang, S., Zhu, P., Chen, W., and Arendt, P. (2013). Concurrent treatment of parametric uncertainty and metamodeling uncertainty in robust design. *Structural and Multidisciplinary Optimization*, 47:63–76.

Zhao, G., Borogovac, T., and Vakili, P. (2007). Efficient estimation of option price and price sensitivities via structured database monte carlo (sdmc). *Proceedings - Winter Simulation Conference*, pages 984–991.

Zheng, J., Shao, X., Gao, L., Jiang, P., and Qiu, P. (2014). A prior-knowledge input LSSVR metamodeling method with tuning based on cellular particle swarm optimization for engineering design. *Expert Systems with Applications*, 41:2111–2125.

Zhou, K., Hou, J., Zhang, X., Du, Q., Kang, X., and Jiang, S. (2013a). Optimal control of polymer flooding based on simultaneous perturbation stochastic approximation method guided by finite difference gradient. *Computers and Chemical Engineering*, 55:40–49.

Zhou, Q., Shao, X., Jiang, P., Zhou, H., and Shu, L. (2015). An adaptive global variable fidelity metamodeling strategy using a support vector regression based scaling function. *Simulation Modelling Practice and Theory*, 59:18–35.

Zhou, X. J., Ma, Y., Tu, Y., and Feng, Y. (2013b). Ensemble of Surrogates for Dual Response Surface Modeling in Robust Parameter Design. *Quality and Reliability Engineering International*, 29:173–197.