A ROBUST MULTI-OBJECTIVE STATISTICAL IMPROVEMENT APPROACH TO ELECTRIC POWER PORTFOLIO SELECTION

A Thesis
Presented to
The Academic Faculty

by

Jonathan Murphy

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2012

A ROBUST MULTI-OBJECTIVE STATISTICAL IMPROVEMENT APPROACH TO ELECTRIC POWER PORTFOLIO SELECTION

Approved by:

Prof. Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Brian German
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Christiaan Paredis
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Tommer Ender
Georgia Tech Research Institute
Georgia Institute of Technology

Dr. Scott Duncan
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 12 November 2012

ACKNOWLEDGEMENTS

I thank my advisor, Prof. Mavris, who has been benevolent and wise; my mentor, Dr. Ender, who has been encouraging and motivating; and my friend, Dr. Lee, who Bayes-ically lent me his brain on occasion.

I thank Prof. Paredis, Prof. German, and Dr. Duncan for demanding rigor and providing insight.

I thank the Shackelford Fellows program for supporting me in pure research, GTRI for supporting me in enjoyable applied research, and ASDL for being a home for the duration.

And I thank my parents, for believing in me, for making life a little bit easier, and for dancing to the appropriate gods when necessary.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Uncertainty makes everything harder, and uncertainty is everywhere.

In the electric power sector, uncertainty comes from fuel prices, from demand uncertainty, and from the weather, among other sources. Utilities must plan their generating portfolios in the face of all of this, and generally make their decisions by balancing expected cost of generating electricity against the riskiness of a portfolio. However, properly accounting for all sources of uncertainty is a computational challenge when each portfolio must be assessed using detailed time series simulations. Utilities generally under-explore their options, or under-explore uncertainty space, because an exhaustive search of both would be intractable.

In engineering design, the uncertainty challenge has been tackled a myriad of ways. Many approaches fall under the umbrella of simulation-based robust design, which separates the inputs of an engineering analysis code into *design variables*, which are under the control of the designer, and *noise variables*, which are beyond the designer's control but can be assigned uncertainty distributions. The computational cost of running simulations is mitigated to an extent through the use of *surrogate models*, which predict the simulation results with quantifiable accuracy at un-sampled points.

Within robust design, this dissertation explores two overlapping classifications of methods. Methods which rely on *crossed arrays* use one set of surrogate models for the effects of design choice on mean and risk, and at every candidate design they use a separate approach to quantify uncertainty. *Combined array* methods, on the other hand, use a single surrogate model for the effects of design and noise variables, and then estimate the effects of uncertainty from the model. In a separate classification, *design of experiments* (DoE) approaches use a fixed set of pre-specified simulation runs to build a model, whereas *statistical improvement* methods use a small set of "warm-start" runs and then adaptively sample in promising regions of the design space. When there are multiple objectives (such as mean

and risk), this document will refer to *multi-objective statistical improvement* approaches or MOSI. These classifications can be combined, yielding four possible methods that will be addressed in this document.

The literature has found that combined array (C) methods generally require fewer samples than crossed array (X) methods. It has also been shown that statistical improvement methods require fewer samples than design of experiments (DoE) approaches. Despite this, combined-array multi-objective statistical improvement (C-MOSI) methods are not found in the literature.

There are challenges to implementing C-MOSI. These include second-order probability analysis ("uncertainties of uncertainties"), re-formulating MOSI to deal with uncertain Pareto sets, and criteria for sampling in noise space. These are addressed using available literature where possible, with extensions where required. C-MOSI is successfully implemented, and shown to work, at the cost of computational overhead.

Once the challenges of implementing C-MOSI have been met, a set of experiments quantify the performance of the four methods on a scalable test problem. These seek to answer three research questions. First, are crossed array (X) methods more or less sensitive than combined array (C) methods to the number of noise variables? Second, are DoE methods more or less sensitive than MOSI methods to the number of design variables? And third, is there ever a design scenario where C-MOSI is more efficient than the other three methods, in terms of achieving some level of accuracy for as few samples as possible?

All four methods gradually reduce error with increasing numbers of samples. A power model is found to represent this progression well. Further, the differences between the methods are represented well by a linear effects model, which can be used to answer the three research questions. First, in the subset of design scenarios that were explored, combined array (C) methods are more sensitive than crossed array (X) methods to the number of noise variables. Second, DoE methods are more sensitive than MOSI methods to the number of design variables. And third, for low sample budgets, C-MOSI is found to be the most efficient of the four methods. However, when the sample budget is high, X-MOSI is able to reduce the error further.

Lastly, the methods are applied to a simulation-based energy portfolio selection problem. C-MOSI is shown to work, though the benefits relative to a combined-space DoE (C-DoE) approach are not as dramatic.

CHAPTER I

INTRODUCTION

It is likely that the decades to come will see significant changes to the world's energy infrastructure. Increasing energy demand, a desire for low-carbon, low-pollution, and sustainable forms of energy, and concerns about reliability and security will motivate a continual stream of changes in the way humanity produces and consumes energy.

An electric power utility faced with increasing demand for energy services must regularly plan infrastructure investments. These investments include increasingly diverse components, including renewable energy sources, energy storage, energy efficiency programs, and demand side management, a selection of which are shown in Figure 1. In order to properly quantify the effects of these diverse components, utilities must use time-series simulation tools. Such tools may take a long time to run, leading to computational budget constraints.

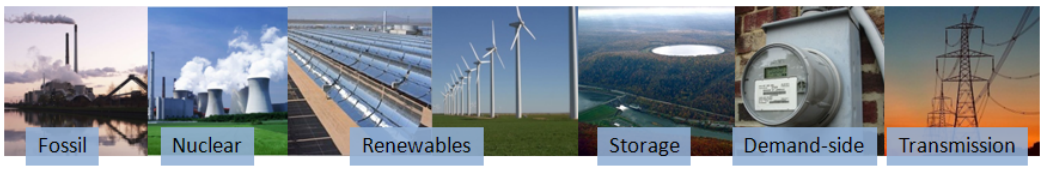


Figure 1: Diverse energy portfolio components

The portfolio selection problem is complicated by the presence of diverse sources of uncertainty, including future fuel and carbon prices, wind farm performance, demand growth, etc., represented in Figure 2. These uncertainties lead to risks, and portfolios must be chosen which balance those risks against expected benefits. The quantification of risk adds to the computational burden, requiring extra simulation runs.

In this dissertation, the electric power portfolio selection problem is defined as a *robust design problem*. The problem of interest is that of finding an *efficient frontier* of candidate energy portfolios that have high expected benefits and low risk; risks are due to uncertainty

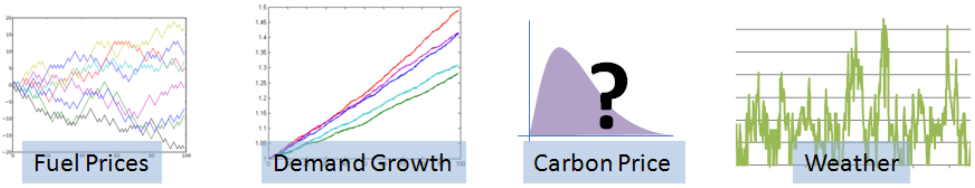


Figure 2: Multiple sources of uncertainty

in external *noise variables*. Methods from the robust design literature are examined that have the potential to solve the problem, specifically *surrogate modeling* approaches that employ either *design of experiments* or *multi-objective statistical improvement methods*. The combination of these two methods is identified as a gap in the literature that has the potential to reduce the number of simulation runs.

A central challenge is the quantification of epistemic uncertainty (due to lack of simulation code samples) on measures of aleatory (external) uncertainty. Enabling methods are found in the literature, and the end result is a set of surrogate models which are locally accurate around efficient portfolios. Computational savings are achieved by allowing high model error in regions where portfolios are dominated in terms of risks and performance. The surrogates can then be used in combination with any of a number of decision-making processes and preference structures, with a quantifiable degree of confidence.

The methods are applicable beyond energy problems, and can be used generally for design problems where simulation codes are expensive, uncertainty due to noise factors varies as a function of the control variables, and there are multiple competing objectives.

The dissertation is organized as follows.

Chapter 2 presents the motivating problem, energy portfolio selection, and describes it as a robust design problem with design variables and noise variables. The problem is simplified slightly, by only indirectly treating the challenge of stochastic time series; a full treatment is left to future work.

Chapter 3 presents background on robust design. It discusses types of uncertainty, types of robust design problems, and measures of risk.

Chapter 4 describes the mathematics behind Bayesian surrogate models, an important enabler. It describes the Gaussian Process that will be used in the remainder of this work,

Table 1: Taxonomy of Methods

	Crossed Array	Combined Array
Design of Experiments	X-DoE	C-DoE
Multi-Objective Statistical Improvement	X-MOSI	C-MOSI

along with linear Bayesian models, which are another possible model choice.

Chapter 5 describes methods found in the robust design literature for sampling expensive simulation codes when there are multiple objectives. It covers design of experiments (DoE), genetic algorithms, and multi-objective statistical improvement (MOSI) methods. It also describes the difference between combined arrays (C) and crossed arrays (X). These two classifications lead to a taxonomy of methods, shown in Table 1. The lower-right method, C-MOSI, is identified as a gap in the literature.

Chapter 6 expands on that gap. It develops an an approach to using multi-objective statistical improvement (MOSI) methods along with with surrogates regressed on a combined (C) design/noise space, referred to as C-MOSI. Several research questions are presented with regard to the effectiveness of C-MOSI relative to other related methods:

1. Are combined (C) or crossed (X) array methods more sensitive to number of noise variables?

2. Are DoE or MOSI methods more sensitive to number of design variables?

3. Is there a design scenario where C-MOSI is more efficient than other methods?

Chapter 7 develops an electric power portfolio test problem, and uses it to characterize the design and noise spaces of the problem of interest. It is found that the space is relatively smooth, but possibly multimodal, with the Pareto frontier representing a small fraction of the output space. The noise space is found to be monotonic and smooth.

In Chapter 8 an analytic test function is developed which shares the gross features of the electric power test problem, but which runs quickly and can be scaled arbitrarily in

terms of number of design and noise input variables.

In Chapter 9, the scalable test problem is used to investigate the behavior of the method. In the first experiment, the numbers of design and noise variables are fixed at 2 each, and the relative performance of four methods shown in Table 1 are characterized. As an answer to Research Question 3, C-MOSI is found to be most efficient for low numbers of samples, though due to ill-conditioning effects in the Gaussian Process surrogates used, it is unable to reduce error as much as the X-MOSI method.

In a second experiment, the sensitivities of the four methods are investigated with respect to number of design variables and number of noise variables. Four settings are used, with both number of design variables (p_D) and number of noise variables (p_S) varied from 2 to 3. These are small numbers of variables, far smaller than would be used in a real design scenario, but in order to fully characterize the performance, several hundred independent DoEs and ten independent MOSI runs are used for each method and dimensionality combination. This level of extensive testing would not be possible for larger numbers of input dimensions, because computational cost scales badly with problem dimensionality.

The methods all are found to reliably reduce error with increasing numbers of samples, and this relationship is found to be well-represented with a power function. The coefficients of the power function are taken to be quantifications of method efficiency. Lastly, this power model structure and all of the data are used to regress a single linear effects model that represents the effects of array type, sampling approach, problem dimensionality, and number of samples on error. This model is used to answer the first two research questions:

1. Combined (C) array methods are more sensitive than crossed (X) array methods to the number of noise variables.

2. DoE sampling methods are more sensitive than MOSI sampling methods to the number of design variables.

Several other interesting interactions are also presented.

Chapter 11 concludes and discusses opportunities for future work.

CHAPTER II

ELECTRIC POWER PORTFOLIO SELECTION

Clearly, electric power generating infrastructure investments have been selected by *some* method for as long as they have existed, and they have been selected with the aid of computers since the technology became available in the late 1960's. This section will begin with a very brief overview of *financial* portfolio selection, and its application to energy portfolios. It will then briefly show the development of computational energy portfolio selection methods, from simple statistical methods to time-series simulation methods, and a description of how modern utilities choose their portfolios. Three important characteristics will define the research scope, and the problem will be re-cast in the context of an engineering design problem.

2.1 Financial Portfolios

Selecting a generating portfolio can be seen as roughly analogous with the selection of a financial portfolio. The decision-maker wishes to choose investments in a portfolio of energy sources and other energy infrastructure components to meet expected increases in demand for energy services. Each component will provide energy or in some way affect energy flows, and different components will be susceptible to different sources of uncertainty. Some examples:

- An open-cycle natural gas plant will provide relatively high-cost power that can be rapidly ramped up and down, and will be vulnerable to volatility in the prices of natural gas and carbon.
- A wind farm will provide energy that incurs no fuel or carbon costs, but with a power profile that is vulnerable to wind speed uncertainty and cannot be controlled.
- A pumped-hydro energy storage facility is a net consumer of energy, but absorbs

power fluctuations and therefore improves power supply/demand matching and reduces uncertainty.

- An investment in a home weatherization program reduces demand rather than increasing supply, but participation would be uncertain.

Since the various sources of uncertainty affecting the different components are not all perfectly correlated with one another, combining multiple elements in a *diversified* portfolio helps to reduce the risk.

2.1.1 Modern Portfolio Theory

In finance, there is a single basic objective, that of *return*. However, for any portfolio, the return is uncertain, and *Modern Portfolio Theory* decomposes the problem into a two-objective problem. It treats the problem as a trade between some measure of expected performance and some measure of risk. This approach was pioneered in 1952 by Markowitz, who treated the problem as a trade between expected return and the variance of the return [75]. Given a set of assets, each with a known mean and variance, and with known correlations between all assets, Modern Portfolio Theory analytically gives a mean and variance for any arbitrary portfolio made up of those assets.

It was later shown that variance is flawed as a measure of risk [101]. Investors are more worried about the consequences of abnormally low returns than about abnormally high returns, and saying that portfolio A has higher variance than portfolio B is *not* the same as saying it is more likely to perform poorly. What's more, since returns may not be normally distributed, variance does not tell enough about the poorly-performing tail of the return distribution. This has led to a variety of alternative risk measures, including value at risk (VaR), which is simply a percentile, [58] and conditional value-at-risk (CVaR) [6]. The basic premise remains the same, however; there is a trade between some measure of *expected performance*, and some measure of *risk*. In the context of optimization, this is a Multi-Objective Problem (MOP), specifically one with two objectives. It can be solved by first identifying a Pareto frontier (also called an efficient frontier) of *efficient portfolios* as shown in Figure 3. If a portfolio is efficient, no other portfolio can be found that has

simultaneously better return and lower risk. Portfolios off of the frontier, on the other hand, are "dominated"; it is possible to find another portfolio that has lower risk and equal or better expected return. Ultimately, a decision-maker will select a portfolio along the efficient frontier based on their risk preference.

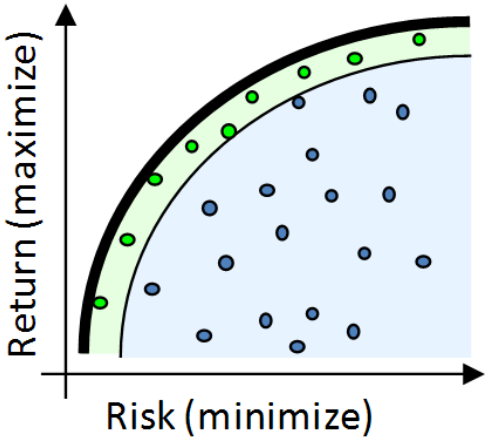


Figure 3: Efficient (Pareto) frontier. Efficient portfolios are along the green band, while dominated portfolios are in the blue region.

2.1.1.1 Portfolio Theory Applied to Energy

Energy portfolios can be selected in the same manner as financial ones. A portfolio can be found which simultaneously has low energy cost and low cost risk, and there has been substantial historical effort in this area. Modern portfolio theory as applied to energy first appears in a paper by Bar-Lev and Katz in 1976, where it is shown that real generating portfolios as selected by utilities were generally efficient, but weighted towards low cost and high risk [11]. This approach assumed that every fuel could be characterized by a mean cost per unit of energy and a variance on that cost, with known correlations between all costs. It then made use of the analytic results of Modern Portfolio Theory. Humphreys and McClain additionally considered changes in efficient portfolios over time, as well as the effects of pricing externalities on efficient portfolios (one way of dealing with a multi-objective problem, to be discussed in the next section) [52]. However, these studies did not account for the specific technical constraints of planning an energy portfolio, such as the necessity of rapidly ramping plants up and down to meet peak loads.

The technical shortcomings were to a degree addressed by Gotham *et al.* [44], who argue that Modern Portfolio Theory approaches are dismissed by practitioners because they ignore these technical factors and produce illogical results. Specifically, they identify *load factors*, that is, the fraction of time that a plant is operational. For a plant with a high fixed cost but a low operational cost, it makes sense to run all the time, at a high load factor, and to provide *baseload* power; this is how coal and nuclear plants are used. In contrast, a plant with low fixed cost but high operational cost, such as a natural gas plant, will be kept off or in standby and only used at times of peak load. To use a simple Modern Portfolio Theory approach, fixed and operational costs must be lumped, resulting in nonsensical results (such as that natural gas plants should not be used at all). Gotham *et al.* rectify this by including a Load Duration approach, a method introduced shortly in a later subsection. Even this, however, will be shown to be an incomplete solution when faced with unconventional infrastructure.

2.1.2 Decision Theory

Rather than viewing the problem as a trade between expected return and some risk metric, the *decision theory* approach instead maps all options onto a single objective, namely *expected utility*. The idea is that for very high values of return, there is less *utility* to be gained from each additional dollar. A risk-averse investor, then, will favor a lower return with high probability over a higher return with low probability; they will prefer the *safe bet*. By examining the decision-maker's risk preference, it is possible to construct a *utility function* that describes (to within a linear transformation) how much *utility* a decision-maker would derive from a given outcome. Under uncertainty, the problem becomes one of finding the option that gives the greatest expected utility.

The expected utility approach was first proposed in 1738 by Bernoulli [16] and developed into its current form by von Neumann and Morganstern in the 1940s. A description by the authors can be found in their 1944 book [112] (which principally develops game theory). It is a theory that is *prescriptive*, in that it tells decision-makers how they *should* make decisions if they are to be rational [49]. Example utility curves of risk-averse, risk-neutral,

and risk-seeking individuals are shown in Figure 4.

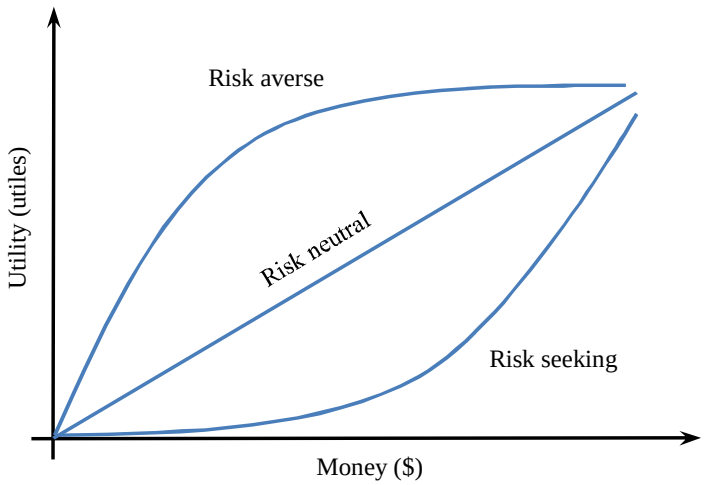


Figure 4: Utility curves showing different attitudes to risk.

Neumann and Morganstern present axioms that define the mathematical properites of *utility*, and establish that it is quantifiable up to a linear transformation. Other authors state the axioms differently. The form presented by Luce and Raiffa [73] and also adopted by Hazelrigg [49] is as follows:

- **Axiom 1.** *Ordering of Alternatives.* Preference and indifference orderings hold between any two outcomes, and they are transitive. That is, for outcomes Ω_i s.t.

$$\Omega_1 \succsim \Omega_2 \succsim \Omega_3 \succsim \cdots \succsim \Omega_r \tag{1}$$

  where $\succsim$ means "preferred or indifferent to", then

$$u_1 \geq u_2 \geq u_3 \geq \cdots \geq u_r \tag{2}$$

  where u_1 is the utility of outcome Ω_1.

- **Axiom 2.** *Reduction of compound lotteries.* Any compound lottery is indifferent to a simple lottery with the same outcomes and associated probabilities. A lottery means just what it does in a colloquial sense, that one can purchase a chance to win

a prize with some value and some defined probability. A compound lottery is one in which winning the first lottery automatically enters one into a second lottery with pre-defined probability and payout. The main point here is that only the end result matters, and there is no benefit derived from the act of gambling itself.

- **Axiom 3.** *Continuity.* Consider the same ordering of outcomes Ω_1 to Ω_r from Axiom 1. There is some probability p_i such that a certainty of outcome Ω_i is indifferent to a lottery whose outcomes are Ω_1 and Ω_r, or in notation,

$$\Omega_i \sim [p_i, \Omega_1; (1 - p_i), \Omega_r] = \hat{\Omega}_i \tag{3}$$

  That is the lottery $\hat{\Omega}_i$ is indifferent to the certainty outcome Ω_i.

- **Axiom 4.** *Substitutibility.* In any lottery L, $\hat{\Omega}_i$ is substitutible for Ω_i.

- **Axiom 5.** *Transitivity.* Preference and indifference among lotteries are transitive relations.

- **Axiom 6.** *Monitonicity.* A lottery $[p, \Omega_1; (1 - p), \Omega_r]$ is preferred or indifferent to a lottery $[p', \Omega_1; (1 - p'), \Omega_r]$ if and only if $p \geq p'$. That is, simply, that given two lotteries with the same outcomes, the one with the higher probability of the favorable outcome is preferred.

From these axioms, the following theorems follow directly (using the language of Hazelrigg [49]):

Expected Utility Theorem: Given a pair of alternatives, each with a range of possible outcomes and associated probabilities of occurrence, that is, two lotteries, the preferred choice is the alternative (the lottery) that has the highest expected utility.

The Substitution Theorem: A decision maker is indifferent between a lottery L and a certainty outcome whose utility is equal to the expected utility of the lottery, and the certainty outcome can be substituted for the lottery.

Thus it has been theoretically shown that a rational decision-maker with self-consistent risk preferences (that obey the above axioms) must have some definable utility function that

fully encodes their risk preferences [112]. The modern portfolio theory approach discussed previously, on the other hand, makes no such claim. From a utility theory standpoint, there is no guarantee that a Pareto frontier consisting of expected return and any given risk metric would contain the portfolio with highest expected utility. However, the modern portfolio theory approach has the (possible) advantage that the candidate set of portfolios can be generated prior to consultation with the decision-maker, whereas a utility theory approach requires an *a priori* elicitation of risk preference before any optimization step. A more detailed treatment of utility theory in the context of engineering design can be found in the next chapter, which focuses algorithmically on the sampling and decision-making process.

References to the use of utility theory to electric power portfolio planning can be found as far back as the 1960's and 1970's [51] [1], and applications to real energy planning problems can be found in the 1970's and 1980's [115] [62] [63]. However, actual applications are rare, and it is not currently used by electric power utilities in any of the IRP documents surveyed.

2.1.3 Multiple Objectives

An energy planner may have more than one objective. This is almost certainly true if the planner is affiliated with a publicly owned utility or planning agency, and is concerned with the public good in addition to making profit. The decision-maker must choose a portfolio that is expected to provide energy services at low cost, with low environmental impact, and with high reliability. This multi-objective nature has been noted in the literature, as early as 1980 by De Simone [27].

Multi-objective problems are well-studied in engineering, operations research, and finance, and a more thorough discussion of methods for solving them will be presented in the robust design chapter. However, it is worth pointing out two general classes of solution methods: those with and without *a priori* preference elicitation.

In the first class of methods, the preference of the decision-maker with regard to the relative importance of the various objectives can be specified *a priori*, either through simple weights or through some more complex function (as are found in Multi-Attribute Utility

Theory (MAUT), which is discussed further in the next chapter). If this is possible, then the analysis task is comparatively simpler: find a single optimum portfolio, that maximizes the single measure of goodness.

If, on the other hand, preferences cannot be elicited *a priori*, the optimization task is more challenging: find the Pareto frontier, so that the decision-maker can later select a non-dominated design according to their preferences.

In this work, it will be assumed that no prior preferences are known. Perhaps the analyst and the decision-maker are too separated by time, space, or bureaucracy to elicit preferences; perhaps there are multiple decision-makers, each with their own preferences, and compromise will only be made in the presence of real information; or perhaps it is expected that preferences will change. In any case, the assumed problem is to find a complete set of Portfolios that are Pareto-optimal ("efficient") with regard to all of the objectives. However, some of the methods developed in this thesis will be applicable to methods that do use *a priori* preference elicitation, and this will be discussed in the "future work" section at the end of the document.

2.2 Load Duration Curve Methods

For a utility or a policy analyst, there is a need to model a candidate energy portfolio both technically and economically. For a set of conventional power plants, and for planning purposes, the traditional method is to use a load duration curve method. First introduced by Baleriaux in 1967 [9] and popularized by Booth in 1972 [19], a *Load Duration Curve* (LDC) is just a re-scaled and re-oriented cumulative distribution function (CDF) of the electricity demand distribution. In load duration methods, all available plants are characterized by their capacity (or by multiple discretized possible levels of output) and by their availability, and are ranked in order of increasing energy cost ("merit order"). It is assumed that as the energy demand increases, power plants are turned on (or ramped up), cheapest ones first; and as demand decreases, they are turned off (or down). If the loads can be predicted, and there are only fossil plants being considered, LDC methods can find the lowest-cost portfolio, and can estimate operating costs and measures of reliability [71]. Though LDC

methods are based on a statistical representation of the load, they are deterministic in that they assume a fixed load distribution, assume fixed operating and fuel costs, and arrive at a deterministic lowest-cost portfolio.

2.3 Simulation Methods

Unlike conventional combustion plants, renewable energy sources are "non-dispatchable", and cannot be turned on or off at will. They thus cannot be modeled with LDC methods, which assume sources are turned on in "merit order". At low renewable penetration levels, however, there are ways of estimating their effects [78]. Energy storage presents further difficulties, though it can be treated with some low degree of fidelity [19].

Modern planners, however, need to consider more complex systems in their portfolio studies, as has been recognized as early as 1980 [27]. High penetrations of renewables place demands on the ability of power plants to rapidly throttle up and down, motivating the use of time-series studies with high time resolution. Distributed storage presents a number of modeling challenges, including multiple localized limitations on power flows. Energy storage in general cannot be accurately modeled without time-series studies because state of charge at any given time depends on all previous time periods. Demand reduction methods can be modeled at low fidelity by assuming net reductions in system-wide demand, but in practice they depend on many distributed localized demands and their interaction with the system-wide demand, and these effects may be important. New transmission infrastructures may be an important portfolio component, but they cannot be studied without some modeling of power flows.

All of these complexities individually motivate the use of time-series simulation. If they are to be considered in unified, diversified portfolios, their cumulative demands push utility planners to the use of complex and computationally intensive models as are commonly used in large-scale energy studies. Indeed, major utilities are already using time-series simulations for their portfolio planning studies [8][84][53][90][89][95]. There is even commercially available time-series simulation software, such as GenTrader [92] and Ventyx System Optimizer [111], specifically marketed to utilities for portfolio planning purposes.

Policy analysts, as well, typically use time-series simulation studies. Examples of studies include the TradeWind study in the European Union, which deals with large-scale power flows [66], and the Eastern Wind and Western Wind studies in the U.S., which model large-scale power flows, distributed storage, and high-resolution wind and solar resources [34],[42]. Many U.S. government energy policies studies use the National Energy Modeling System (NEMS), which does not use time series energy simulations, but does have a detailed national economic model that incurs significant computational cost [33].

For any given problem, a decision-maker should certainly not rely on more complex a model than is necessary, and for many portfolio planning problems it may be perfectly acceptable to rely on low-fidelity fast-running simulations or even load duration methods. However, high-fidelity methods are already employed for utility planning and policy studies, and it is natural that as the complexity of the portfolios being considered increases so too will the complexity and computational burden of the simulation methods. For the purposes of this research, then, it is assumed that simulation codes are sufficiently computationally expensive that the number of simulation runs available to the analyst is constrained by available computer time.

2.4 Treatment of Risk in Energy Studies

Much of the previous sections has dealt with the presence of uncertainty and risk in the energy portfolio planning problem. It has been shown that studies which approach the energy portfolio problem from a financial portfolio perspective tend to deal centrally with risk. However, these approaches find limited use in practice. How, then, is risk treated in practical energy portfolio selection studies?

A Lawrence Berkeley National Labs study looked at the Integrated Resource Plans (IRPs) of twelve utilities in the Western United States, and published several papers and reports [114][13][12]. They found that the treatment of risk varied substantially between utilities, but a general characterization of the more advanced plans would be that they use Monte Carlo simulations and scenario analysis. For noise variables which can be assigned distributions from historical data and forecasting, such as natural gas prices and

weather uncertainty, Monte Carlo simulations are used to characterize the distributions of the responses of interest. For noise variables which cannot be assigned historical or forecast distributions, notably carbon price, scenario analysis is more often used to characterize the effects of high, medium, and low values. The specific treatment of risk is explored further in the next chapter, and a table of risk metrics used by specific utilities can be found in Table 2.

In the case of Monte Carlo analysis, utilities use the results to construct efficient frontiers for expected energy cost and energy cost risk, and select portfolios from this frontier. If sensitivities are used as well, multiple frontiers are constructed [13].

With regard to the combination of Monte Carlo and scenario analysis, the Lawrence Berkeley studies found that the two methods were done serially, to their possible detriment. As an example, Monte Carlo studies which included the effects of natural gas price volatility might be conducted first, and used to screen out non-efficient portfolios. These down-selected portfolios would then be subjected to a carbon sensitivity study, but only after screening out natural gas heavy portfolios which might better deal with fluctuations in renewable energy [13].

In the case of policy studies, where there is a less clear "portfolio selection" objective, sensitivities tend to be used instead, with only a handful of simulation cases being run at all [34][42].

Therefore, in the universe of electric power portfolio studies, there seems to be a chronic under-exploration of uncertainty space.

2.5 Exploration of the Portfolio Space

In order to select a portfolio from the efficient frontier, that frontier must first be found. From any set of portfolios with known expected cost and known cost risk, a "Pareto set" can be found, of portfolios which are not dominated by other known portfolios. However, there may be other unexplored portfolios, not included in the data set, that dominate the known Pareto set. As more and more portfolios are examined, the Pareto set found from the data will more and more closely resemble the "true" Pareto frontier of all possible portfolios.

Figure 5 shows two efficient frontiers found in the Integrated Resource Plans of PacifiCorp. It uses an estimate of Conditional Value-at-Risk as a metric (though it is not labeled as such), and shows 16 portfolios, for two partifular carbon tax scenarios. The frontier itself is quite small relative to the overall range of values. The same document contains similar plots that are made under different carbon tax scenarios. Note that the risk in the plot is due to (quantified) fuel price uncertainty, whereas uncertainty about future carbon prices is treated in a fundamentally different manner.

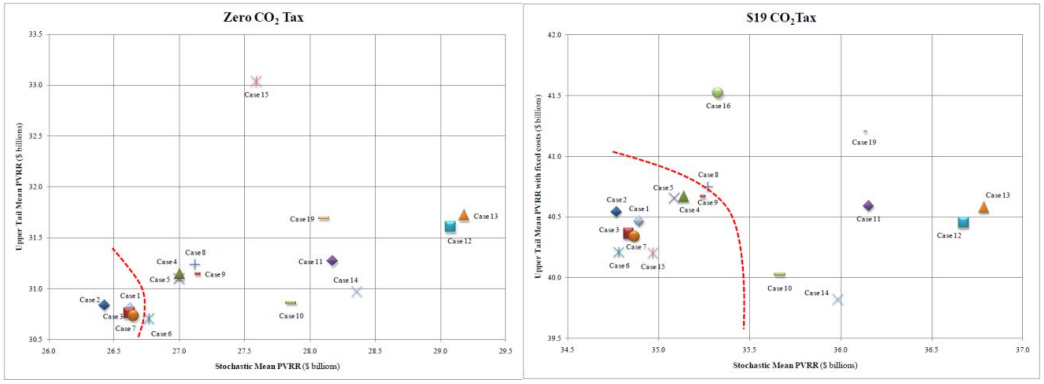


Figure 5: PacifiCorp's frontier plots for two carbon price scenarios. The IRP document contains additional plots for other carbon scenarios [90]

Unfortunately, energy utilities tend not to examine very many portfolios. Of all the resource plans examined by the Lawrence Berkeley group, nearly all examined fewer than 100 portfolios, and some examined as few as 20 [13]. The portfolios were generally hand-picked according to the expertise of the decision-makers. One utility, PacifiCorp, used optimization to select their portfolios, and found one optimized portfolio for each of about 50 noise variable scenarios. They explicitly noted that they limited the number of scenarios due to the data-processing and model run-time requirements [90].

Computational budget, then, is a very real constraint, and limits the extent to which utilities are exploring the portfolio and noise spaces. If portfolio and noise space can be sampled more carefully, it seems likely that there will be an opportunity to better approximate the "true" efficient frontiers, and ultimately to find better portfolios.

2.6 Generalized Description of the Energy Portfolio Selection Problem

The characteristics of energy portfolio planning problems define the problem to be solved, but methods useful for this problem will be applicable to many similar problems. The problem can be treated generically as belonging to a class of decision-making problems characterized by the following:

(a) Select a value of D, where D is a vector of decision variables, in this case representing the amount of investment in each portfolio option. A particular setting of D can be referred to as a *portfolio* or, in engineering literature, as a *design*.

(b) There are multiple measurable responses which should me minimized (or maximized) as objectives.

(c) The response values at each D can only be calculated with a simulation code that is computationally expensive. It is assumed for the sake of generality that the responses may have local minima or be otherwise deceptive, though it is assumed that they are at least locally smooth.

(d) A vector S represents *noise variables*, which are additional deterministic inputs to a simulation code but represent uncontrollable environmental factors in the real world. It is assumed that their distributions can be known or estimated from data.

(e) Some simulation inputs take the form of *stochastic time series*, and fluctuate randomly from time step to time step.

(f) For any D, there is a probability distribution associated with each response due to uncertainty. The probability distributions of the responses may be correlated with each other.

From the generalized problem definition above, three central characteristics of the problem are identified, each motivating a general research objective.

2.6.1 Characteristic 1: Presence of Noise Variables

As is seen in items (d), and (f), it is assumed that the decision-maker is concerned with choosing a portfolio that is robust to uncertainty in the *noise variables*, represented by a vector S. Noise variables might include growth in energy demand, average fuel prices, carbon prices, or average wind speed (due to uncertainties in wind farm siting or average weather), among other factors. For the purposes of this research, it is assumed that these variables obey a known probability distribution, $p(S)$, which in practice can estimated from data, forecasting, or expert opinion. The responses of the simulation tool will be sensitive to these noise variables, leading to the first objective:

Objective 1: Characterize the uncertainty of all responses of interest due to uncertainty in the noise variables.

This characterization of response uncertainty will be necessary for the next objective.

2.6.2 Assumption: Exclusion of Stochastic Time Series

Item (e) deals with uncertainty due to *stochastic time series*. Stochastic time series might include hourly or even minute-to-minute wind speeds, hourly cloudiness, hourly temperature, or daily natural gas prices. For series such as wind speed, an assumption of a constant value would provide a very wrong result, since random fluctuations are an essential characteristic. Instead, a noisy wind speed time series must be generated that has all of the appropriate statistical properties. Simulations run with different time series might produce different results, even if the statistical properties of the input time series are identical. This means that electric power simulations are inherently stochastic, and multiple runs are required to fully characterize a single design, even when all of the *noise variables* (such as average wind speed, as discussed in the previous subsection) are held constant.

Not only are electric power simulations inherently stochastic, but the output distributions will vary as a function of the design. All else held equal, a portfolio with low wind penetration might have lower variability than one with high wind penetration, for example. Problems of this type are called *heteroscedastic*.

In practice, it is possible to evaluate a portfolio by using a single fixed time series that has the appropriate statistical properties, and to hope that the simulation time period is long enough to capture the mean and variability trends. This is not ideal, but in order to reduce the scope of the problem, it is the approach used in this research. Full treatment of uncertainty due both to heteroscedasticity *and* noise variables is left to future work.

2.6.3 Characteristic 2: Multiple Objectives

As is seen in item (b), there may be multiple measures of "performance", each of which is an objective.

> **Objective 2:** Find portfolios that are efficient in satisfying multiple objectives and are selected with proper consideration of risks.

Under a portfolio theory framework, this means finding Pareto frontiers of non-dominated portfolios in a combined expected-performance/risk objective space. Under a utility theory approach, this means constructing appropriate utility functions through decision-maker preference elicitation with regard to different objectives and risk.

When utilities plan their generating portfolios, they usually consider only a single measure of performance, namely cost of electricity [8][84][53][90][89][95]. Only rarely do they consider other objectives such as job growth [2]. If cost is the only stochastic objective, then expected cost and cost risk are the two deterministic objectives under a portfolio theory approach. Methods applicable to a 2-objective problem can be generalized to problems with higher numbers of objectives, so all testing will be with respect to a 2-objective mean/risk problem.

Under a utility theory approach, the classic single-objective formuation can be used without involving the more complex methods of Multi-Attribute Utility Theory. However, again the methods should be extensible to the multi-objective case if necessary.

2.6.4 Characteristic 3: Expensive Simulations

As has been stated in item (c), it is assumed that any given energy portfolio (design) must be evaluated with a time-series simulation computer code. Further, it is assumed that these evaluations are computationally expensive. Though in simplified scenarios (and indeed, in the the test cases for this research) a less expensive simulation may be used, it is assumed that if a utility is seriously planning changes to its portfolio, it will want a level of detail and accuracy that will motivate the use of a more expensive code. Model features may include fine spacial resolution to resolve wind performance, short time steps to resolve transients and reliability, and detailed modeling of distributed generation and storage. It is assumed that there will always be a trade between fidelity and computational speed, and that a desire for better information will always motivate the use of a higher fidelity (and thus more expensive) code.

This characteristic motivates the last general objective:

> **Objective 3:** Meet all other objectives with fewer simulations than the state of the art.

Even if a method developed through this research effort can be demonstrated to use fewer simulations in a particular test case, it will be impossible to demonstrate it for all possible scenarios. Nonetheless, reducing the number of simulations is an important objective, and any indication that a method does so in even a single limited test case would be an encouraging result.

2.6.5 Energy Portfolio Selection Problem in the Context of Engineering Design Literature

The three general characteristics from the previous sections can be used to place the energy portfolio selection problem in the context of the engineering design literature, rather than the finance or electric power literature.

- Objective 1, to characterize the sensitivity of portfolios to noise, is the (non-exclusive) domain of *robust design*. Robust design will be discussed generally in Chapter 3.

- Objective 2, involves a combination of robust design with the field of *multi-objective* design. Relevant literature will be discussed in Chapter 5.

- Objective 3, to reduce function calls, will motivate the use of *optimization* and *surrogate modeling* methods. Surrogate modeling will be treated in Chapter 4, and optimization will be discussed in Chapter 5.

The focus of the literature surveys, then, will be on *robust design*, and specifically its intersection with *multi-objective*, *optimization*, and *surrogate modeling* methods. As a shorthand, this class of problems will be referred to as Robust/Multi-Objective/Expensive problems. In Chapter 6, a method will be developed which incorporates all of the above elements.

CHAPTER III

ROBUST DESIGN BACKGROUND

In the previous chapter, the problem was classified generally as a Robust/Multi-Objective/Expensive problem. Many engineering problems take a similar form. In aircraft design, for example, the designer seeks an aircraft that has low fuel burn and low pollutant emissions in the presence of uncertain environmental conditions. Unfortunately for the designer, analysis methods such as computational fluid dynamics codes are computationally very expensive. There is much existing work on this type of problem to be found in the engineering design literature.

This chapter will discuss the basics of robust design. It will cover classification of uncertainty, types of robust design, and definitions of performance and risk. The next two chapters will also be devoted to aspects of robust design. Chapter 4 will cover surrogate models, an important enabler for most robust design methods. Chapter 5 will discuss sampling methods for deciding what analysis cases to run.

3.1 Classification of Uncertainty

It is worth taking some time to classify uncertainty, as this will be important in later discussions of surrogate modeling, robust design, and adaptive sampling.

3.1.1 Aleatory Uncertainty

The preceding chapters spent considerable time discussing *environmental uncertainty* due to sources such as fuel prices, weather, etc., and how these sources lead to risk. Environmental or "true" or "natural" uncertainty is also known as *aleatory* uncertainty, or alternately as "irreducible uncertainty". If the observer has perfect knowledge of nature, they will still observe aleatory uncertainty. In this research, aleatory uncertainty is represented through the use of *noise variables*, that is, quantities like carbon price that while unknown in nature can be specified exactly in a model. In practice, especially in energy simulations, there is

additional aleatory uncertainty that is due to the use of random time-series data, and may result in different simulation results for the same set of inputs when results are aggregated over months or years. While important, treatment of this type of aleatory uncertainty is left to future work.

It could be argued that much environmental uncertainty could be reduced with better models, and is not truly aleatory after all; for example, advances in weather modeling have reduced the uncertainty about whether it will rain tomorrow. However, for the purposes of this work, any uncertainty which is external to the designer's model will be considered aleatory.

In this document, the only source of aleatory uncertainty will be distributions on noise variables. These distributions will be assumed to be known from data or estimated by the decision-maker.

3.1.2 Epistemic Uncertainty

When dealing with experiments, there is a further type of uncertainty due to measurement errors or incomplete observation. This is called "reducible" or *epistemic* uncertainty. In the context of this research, the focus will be on a narrow sub-set of epistemic uncertainty, specifically uncertainty that is due to *not having sampled* at a particular setting of variables. The experimenter *could* sample, and reduce epistemic uncertainty, but because simulations are expensive they may choose not to. This choice is at the root of statistical improvement optimization methods, which will be discussed later, and the decision relies on a quantification of this type of epistemic uncertainty.

There are other types of epistemic uncertainty which, while important, are left outside the scope of this research. For example, it is assumed here that the computer experiments are free of epistemic noise; in practice, it may be the case that small changes in input variables lead to "noisy" changes in the outputs. There will also be epistemic errors due to the use of an approximate simulation model; however, this type of error is not directly relevant to this work. In this document, epistemic uncertainty is assumed to be due only to lack of samples.

3.2 Robust Design Classification

In engineering design, there is uncertainty associated with both the manufacturing implementation of a design and with the environmental conditions that will be seen by the final product. Note that in the classification above, these are *aleatory uncertainties.*

If a design is chosen without regard to these uncertainties, and is "optimized" to maximize some measure of performance, the final manufactured product may perform poorly due to imprecision in the manufacturing process or off-design operating conditions. The object of robust design is to choose a design such that, even in the face of these uncertainties, the final product will perform well with high probability.

The origins of robust design can be traced back to Taguchi methods [107], but the field has changed significantly since that time, and a detailed discussion of its evolution is not necessary. It will suffice to define robust design as it appears in the current literature, to give context for the approaches found in a later chapter.

Chen *et al.* use the following classification of robust design problems [21]:

Type I - minimizing variations in performance due to variations in noise factors (uncontrollable parameters)

Type II - minimizing variations in performance caused by variations in control factors (design variables)

A depiction of the two types of robust design problems, modeled after a figure found in Chen *et al.*, is found in Figure 6.

In this classification system, the focus of this research is on *Type I* robust design. There are environmental factors beyond the control of the decision maker, such as fuel prices, weather, and demand. The decision-maker wishes to choose a portfolio that performs well with high confidence in the face of these uncertainties.

It is also possible that the decision-maker might wish to choose a portfolio which exhibits Type II robustness, where it is insensitive to changes in the design variables. A wind farm might not be built to the same capacity as expected, or a demand reduction program might see fewer participants than intended. However, the focus of utility portfolio selection

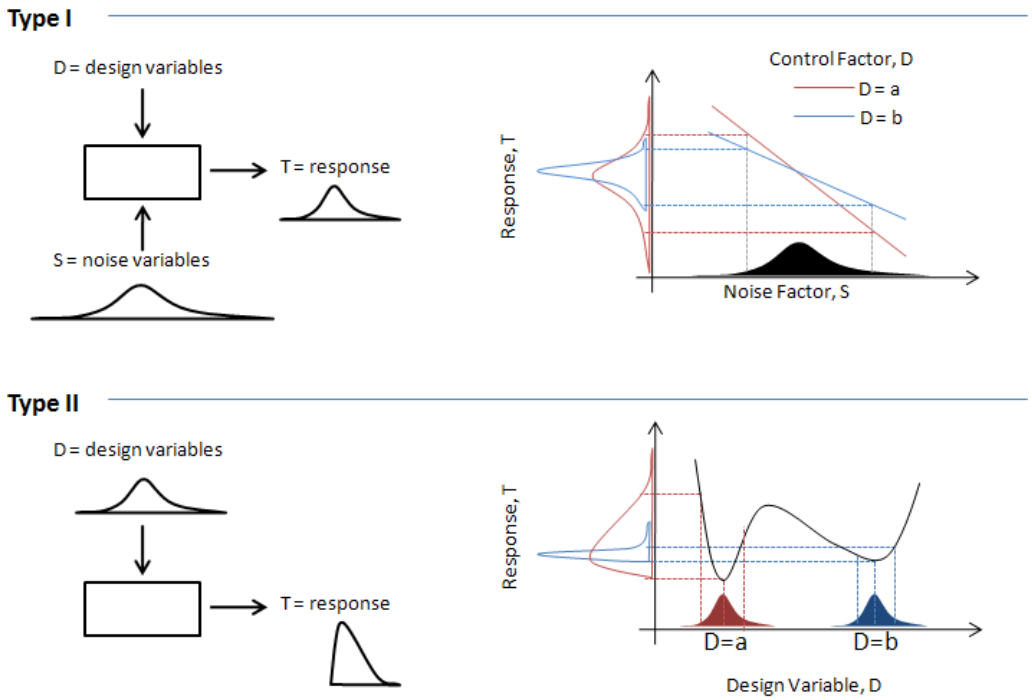


Figure 6: Robust Design Problem, robustness of the response T for two designs, $D = a$ and $D = b$: (a) Type I - robust to uncertainty in noise variables, (b) Type II - robust to uncertainty in design variables. (Figure after Chen *et al.*[21])

is generally on Type I robustness, and in any case Type II is easier to deal with. Any approach to solving Type I robust design problems can be easily altered to handle Type II problems, but the reverse is not true. As can be seen from Figure 6, Type I requires an extra set of variables, the noise variables, and methods for identifying their effects. Type II, however, can be handled by simply "wiggling" already existing design variables. The focus of this work, then, will be exclusively on Type I robust design.

3.3 Performance and Risk

It has been established that the objective of this work is to find portfolios that perform well, yet are insensitive to uncertainty in noise variables (they are "robust"). But how are performance and robustness measured?

Fundamentally, robust design deals with performance measures that are stochastic. Given assumed probability distributions on the noise variables, there will be probability distributions on the performance metrics. Ultimately, it will be necessary to compare one

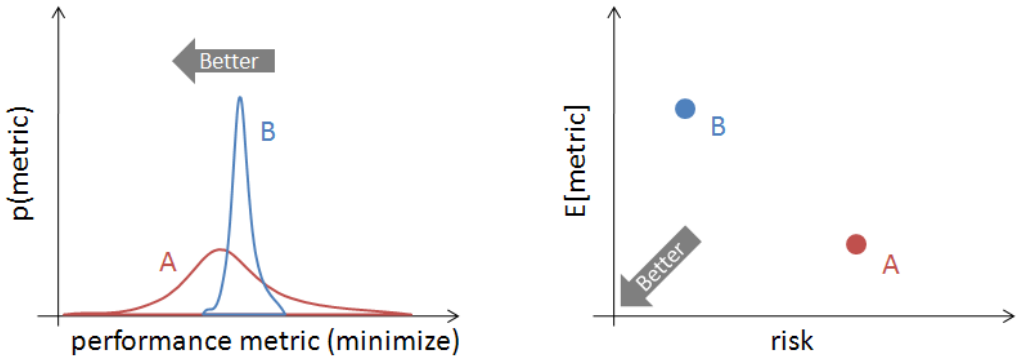


Figure 7: Decomposing a stochastic optimization objective into an equivalent deterministic problem (a) Two portfolios in probability space, with probability density p(metric) as a function of the performance measure (which is to be minimized) (b) The same two portfolios in a decomposed two-objective space, plotted on axes of expected performance E[metric] and risk

design to another. To do so, the designer must have a way of transforming the stochastic design problem into an *equivalent deterministic problem* [106].

As has already been discussed, financial portfolio theory creates an equivalent deterministic problem by breaking the stochastic objective into two deterministic objectives, a mean and a measure of risk. In Markowitz, the trade is between mean and variance [75]. Alternatives to variance include value at risk (VaR) [58] and conditional value-at-risk (CVaR) [6]. In Figure 7, a pair of portfolios are conceptually depicted in a stochastic performance space (left), and decomposed into a mean/risk space (right).

In the most general case, there may be multiple stochastic objectives. This adds complexity, because it is possible that there are correlations between the various risk measures. In the most complex case, there is a risk objective and an expected performance objective associated with each stochastic objective, and a correlation between every possible combination of risk objectives. This correlation may itself change from portfolio to portfolio.

It is difficult to express the entire multi-objective/multi-risk concept in a single graphic, but a conceptual depiction of a problem with two simple objectives (and an expected performance and a risk objective for each) is depicted in Figure 8. The left side (a) shows conceptual scatterplots of both benefit and both risk objectives, with each point corresponding to a portfolio. Each sub-graph shows a 2-D projection of the 4-dimensional Pareto frontier.

In this notional scenario, there is a trade-off between every pair of objectives. On the right side (b), a single notional portfolio is selected, and a joint probability distribution of the two basic objectives shows that their distributions are positively correlated; and since risk is a function of the distribution, that means their *risks* are correlated. A decision-maker might want to avoid a positive correlation in risks, since it means that poor performance in one objective will tend to occur simultaneously with poor performance in another objective. In the extreme, each of these risk correlations *could* be treated as an objective of its own, but this complicates the problem further.

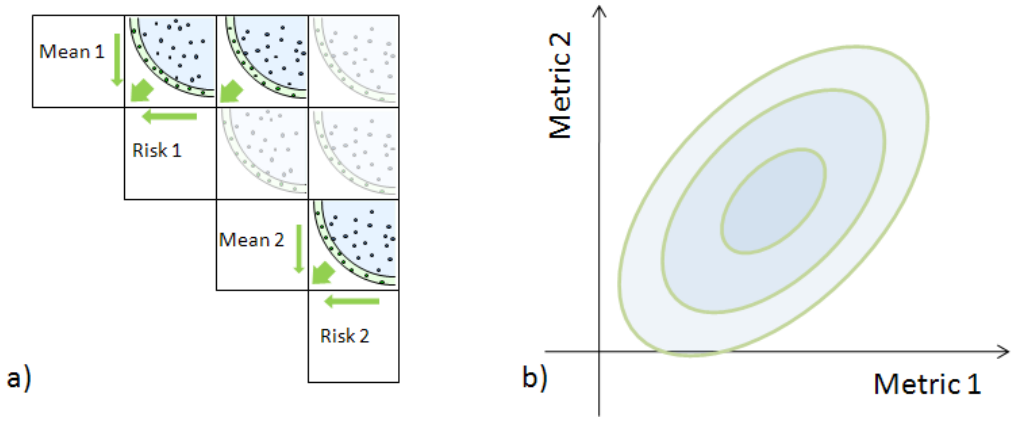


Figure 8: a) Notional multi-objective scatterplot with two expected performance and two risk measures b) Notional joint probability distribution for two objectives for a single portfolio, illustrating a portfolio with two correlated stochastic objectives

This multi-expected-performance/multi-risk decomposition of the stochastic multi-objective problem is used by Chen [21], and is referenced by Daskilewicz *et al.* [26].

There are other ways of turning a stochastic multi-objective problem into an equivalent deterministic one. Taguchi was mostly concerned with matching a target rather than minimization or maximization, and used a signal to noise ratio or a loss function [107], and this approach was adopted by others [113]. If the objective is minimization or maximization but specific targets can be chosen for each objective, Joint Probability Decision Making characterizes each portfolio by a single joint probability of simultaneously meeting all targets [10]. This captures the correlations between all risk measures, but does not allow for variable targets, and thus requires *a priori* input from the decision-maker. Another approach, used by Patel [91], is to create "layers" of Pareto frontiers, each at a particular pre-specified

confidence level.

3.3.1 Decision Theory in Engineering Design

In the previous chapter, the basics of decision theory were presented as a means of choosing options under uncertainty. Though developed as an economic theory, decision theory can be applied to simulation-based engineering design. In the 1960's, Ronald Howard developed *decision analysis* as a methodology for making model-based decisions, including those in engineering [51]. At its core is the use of utility theory, combined with appropriate engineering and economic models, in an iterative process that includes decision-maker preference elicitation and further information gathering as needed. Howard's decision analysis cycle is shown conceptually in Figure 9.

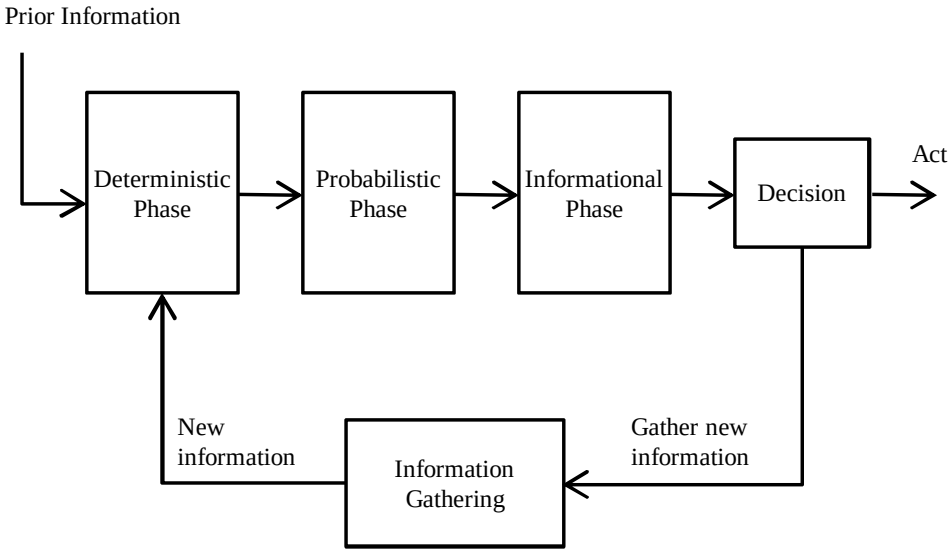


Figure 9: Howard's decision analysis framework [51]

Howard's framework includes the same concept of noise variables (which he calls state variables) and design variables (which he calls decision variables) as found in the robust design literature. Howard's state variables are explicitly defined according to the *subjective* or *Bayesian* view of probability; that is, they represent the beliefs of the decision-maker, whether this is informed through extensive specific data or through other means, and are updated according to Bayes' theorem as new information becomes available.

Briefly, his framework has three main phases. Since the framework was developed in the 1960's, when computing resources were precious, it is very conservative with regard to the use of simulation models, though it is a model-based framework. In the deterministic phase, the sensitivity of the model to its inputs is assessed in a screening step, and unimportant factors are then left at default settings. In the probabilistic phase, the effects of uncertainty in the state (noise) variables are considered; it is here that the decision-maker's risk preferences are elicited. Howard goes beyond pure utility theory, and monetizes all outcomes, by asking the decision maker what *certain equivalent* amount of money they would accept as a substitute for an uncertain lottery.

Finally, the information phase determines the value to the decision-maker of gathering information. If the state of knowledge about the system in question can be improved through data gathering or experimentation, the value of that information is quantified using utility theory and certain equivalents, and is compared with the cost of obtaining it. The process is iterative: further information may be gathered, which will update the beliefs of the decision maker, and the process is repeated until it is not worth gathering information, at which point the option with the highest expected utility is chosen. Howard's framework assumes that the decision-maker is an active part of the process, and their preferences with regard to the utility of outcomes, risk, and time are all elicited within the framework.

Howard proposes that this framework can be useful in a wide range of problems. Among the most complex, he explicitly mentions the problem of power system planning [51].

3.3.1.1 Hazelrigg's Design Framework

An updated approach to the use of decision theory in engineering product design can be found in Hazelrigg [49]. Hazelrigg's objective is a fully rational and rigorous theory of engineering product design, and he develops his own framework, central to which is again the use of utility theory.

Hazelrigg's framework can be seen in Figure 10. It is described as a nested optimization cycle. The creativity of the designer is explicitly invoked in the creation of a product configuration. From this point, appropriate modeling is used to find the performance of the

design as a function of a set of design (decision) variables. As in Howard's framework, noise variables represent the subjective beliefs of the designer. The preferences of the desicion-maker are required with respect to outcomes, including preferences for risk (using a utility theory approach) and time (using a discount rate approach). The performance of all possible designs must be traced to their effects on those outcomes: if the desired outcome is to make profit on a product, there must be modeling to estimate how the phyical performance and pricing of a product will affect its sales and, ultimately, the profits derived from those sales. The design process can then be formulated as an optimization problem, where decision variables are adjusted to find the product design that maximizes the expected utility of the decision-maker. This optimization loop is nested in a larger optimization loop, where the product concept itself can be changed. Because the framework uses utility theory, and every possible outcome can be mapped to an expected utility, all product concepts can be compared to each other using a single metric.

In practice, Hazelrigg notes that there may be technical challenges. The optimization space is huge, and there are many layers of modeling required. Additionally, there are assumptions that may be difficult to meet in practice. Importantly, in order for the method to be rigorous and rational, the concept of a "decision" must be strictly defined as having the following properties:

1. A decision is a mental commitment to action, a commitment of resources.
2. A decision is made in the present, and is irrevocable. An part which is revocable is not part of the decision.
3. A decsion is made by *a single individual*.
4. A decision is a choice from a set of alternatives
5. A decision is made to affect a desired outcome
6. All real decisions involve an element of risk
7. Decisions demand an expression of preferences

For the most part, these characteristics correspond to the general colloquial usage of the word "decision". However, some of these aspects may differ somewhat from the general usage of the word, and place specific restrictions on what kind of problems can be solved with Hazelrigg's design framework. Property 3, that a decision is made by a single individual, is necessary in order that the decision be guaranteed to be rational. Hazelrigg invokes Arrow's impossibility theorem [5] to argue that any decision made by a group is susceptible to irrationality. However, Hazelrigg acknowledges that in practice, engineering design is performed by large groups of people, and methods of dealing with this must be found.

The last property, that decisions demand an expression of preferences, is not in itself inherently restrictive, and it must ultimately be true. But Hazelrigg's framework embeds this preference expression early in the process, before the optimization step. In practice, when the process extends over multiple people and layers of organizations, it could be necessary to begin simulation before preference can be elicited.

Ultimately, Hazelrigg asserts that when it comes to a mathematically rigorous theory of how design *should* be done, there is little room to change the process; but in order to have a process that works in practice, it may be necessary to relax the strict rationality requirement, and to develop practical approaches that attempt to minimize the negative impacts of any irrationality which is introduced.

3.3.1.2 Multi-Attribute Utility Theory

Classically, utility theory deals with the utility of a single monetary quantity. Usually, especially in the cases of for-profit companies, it is possible to specify some single quantity such as profit as a single objective. Utility theory can just as easily deal with a single non-monetary quantity. However, in cases where there truly are multiple objectives, some method is needed for combining multiple metrics into a single measure of utility.

Utility theory for multiple objectives was developed starting in the 1960's, with early work by Pruzan and Jackson [94] and Ting [109]. Keeney and Raiffa elaborate in their 1976 text [64]. The method involves using decision-maker preference elicitation to first

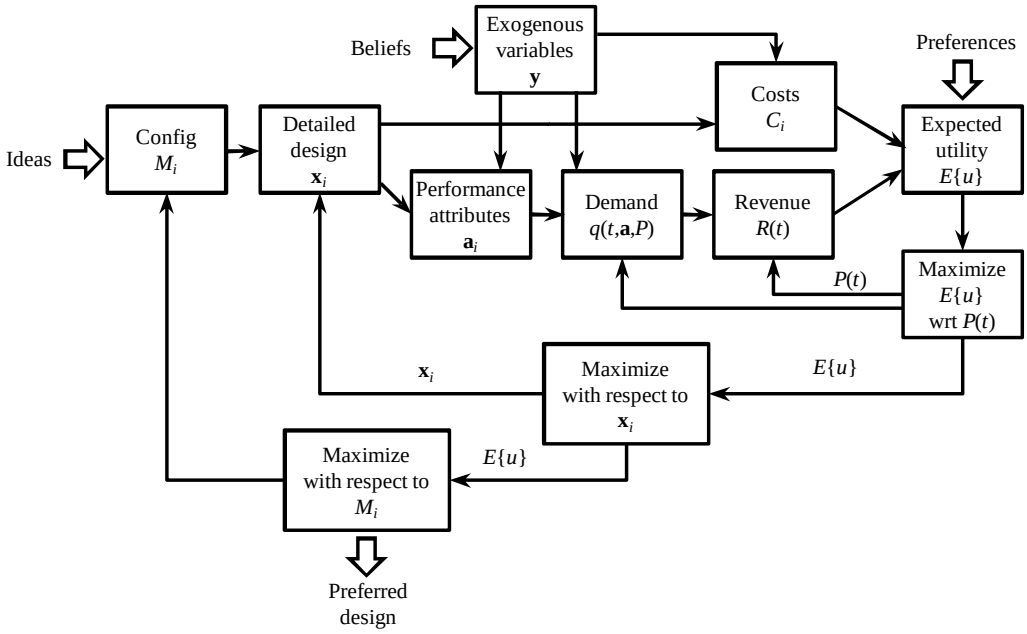


Figure 10: Hazelrigg's engineering design optimization framework [49].

build utility functions for each of the objectives. Then, if the independence of the utilities of the objectives can be established, the utility functions are combined into a single utility function. As with single-objective utility theory methods, the method prescribes that the decision maker should choose the option with the highest expected utility. Once again, preference elicitation is an integral part of the method, and in a simulation-based optimization approach, decision-maker preferences must be elicited prior to optimization. Multi-Attribute Utility Theory has been used in electric power planning [1] [115] [62] [63].

3.3.2 Choice of Equivalent Deterministic Problem Formulation

There are many options for transforming the problem of decision under uncertainty into an equivalent deterministic one. The most rigorously developed approaches are probably those that incorporate expected utility, though practical considerations may require the relaxation of some assumptions. These methods have been used in practice [115] [62] [63]. The most recent utility planning documents that were reviewed for this effort, however, relied on a trade-off between expected cost of electricity and some measure of risk, when they used a formal process at all (see Table 2). In order to make this effort more relatable to the current status quo, the mean/risk paradigm will be used. However, some of the methods developed

in this effort can be applied more widely, including within a utility theory framework; for more discussion on this, please see the "future work" section in the final chapter.

In this research, robust design decisions will be made as a trade between measures of *expected performance* and measures of *risk*. This is not the exact terminology typically used in robust engineering, which tends to use "performance" and "variation" (or its opposite, "robustness"). Nor is it used in finance: since finance has only a single benefit measure, it is simply called by name, as "return" or "cost", while the term "risk" is the same. In this case, since the application is related to finance, the term "risk" will be used; and since cost may not be the (only) objective, the general term "performance" will be used. So for any single stochastic objective, measured by a "performance" metric, there will be a trade between "expected performance" or simply "mean", and "risk".

In all test cases in this study, a single stochastic objective will be used, resulting in a single measure of expected performance and a single measure of risk. This is directly applicable to current energy portfolio selection problems, where usually only cost and cost risk are considered as objectives (see Table 2 for risk metrics used by a selection of utilities). The methods developed for this study should be usable without significant modification in cases with multiple stochastic objectives, but a demonstration of this is beyond the scope of the study.

3.3.3 Choice of Risk Measure

It has been established that this research will use the twin objectives of "mean" and "risk". Up until now, however, the measure of risk has been left generic and non-explicit.

The robust design literature tends to favor standard deviation or variance as a risk measure [68][22][69][17][56]. In the financial literature, this is consistent with the usage found in the genesis of portfolio theory, dating back to Markowitz in 1952 [75]. Several energy utilities also use standard deviation as a measure of risk in their planning documents [35][8]. However, standard deviation has fallen out of favor as a risk measure in finance. The risk metrics used by 11 different utilities in their Integrated Resource Plans is shown in Table 2.

Table 2: Some Electric Power Utilities and their Choices of Risk Metric

Utility	Risk Metric	Decision Method	Year of IRP	Ref
Ameren Missouri	VaR	Weighted sum	2011	[2]
Xcel Energy	VaR	Informal	2010	[116]
Pacific Gas and Electric	VaR	Multi-Criteria	2007	[89]
Northwestern	VaR - μ	Portfolio Theory*	2009	[84]
Puget Sound Energy	CVaR	Informal	2011	[95]
PacifiCorp	CVaR	Portfolio Theory*	2011	[90]
Entergy Louisiana	σ	Portfolio Theory*	2010	[35]
Avista	σ	Portfolio Theory*	2009	[8]
Progress Energy Carolinas	sensitivity	Informal	2009	[93]
Idaho Power	N/A	Informal	2011	[53]
Florida Power and Light	N/A	Informal	2010	[37]

*All Portfolio Theory approaches also used scenarios for carbon price

The different utilities surveyed used a variety of risk metrics, with value at risk or a variant the most common [2][84][89][116]. Value at risk (VaR) is defined for a particular probability level, usually 5%. In finance, the 5% VaR is is simply the value of loss which is expected to occur less than 5% of the time, see Figure 11(a), or simply the value of the 5th percentile of the predictive distribution of the losses [58]. Explicitly, for a maximization problem:

$$VaR_\alpha = F^{-1}(\alpha) \tag{4}$$

Where α is a probability level (again, usually taken to be 0.05), and F^{-1} is the inverse cumulative distribution function of the returns.

For an energy utility, VaR is applied to the cost of energy. In this context, lower values are better, and there is no "loss" or "gain", simply higher or lower costs. Rather than a 5% VaR, then, the utility would instead be concerned with the 95% VaR, the cost which energy is expected to remain below 95% of the time. Again, it is simply a percentile of the cost distribution.

This concept of risk also jibes with the general definition of risk proposed by Kaplan and Garrick, who define a risk as a consequence combined with a probability [59].

The financial literature also uses Conditional Value at Risk (CVaR) or Expected Shortfall, which is shown to have theoretical advantages by [6]. CVaR is also specified at a

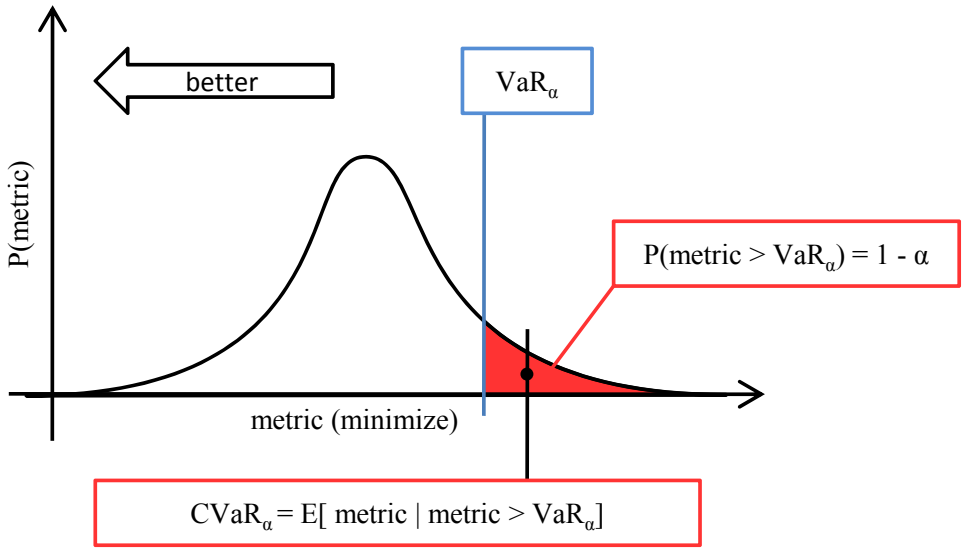


Figure 11: Performance/Risk Terminology. Value at Risk (VaR_α) is the αth percentile. Conditional Value at Risk (CVaR_α) is the expected value given that the metric exceeds VaR_α

probability level, generally 5%, and is simply the expected value of the losses that are below a certain VaR level, or:

$$\text{CVaR}_\alpha = \frac{1}{\alpha}\int_0^\alpha \text{VaR}_\gamma d\gamma \tag{5}$$

Where α is the probability level, and the variable of integration γ is the probability level for the VaR measure. For a minimization problem, this is re-stated as:

$$\text{CVaR}_\alpha = \frac{1}{1-\alpha}\int_\alpha^\infty \text{VaR}_\gamma d\gamma \tag{6}$$

This is shown conceptually in Figure 11. At least two energy utilities use CVaR as a portfolio selection criteria [90][95].

In the context of this research, the measure of risk chosen is not a central question. All three measures discussed here require quantification of the aleatory distribution of the response, due to uncertainty in the environmental noise variables. If the aleatory response distribution is known, the standard deviation, VaR, or CVaR can be computed. Later results from the literature will be shown that allow analytical computation of standard deviation, and a "VaR-like" risk metric will be used for the test cases, for the sake of computational

speed. The overall method is general enough that it is meaningful regardless of what risk measure is chosen.

For the sake of notation, aleatory mean will be written as μ_a, and an un-specified aleatory risk measure will be written as ρ_a.

- Aleatory mean is μ_a
- Aleatory risk (generic) is ρ_a

CHAPTER IV

BAYESIAN SURROGATE MODELS

Before discussing specific approaches to Robust/Multi-Objective/Expensive problems, it is worthwhile to briefly touch on surrogate modeling, since it is found in many diverse approaches and will be central to the new approach presented later.

Briefly, surrogate models are simply regressed fits which model the responses of a simulation code as a function of its inputs. They are a representation of the designer's knowledge about a simulation, and predict the simulation responses at un-sampled input settings.

Early robust design exercises used polynomial equations, called Response Surface Equations [113][81], which due to their polynomial form can only accurately model relatively smooth spaces, and in the quadratic forms typically used can only model unimodal spaces. More complex spaces, with multiple modes or more non-linear behavior, can be modeled with non-parametric models, that is, models that make less restrictive assumptions about the shape of the space (though they still assume some degree of smoothness). Non-parametric models include Radial Basis Functions (RBFs)[54], Multi-Adaptive Regressive Splines (MARS), Neural Networks [104], and Bayesian treatments such as Kriging [98] (also known as Gaussian Process models) and linear Bayesian models [18].

A full and detailed review of surrogate modeling methods is not attempted here; there are many good textbooks, for example Bishop [18]. However, a brief overview of *Bayesian* surrogates is warranted, since they are central to many advanced methods, and will be important to the proposed method.

Any prediction of a simulation response at an un-sampled point will be subject to epistemic uncertainty. In non-Bayesian regression, an effort is made to ensure that this uncertainty is below some acceptable threshold, using goodness of fit metrics such as R^2 and cross-validation error.

In Bayesian regression, on the other hand, the epistemic uncertainty of the model is

quantified directly using an application of Bayes' Theory. Based on some prior knowledge of the space, the designer will impose some kind of assumptions about the shape of the space and how epistemic uncertainty will behave. Based on this prior and informed by the data, a Bayesian model will give a probability distribution representing the epistemic uncertainty at un-sampled points.

4.1 Linear Bayesian Surrogates

The following section follows the textbook by Bishop [18] unless otherwise noted, with some minor notational differences.

4.1.1 Least Squares Regression

Assume that the designer has data consisting of a series of observations, $\{X^{(n)}\}$, for $n = 1 \dots N$, where N is the total number of observations. Say there are M variables, and each observation is a row vector:

$$X^{(n)} = [X_1, \dots X_m, \dots X_M]$$

All the N observations together, taken as a whole, are a matrix:

$$\mathbf{X} = \begin{bmatrix} X^{(1)} \\ \vdots \\ X^{(n)} \\ \vdots \\ X^{(N)} \end{bmatrix}$$

Each observation $X^{(n)}$ will also have a response value, $T^{(n)}$, so the entire set of response observations is:

$$\mathbf{T} = \begin{bmatrix} T^{(1)} \\ \vdots \\ T^{(n)} \\ \vdots \\ T^{(N)} \end{bmatrix}$$

For convenience, the set of all input and output variables will be called $\mathbf{D} = (\mathbf{X}, \mathbf{T})$, the "data set".

In a non-Bayesian context, a linear model is a regression model where the data is explained by a series of *basis functions*, $\phi_l(X)$, each multiplied by a linear *coefficient*, w_l. In linear algebra notation, this can be written as:

$$\hat{T}(X, \mathbf{w}) = \mathbf{w}^T \phi = \begin{bmatrix} w_1 \\ \vdots \\ w_l \\ \vdots \\ w_L \end{bmatrix}^T \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_l \\ \vdots \\ \phi_L \end{bmatrix} = w_1\phi_1(X) + \cdots + w_l\phi_l(X) + \cdots + w_L\phi_L(X) \quad (7)$$

These basis functions can be any function of X, from the familiar polynomials of response surfaces, to sine waves, to sigmoids, to Gaussians. In practice, one of the bases should always be a constant, to act as a bias term.

In least squares regression, the weights in the vector $\mathbf{w} = [w_l \dots w_L]^T$ are adjusted so that the sum of squares error over the dataset is minimized. This can be easily accomplished with a matrix inversion. First, a matrix $\mathbf{\Phi}$ is created:

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1(X^{(1)}) & \phi_2(X^{(1)}) & \dots & \phi_L(X^{(1)}) \\ \phi_1(X^{(2)}) & \phi_2(X^{(2)}) & \dots & \phi_L(X^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(X^{(N)}) & \phi_2(X^{(N)}) & \dots & \phi_L(X^{(N)}) \end{bmatrix}$$

This is called the *design matrix*, and it contains the effects of all of the input data fed through all of the basis functions. To find the least-squares estimate of $\mathbf{w}$, the details will be omitted, but the result is:

$$\mathbf{w}_{ML} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{T} \quad (8)$$

Here the subscript ML denotes that from a Bayesian perspective, this represents the *most likely* vector $\mathbf{w}$ given the data $\mathbf{D}$.

4.1.2 Bayesian Regression

From a Bayesian perspective, $\mathbf{w}_{ML}$ is only the most likely $\mathbf{w}$, not the whole story. The data $\mathbf{D}$ does not represent complete knowledge about the whole space, and so $\mathbf{w}$ can be considered a random variable. It is assumed that the "true" space really does follow the form of the chosen basis functions, with deviations caused by zero-mean random error ϵ:

$$\epsilon = \mathcal{N}(\epsilon|0, \beta^{-1})$$

Here $\beta = 1/\sigma^2$ is the *precision* of the random noise. Precision is specified rather than variance for later notational convenience.

Assuming some prior distribution $p_0(\mathbf{w})$, Bayes theorem allows the calculation of a *posterior distribution* on the weights, given the data. A prior is assumed of the form:

$$p_0(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{S}_0)$$

Which means that the sign of $\mathbf{w}$ is not known (thus zero mean), and the covariance matrix $\mathbf{S}_0$ is the prior assumed covariance between all of the weights. For a moment, assume that it is known; this means that the designer has some kind of prior knowledge about how much the weights will deviate from zero. For computational tractability, and because there generally is no prior reason to think that any one weight would be correlated with another, $\mathbf{S}_0$ will be assumed to be a diagonal matrix:

$$\mathbf{S}_0 = \text{diag}[\alpha_1^{-1} \dots \alpha_l^{-1} \dots \alpha_L^{-1}]$$

Where the elements of the diagonal, α_l^{-1} are the prior variances on the weight distributions. Again, the α_l elements are *precisions*.

A derivation is not given here (see Bishop for readable explanations [18]), but an application of Bayes' theorem using the design matrix $\mathbf{\Phi}$ and given the response data $\mathbf{T}$ yields the posterior distribution on $\mathbf{w}$, which is a joint normal distribution of the form:

$$p(\mathbf{w}|\mathbf{T}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \tag{9}$$

Where:

$$\mathbf{m}_N = \beta \mathbf{S}_N \mathbf{\Phi}^T \mathbf{T}) \tag{10}$$

$$\mathbf{S}_N = (\mathbf{S}_0^{-1} + \beta \mathbf{\Phi}^T \mathbf{\Phi})^{-1} \tag{11}$$

The current situation, then, is that for a *known* global error given by precision β, and a known set of prior precisions for the weights, $\{\alpha_l\}$, posterior distributions on the weights can be found using (9). These posterior distributions will quantify the Bayesian uncertainty in the model.

4.1.3 Predictive Distribution

From this model uncertainty, as quantified by $p(\mathbf{w}|\mathbf{T})$, a predictive distribution on the response of the model can be derived. Again, the derivation is not provided here, but the result is, for any new un-sampled point X:

$$p(\hat{T}|X, \mathbf{T}, \mathbf{S}_0, \beta) = \mathcal{N}(\hat{T}|, \mu_{\hat{T}}, \sigma^2_{\hat{T}}(X)) \tag{12}$$

$$\mu_{\hat{T}}(X) = \mathbf{m}_N^T \phi(X) \tag{13}$$

$$\sigma^2_{\hat{T}}(X) = \frac{1}{\beta} + \phi(X)^T \mathbf{S}_N \phi(X) \tag{14}$$

Where $\hat{T}$ is the prediction, $\mu_{\hat{T}}$ is its mean, and $\sigma^2_{\hat{T}}(X)$ is its variance.

There is a problem, however: this formulation requires that the designer know the global precision β with perfect accuracy. It also requires a prior on the weight precisions, $\{\alpha_l\}$, and an unrealistic mis-specification of these values may produce a poor fit.

Ideally, the designer could add another layer of Bayes theorem, and specify a "hyper-prior" distribution on the $\{\alpha_l\}$ and β priors, then let the data inform the posteriors of the priors. In practice, adding layers of Bayes-ification becomes computationally intractable, and various approximations are used.

4.1.4 Evidence Approximation

In Evidence Approximation, also called Type-II maximum likelihood estimation, the *most likely* values of $\{\alpha_l\}$ and β are found, and these are used as point estimates. An expression for the marginal likelihood of the data, given values of $\{\alpha_l\}$ and β, is maximized numerically.

In practice, it is numerically easier to maximize the log of the likelihood function, which is presented here without derivation. For details, the reader is directed to MacKay [74].

$$\ln p(\mathbf{T}|\mathbf{S}_0, \beta) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}\mathbf{T}^T\mathbf{C}^{-1}\mathbf{T} \tag{15}$$

Where:

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \mathbf{\Phi}\mathbf{S}_0\mathbf{\Phi}^T \tag{16}$$

Here, β and $\{\alpha_l\}$ are inputs to (16), and they can be adjusted with an optimizer to maximize (15).

4.1.5 Linear Bayesian Models in Practice

Many details of the implementation of Bayesian linear models have been concealed thus far by not specifying a particular set of basis functions, $\{\phi_l(X)\}$. If the designer has a good idea of what the space looks like, then a "parametric" model that matches the designer's expectations can be used. For example, if the designer has a good idea that the space will look like a polynomial, then Response Surface Equation polynomials could be used. However, if the functions that are chosen are not capable of representing the true function very well, the fit will be poor.

If there is little knowledge of the space, and it is potentially multi-modal or "poorly behaved", the designer may wish to use non-parametric models. Radial Basis Functions, often used in a non-Bayesian context, can be used to approximate arbitrary landscapes, as interpolators [18]. The basic idea is to center a function around each data point that locally influences the predicted values. A common choice is the Gaussian, which takes the form:

$$\phi_n(X) = \exp\left(-\frac{\|X - X^{(n)}\|^2}{2h^2}\right) \tag{17}$$

Where h is a global parameter common to all of the basis functions, and the $\|...\|^2$ represents Euclidean distance. With radial basis functions, there is one function per data point; for a non-Gaussian regression, a properly fitted RBF is an interpolator.

The term h is left up to the designer to specify; it is a *tuning parameter*, which can be thought of as the size of the surface features being modeled. In practice, a poorly chosen h can result in a poorly fitting model.

To select h in practice, classical frequentist notions of fit error can be combined with the Bayesian models, and cross-validation methods can be used to iteratively find a value that provides good fit.

The function given in (17) belongs to a class of functions called *kernel functions* that operate on pairs of points, represented generically as $k(X, X')$.

4.1.6 Encoding Epistemic Uncertainty

Though this section has glossed over most of the mathematical details, the treatment given here is just about sufficient to program a Bayesian linear regression code. Most importantly, it should be sufficient to understand one of the methods proposed in Chapter 6. The crucial nugget of take-away information is simply this: assuming that the model form is correct, a Bayesian linear model **encodes its epistemic uncertainty in the posterior distribution of the weights**, $\mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ (9). All uncertainty shown by the model is a function of this multivariate Gaussian distribution.

4.2 Gaussian Process Models

In their influential 1989 paper, Sacks *et al.* advocated the use of Bayesian models for computer experiments. Notably, they advocated the use of Gaussian Process (GP) models [98]. In a GP model, the un-sampled output of a computer code is treated as a random process. In most implementations, a GP model treats already sampled points as known with perfect accuracy. This makes sense for many computer modeling applications, where a set of inputs will always produce the same outputs. Uncertainty about the predicted function will increase with distance from samples.

In a general GP model, the true function $T(X)$ is assumed to be a realization of a Gaussian random process that is a function of the space, $G(X)$. It can be assumed to have a linear prior mean function, $\phi(X)^T\beta$. This is the approach followed here. The math follows primarily O'Hagan [88], and to some extent Forrester [38]. Like in O'Hagan's papers, it will be assumed that there are weak prior distributions on the linear function weights, β, and the global variance parameter, σ^2. The latter technically makes the surrogate a t-Process (tP), rather than a Gaussian Process [88], though the term "Gaussian Process" will be used

regardless. It is further assumed that all hyperparameters in the correlation function have fixed (optimized) values.

4.2.1 Gaussian Process Model Regression

As in the linear Bayesian model, it is assumed that the simulation code has been run, and data has been collected. The input data consists of individual observations:

$$X^{(n)} = [X_1, \ldots X_m, \ldots X_M]$$

All the N observations together, taken as a whole, are a matrix:

$$\mathbf{X} = \begin{bmatrix} X^{(1)} \\ \vdots \\ X^{(n)} \\ \vdots \\ X^{(N)} \end{bmatrix}$$

The entire set of N response observations is:

$$\mathbf{T} = \begin{bmatrix} T^{(1)} \\ \vdots \\ T^{(n)} \\ \vdots \\ T^{(N)} \end{bmatrix}$$

The data set will be called $\mathbf{D} = (\mathbf{X}, \mathbf{T})$.

It is assumed that the GP has a linear model as a prior. For the purposes of this dissertation, the linear model will consist of a mean term and one *linear term* for each input dimension (note that the two words *linear* here have different meanings).

$$\phi(X) = \begin{bmatrix} 1 & X_1^{(1)}, & \ldots & X_m^{(1)}, & \ldots & X_M^{(1)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & X_1^{(n)}, & \ldots & X_m^{(n)}, & \ldots & X_M^{(n)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & X_1^{(N)}, & \ldots & X_m^{(N)}, & \ldots & X_M^{(N)} \end{bmatrix} \tag{18}$$

The responses at any pair of points $X^{(i)}$ and $X^{(j)}$ are assumed to covary as:

$$\mathrm{Cov}[T(X^{(i)}), T(X^{(j)})] = \hat{\sigma}^2 k(X^{(i)}, X^{(j)}) \tag{19}$$

The term $k(X^{(i)}, X^{(j)})$ is a *kernel function*:

$$k(X^{(i)}, X^{(j)}) = \exp\Big(-\sum_{m=1}^{M} \theta_m |X_m^{(i)} - X_m^{(j)}|^{p_m}\Big) \tag{20}$$

where m indexes over the input dimensions. Note that the terms θ_m and p_m are un-defined thus far, and will be tuning parameters eventually. Notationally, they will be referred to as vectors, $\theta = \{\theta_m\}, \mathbf{p} = \{p_m\}$.

When the a correlation kernel of this form is used, the model can be called a *Kriging* model, perhaps the most commonly used GP for engineering design.

Now a correlation matrix can be constructed:

$$\mathbf{\Psi} = \begin{bmatrix} k(X^{(1)}, X^{(1)}) & \dots & k(X^{(1)}, X^{(N)}) \\ \vdots & \ddots & \vdots \\ k(X^{(N)}, X^{(1)}) & \dots & k(X^{(N)}, X^{(N)}) \end{bmatrix} \tag{21}$$

4.2.2 Estimating the Tuning Parameters

Before the model can be used for prediction, the parameters θ and $\mathbf{p}$ must be tuned. Because a full Bayesian treatment would be computationally unwieldy, it can be approached with a Maximum Likelihood method. Derivation will not be provided here, but computationally, it involves the following steps. First, a maximum likelihood estimate is found for a global variance parameter:

$$\hat{\sigma}^2 = \frac{1}{N - L - 2} \mathbf{T}^T (\Psi^{-1} - GWG^T)\mathbf{T} \tag{22}$$

Where L is the number of basis vectors in the prior, and the terms G and W are defined as:

$$G = \Psi^{-1}\phi \tag{23}$$

$$W = (\phi^T \Psi^{-1} \phi)^{-1} \tag{24}$$

These are then used to create a ln-likelihood function,

$$\ln(p(\mathbf{T}|\theta,\mathbf{p}))) \approx -\frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln|\Psi| \tag{25}$$

The ln-likelihood function can be fed into an optimizer, and maximized as a function of θ and $\mathbf{p}$. In practice, all $\mathbf{p}$ values can be set to 2.0 and only the θ modified.

This completes the regression stage.

4.2.3 Prediction with a Kriging Model

Now say that the model is to be used for prediction at an un-observed data point, x.

A vector of correlations is constructed between the observed data and the new point:

$$\psi = \begin{bmatrix} k(X^{(1)},x) \\ \vdots \\ k(X^{(N)},x) \end{bmatrix} \tag{26}$$

Now the prediction $\hat{T}(x)$ is Gaussian:

$$\hat{T}(x) = \mathcal{N}(\hat{T}|\mu_{\hat{T}},\sigma^2_{\hat{T}}) \tag{27}$$

and the posterior epistemic mean $\mu_{\hat{T}}$ and variance $\sigma^2_{\hat{T}}$ can be calculated [38][60][88] from:

$$\mu_{\hat{T}}(x) = \phi(x)^T\hat{\beta} + \psi^T\Psi^{-1}(T - \phi(\mathbf{X})\hat{\beta}) \tag{28}$$

Where $\phi(x)$ is the prior basis functions evaluated at the new point x, and $\phi(\mathbf{X})$ is the same set of basis functions evaluated for the original data $\mathbf{X}$. The term $\hat{\beta}$ is the posterior predictive mean of the basis function weights:

$$\hat{\beta} = WG^T\mathbf{T} \tag{29}$$

For any pair of un-sampled points $x^{(i)}$ and $x^{(j)}$, the predictive *covariance* can be calculated as:

$$\begin{aligned}\mathrm{Cov}[\hat{T}(x^{(i)}),\hat{T}(x^{(j)})] = \hat{\sigma^2}[&k(x^{(i)},x^{(j)}) - \psi(x^{(i)})^T\Psi^{-1}\psi(x^{(j)}) \\ &+ \{\phi(x^{(i)}) - G^T\psi(x^{(i)})\}^T W\{\phi(x^{(j)}) - G^T\psi(x^{(j)})\}]\end{aligned} \tag{30}$$

Where $\hat{\sigma^2}$ is the same global variance parameter used in regression.

Note that unlike in a linear Bayesian model, epistemic uncertainty is not encoded in a joint posterior weight distribution. Instead, it is expressed through the correlation structure. This will be important in a later section, namely in the discussion of Monte Carlo methods for the calculation of second-order probabilities.

CHAPTER V

SAMPLING METHODS FOR SIMULATION-BASED ROBUST DESIGN

The previous chapter described surrogate modeling methods for estimating the response of a simulation code based on sampled data. But how are the data samples selected? The usefulness of the surrogate will depend critically on the placement of samples both in design and noise space.

This chapter describes the sampling methods found in the robust design literature. The methods are broadly classed as design of experiments, when all points are selected prior to evaluating the simulation code, and sequential sampling, when information from previous samples is used to select new samples. Bear in mind that the overall objective is to find a mean/risk frontier, which is the domain of *optimization*. However, the optimization literature does not generally refer to design of experiments, even though DoE can be used for purposes of optimization. Optimization almost always refers to sequential sampling methods.

Note also that there are many optimization methods that do not fit surrogates to the model at all. When surrogates are fit to the data, they can be used for more than just optimization; they can also be used for visualization and exploration. So a sequential sampling method that fits surrogates to the data is more than just an optimization method, though it might well be used effectively for optimization.

5.1 Design of Experiments

The earliest robust design methods, dating back to Taguchi [107], were based on a design of experiments approach, and there are many iterations and modifications that have arisen since.

A design of experiments approach uses a fixed *design*, or list of simulation cases to run. Response data is collected, and the effects of the various factors can be estimated, as well as

higher-level effects and interactions between the factors. The more cases are run, the more complex the effects that can be discerned. Generally, in engineering design, a surrogate model is fit to the data.

In Type I robust design, where noise variables are present, there are two basic types of experimental designs: *crossed arrays* and *combined arrays*. The two approaches differ in their treatment of design and noise variables.

5.1.1 Crossed Array Designs

The earliest Robust Design approaches, proposed by Taguchi, used a set of "crossed" or "inner and outer" arrays to deal with design and noise variables. An "inner" design of experiments specified settings of the design variables. At every run in this array, a full set of noise cases were run, specified by the "outer" array. This configuration is shown conceptually in Figure 12. The noise runs were used to find some measure of robustness [107]. These noise cases can be a specific design, as practiced by Taguchi, or they can be a series of Monte Carlo runs that depend on assumed distributions for the noise variables.

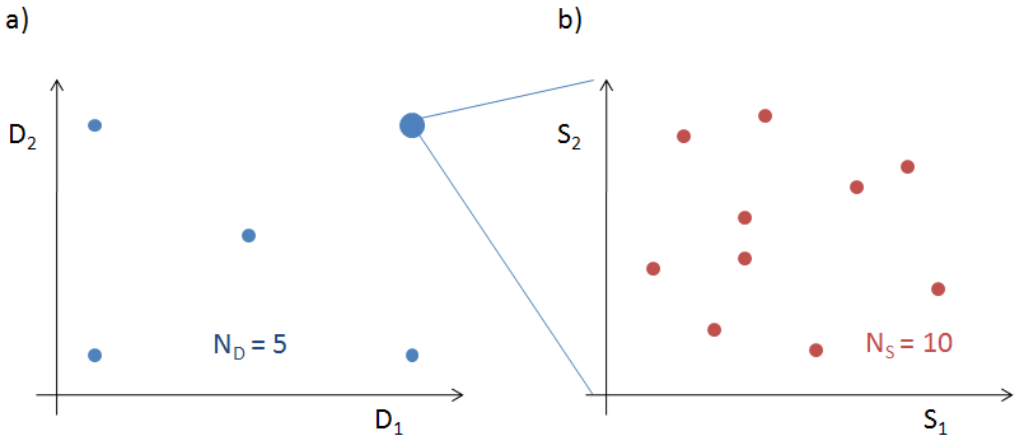


Figure 12: Design of Experiments, crossed arrays. (a) Inner design array ($N_D = 5$) (b) Outer noise array ($N_S = 10$). Total samples for this design is $5 \times 10 = 50$

Taguchi himself did not use surrogate models, but later approaches combined the inner/outer array concept with surrogates [72][81]. Since robustness measures can be found at every design point, a surrogate can be fit to those measures (or to a single measure of robustness, depending on the method used). Thus with an inner/outer array approach, the

input variables to the surrogate are the *design variables only*.

With crossed arrays, the design of the arrays can be chosen independently. Indeed, there is no reason the noise array need even be of the same type as the design array. Since the objective of running the noise array is to estimate the distribution of the result, and not to estimate the shape of the response, the designer can use an array designed specifically for this purpose. Most basically, Monte Carlo simulation can be used to randomly place points in noise space, and the output statistics can be calculated numerically; however, this has low efficiency. A more efficient approach is to use Latin Hypercube Sampling (LHS) [77].

If the designer is able to make samples sequentially, Markov Chain Monte Carlo (MCMC) methods can be used [3], at which point the approach is no longer strictly a design of experiments, because all of the runs cannot be specified beforehand. However, it still retains most of the advantages of a DoE approach to computer experiments: it represents the global design space accurately, and it can still be parallelized by running each design point separately. An advantage of the MCMC approach is that it can be used to find any statistic of interest, including value at risk or conditional value at risk.

Another sequential sampling approach was presented by Rasmussen and Ghahramani in 2003, called Bayesian Monte Carlo sampling. The authors fit a Gaussian Process model, and use it to estimate statistics [96]. Kumar further investigated Bayesian Monte Carlo for use in robust design[67].

5.1.2 Combined Array Designs

One criticism of Taguchi-style inner and outer arrays was that the method was inefficient in terms of the number of experiments required. Many subsequent approaches instead used *combined* arrays, where the design and noise variables were lumped together for the purpose of experimental design, and a surrogate model was fit to both sets of variables simultaneously. This type of design is shown conceptually in Figure 13.

Welch *et al.* first proposed this approach for robust design, and found that for their application the use of a combined array resulted not only in reduced simulation runs but *better* accuracy than an inner/outer array approach. Their robustness metric was a *squared*

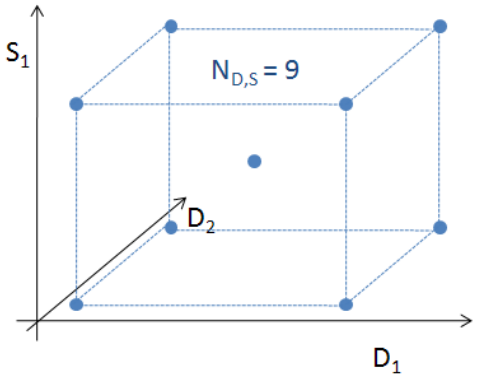


Figure 13: Design of Experiments, combined array

loss metric, or the deviation of performance from a target squared. This loss function, mapped over the design variables, was fit poorly by the polynomial response surface equations that they were using, and this was proposed as the reason for the poor performance of the inner/outer array approach. The "pure" performance function, on the other hand, as a function of both design and noise variables, was fit better by polynomials. This was proposed as the reason for the improved accuracy of the combined approach [113].

Shoemaker *et al.* also compared combined and inner/outer array approaches, and also found a reduction in computational cost with a combined approach. However, they note that the effectiveness of the combined approach depends more critically on how well the surrogate model fits, whereas an inner/outer array approach allows the robustness measures to be estimated directly [103]. This is an important distinction. With a combined array approach, the aleatory noise distributions must be propagated through a surrogate model in order to estimate the aleatory statistics, and any errors in the surrogate model will be propagated as well, resulting in errors in the statistics. With an outer array of noise variables, on the other hand, it may be possible to estimate the statistics more directly, without any surrogate at all; or if a surrogate is used, it is a *local* surrogate of noise variables only.

Chen *et al.* used a combined array approach to robust design, with mean of the response and variance of the response as twin objectives. They fit polynomial response surface equations to the design and noise variables, and then from those used analytic expressions

for the mean and variance of the response, assuming that the noise factors had Gaussian distributions. They choose a final design using the compromise decision support problem (DSP) approach [22].

Mavris *et al.* used the combined array approach in a design method called Robust Design Simulation, or RDS. In RDS, a combined surrogate is fit to both design and noise variables. Then, at design points of interest, Monte Carlo simulation can be used to vary the noise variables, and the aleatory statistics can be estimated by running the surrogates. Since the surrogates are cheap, large numbers of Monte Carlo cases can be run inexpensively. A second design of experiments is chosen, this time only for the design variables, and this Monte Carlo estimation is run to find the aleatory statistics of interest for every case. Finally, new surrogates are fit to those aleatory statistics of interest (Mavris *et al.* use probability of success), as a function of the design variables only [76]. It is worth noting that the accuracy of these surrogates depend not only on their fit, but also on the fit of the original surrogates used to model the responses as a function of both design and noise variables, and on the accuracy of the Monte Carlo methods.

From the literature, it would seem impossible to generalize whether combined array or inner/outer array approaches result in greater accuracy. Welch [113] and Shoemaker [103] found higher efficiency with a combined approach, and Welch even found higher accuracy with the combined array for fewer simulation runs. But Shoemaker points out that the effectiveness of a combined array depends critically on the quality of the fit [103], and indeed a paper by Frey and Li found that in cases where the degree of the true function was greater than the degree of the fit, an inner/outer array approach showed better results than the combined approach [40].

Making generalization even more difficult, these papers all deal with cases where the space is well-behaved and unimodal, and the surrogate fits are all polynomial response surface equations. If the space is multimodal, a more generalized non-parametric model such as Kriging [98][57] or Radial Basis Functions (RBFs) [54] will be more appropriate. In such scenarios, space-filling designs such as Latin Hypercubes are preferred. When a spacial correlation structure is assumed as part of the model structure, as it is in Kriging

and RBFs, designs are desired which maximize *entropy* [102][98][24][77][80]. What that means computationally will be discussed later.

Bates *et al.* used a combined array method with a Kriging model, and compared it with a Taguchi-style crossed array method, but it is difficult to draw conclusions because only a single test case was run, different surrogate types were used in both cases, and neither method performed better than the other [14].

There is a mechanism by which a combined array approach can result in greater efficiency of knowledge use: in an inner/outer array approach, information about the *shape* of the space does not propagate beyond each design point, whereas with a combined array the response model will have global knowledge of noise behavior. Therefore if a combined and a crossed array both have surrogates that make equally efficient use of the information available to them, one would expect the combined array, having more information available to it, would achieve the overall design objectives more efficiently. However, fitting a surrogate to a very large number of data points may be challenging or lead to numerical problems (as will be encountered in a later chapter).

A designer, given a particular robust design problem, will have to choose between using combined arrays or crossed arrays. As a matter of research, this question cannot be answered for all cases. However, there are certain properties of the problem that one would expect to affect the choice. These include difficult-to-quantify properties such as the shape of the space and its non-linearity with respect to noise variables. However, one would also expect that the *number* of noise variables would affect the relative merits of crossed vs. combined arrays, and this leads to a research question:

Research Question 1: For finding mean/risk Pareto frontiers, how does the relative efficiency of combined and crossed arrays depend on the number of noise variables?

"Efficiency" must be defined, and a working definition will be presented later. For now, however, it will suffice to define it imprecisely as meaning better accuracy in representing the Pareto set for the same number of function calls, or fewer function calls for the same

accuracy.

Even this question cannot be answered fully. The possible number of noise variables is infinite, and there may still be interaction effects with other problem characteristics. However, a series of sensitivity experiments around a baseline can lend evidence to support a hypothesis on this matter. Experimental design details will be left to a later section.

Crossed and combined array methods estimate the aleatory statistics using different methods. In crossed array methods, at every candidate design, an experiment is carried out to estimate the statistics. Statistics may be estimated with Monte Carlo methods, which do not suffer in accuracy as the dimensionality of the space increases [18].

Combined array methods, on the other hand, require that the effects of the noise variables be modeled explicitly with a surrogate. Estimates of the statistics will depend on the surrogate fit, as pointed out by [103], and surrogate fit quality suffers as the number of dimensions increases, due to the "curse of dimensionality" [18].

The following hypothesis therefore seems reasonable:

Hypothesis 1: As the number of noise variables increases, the efficiency of combined array methods will suffer relative to the efficiency of crossed array methods.

Note, however, that the reasoning behind this hypothesis is based very large numbers of input variables causing a decrease in the quality of a surrogate fit. For smaller numbers of noise variables, this might not be a strong effect, and there might be other effects that would cause the reverse.

Again, proposed experiments will be left to a later chapter, and it should be pointed out that it will be impossible to generally prove this hypothesis, but a well-chosen experiment should yield evidence one way or the other.

5.2 Sequential Sampling Approaches

In the design of experiments approaches described in the previous section, an experimental design is selected so that over the design variable ranges of interest, there is uniformly or near-uniformly high accuracy. However, in practice the designer may not care equally about

all areas of the design space. Given the opportunity, the designer might willingly sacrifice global accuracy for increased knowledge at or around the Pareto frontier. This can be achieved by using knowledge from previous samples to guide the selection of future samples around the frontier. This is domain of *multi-objective optimization*.

A review of optimization methods applied to robust design problems can be found in Beyer and Sendhoff [17]. This section will restrict its focus to methods that can be applied to a mean/risk decomposition approach. It will also ignore methods which are only applicable to linear programming problems; though such methods can be used for certain simple portfolios, the same factors that motivate the use of simulation in energy portfolio selection also preclude the use of linear programming approaches.

5.2.1 Single Objective, Multi-Objective, and Robust Optimization

Most optimization problems and techniques concern the minimization of a single objective function. In such cases, the optimizer seeks a single design, and only needs to characterize the space to the extent necessary to reject all other portions of the space as inferior.

However, this is a multi-objective problem. At the very least, there are the twin objectives of mean performance and risk, and there may be multiple pairs of such objectives. There are a number of methods for applying optimization techniques to such problems, and collectively they can be referred to as robust design optimization.

Most simply, the multiple objectives can be aggregated into a single objective, and standard optimization methods can be used. However, this requires some *a priori* preference structure, and it is assumed for this problem that no such preference structure is yet known. Therefore, such methods will not be discussed.

If preferences are not known, then the Pareto frontier ("effient frontier") of non-dominated designs must be found. A number of methods exist for finding such frontiers, all of them modified versions of single-objective optimization methods. The most well-known methods are modified evolutionary algorithms. Also found in the literature are statistical improvement methods. The following sections will describe the relevant optimization techniques in the single-objective case only as far as needed to discuss their application to multi-objective

problems; and further, the focus will be on applications to robust design problems.

Multi-objective particle swarm methods can be found in the literature [97], but as there are no papers applying them to robust design problems, they are not discussed here. It is worth noting, though, that any multi-objective optimization method may be used with "crossed" arrays to solve a robust design problem. At every design point explored by the optimizer, the objective functions of mean and risk can simply be estimated with an "outer" array. There is no longer an "inner array", but rather an inner loop, but the term "crossed arrays" will still be used in this scenario.

5.2.2 Evolutionary Algorithms

In an evolutionary algorithm, a set of of candidate designs (called the population) are evolved using a mechanism inspired by biological evolution. The "fittest" (most optimal) individuals are "bred" with each other, producing new "generations" of offspring that bear traits of their parents along with random "mutations". The most popular methods for multi-objective design are evolutionary algorithms, where successive generations have sub-populations that move closer and closer to the true Pareto frontier. Two popular algorithms are NSGA-ii [28] and SPEA2 [117].

A number of sources in the literature have used surrogate models to enhance the performance of multi-objective evolutionary algorithms. These include efforts by Chafekar *et al.*[20], Farina [36], Nain and Deb [82], and Gaspar-Cunha and Vieira [41]. Generally, improved performance was seen.

5.2.2.1 Evolutionary Algorithms for Robust Design

A multitude of authors have used evolutionary algorithms for the purpose of robust design. Jin and Branke reviewed the state of the art in evolutionary algorithms applied to robust design in 2005 [55], and Beyer and Sendhoff also review several instances in the literature [17]. A subset of those efforts have focused on finding Mean/Risk frontiers for Type I problems. Sharma *et al.* use NSGA-ii to optimize mean and variance; at every design point, they run 5,000 Monte Carlo samples to estimate the aleatory statistics [100]. Jin and Sendhoff trade between variance and nominal (rather than mean) value, and they mention

an application to Type I robust design, though the examples are for Type II [56]. Tan and Goh consider the case of multiple stochastic objectives, each broken into a measure of risk and a measure of central tendency (though they, like Jin and Sendhoff, consider nominal rather than mean performance) [108].

A surrogate-enhanced evolutionary algorithm was used specifically for Type I robust design by Kumar [67]. His algorithm is used to solve the exact type of problem proposed here, and a conceptual description is shown in Figure 14. Kumar's method is a crossed-array-type method. First, Kumar runs a sparse design of experiments in design space. At every design, he uses a method called Bayesian Monte Carlo, wherein he fits a Gaussian Process surrogate model to the *noise variables only*, and uses this to estimate the aleatory mean and standard deviation for that design. He also uses that Kriging model to estimate the *uncertainty* in those statistics, and if the uncertainty is too high, he samples additional noise points, until he has adequate estimates of the aleatory mean and standard deviation for all design points.

Kumar then fits two separate Kriging models, one to the aleatory mean and one to the aleatory standard deviation. He uses these Kriging models to enhance an NSGA-ii multi-objective optimizer, by optimizing the Kriging models and running the full Bayesian Monte Carlo only at the optimal set of points, once the optimizer converges.

An (admitted) flaw in Kumar's method is that it does not *explore* the space very well. If, after the initial population, a region of the space *is* a part of the true Pareto frontier but is not *thought* to be, the optimizer may never reach it. The method sounds promising for solving Type I robust design problems with fewer function calls than DoE methods, though no comparisons are given and the relative efficiency is unclear.

To reduce function calls further and to increase exploration, Kumar tries a modified method. He reduces the number of Bayesian Monte Carlo samples and allows a higher epistemic uncertainty in the aleatory statistics. For his twin Kriging models of aleatory mean and standard deviation, he uses a modified form that fits to noisy functions, and uses a modified fitness function that allows for a "fuzzy" Pareto frontier.

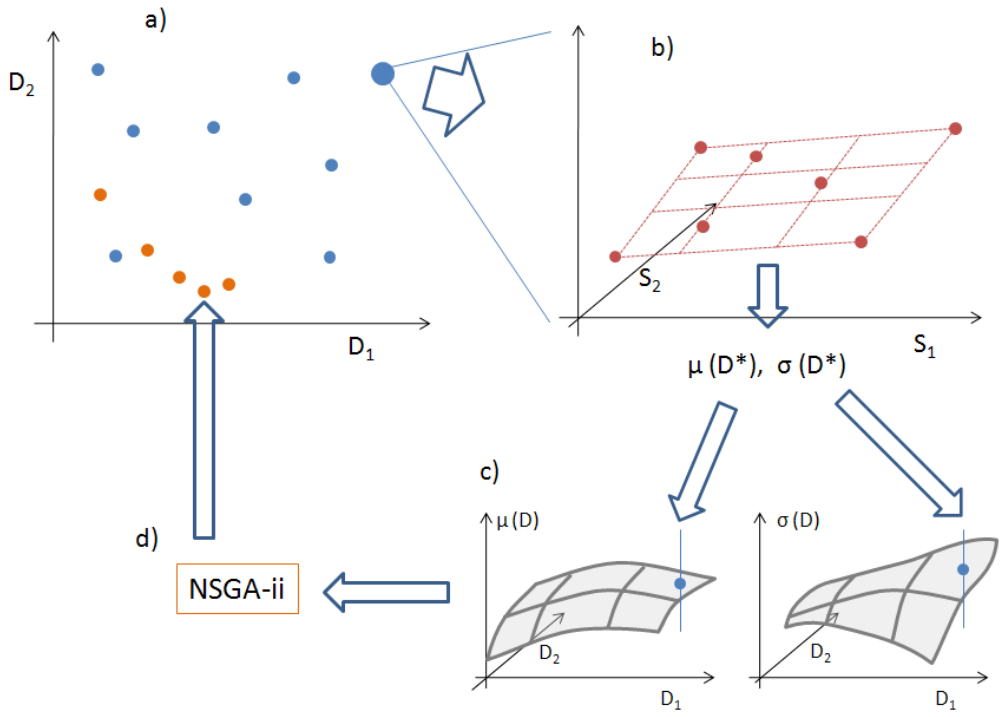


Figure 14: Kumar's method for optimization of mean and standard deviation. For a population of designs $D_1^*...D_N^*$ (a), Bayesian Monte Carlo Simulation (BMCS) is used to estimate the mean and standard deviation (b). These are used to update Kriging models over the design space (one for each statistic) (c), and NSGA-ii is used to optimize with these models (d). The Pareto population is fed back into step (a) [67]

This method has many characteristics which potentially can reduce the number of function calls. It replaces Monte Carlo sampling on the noise variables with a "surrogate-enhanced" Monte Carlo that uses fewer function calls. It also uses surrogates to enhance an NSGA-ii optimizer. The lack of exploration of the previous method is potentially ameliorated as well. However, no implementation results are given, and it is unclear whether this method will perform well.

5.2.3 Statistical Improvement Methods

Statistical improvement methods are a class of optimization methods that rely on Bayesian surrogate models to guide successive samples. First introduced in some form by Mockus *et al.* in 1978 [79], its usage became more widespread after it was picked up by Cox and John in 1997 [23], and implemented in its now-common form by Jones *et al.* in 1998 as Efficient Global Optimization, or "EGO" [57]. Descriptions can be found in several texts, including

Keane and Nair [61] and Forrester [38].

For a *single* objective, statistical improvement methods work as follows. Aspects are shown graphically for a simplified one-dimensional case in Figure 15. First, a sparse design of experiments is run, the current best design D^{best} is noted, and a Bayesian surrogate is fitted. This surrogate must have zero uncertainty at un-sampled points; in practice, this generally means a Gaussian Process/Kriging model.

Next, at all candidate designs which might be sampled, the posterior predictive distribution is used to find either either the *Probability of Improvement* or the *Expected Improvement* relative to D^{best}.

The Probability of Improvement, or P(I) is just the probability that a design sampled at a particular design, D^*, will be better than the current best design, D^{best}, or (for a minimization problem) [38]:

$$P[I](D^*) = \int_{-\infty}^{D^{best}} p_Y^{D^*}(\delta) d\delta \tag{31}$$

Where $p_Y^{D^*}(\delta)$ is the posterior probability density function for the objective Y at design D^*.

Another criteria which can be used is the Expected Improvement, or E[I], which is defined as:

$$E[I](D^*) = \int_{-\infty}^{D^{best}} \delta \cdot p_Y^{D^*}(\delta) d\delta \tag{32}$$

$$\tag{33}$$

And can be thought of as [57]:

$$E[I](D^*) = E[\max(0, Y^{best} - Y(D^*))] \tag{34}$$

An optimization is performed over the space to find the candidate design with the greatest E[I] or P(I), and this point is sampled. The Bayesian surrogate is updated, and the process is repeated.

Statistical improvement methods automatically trade between *exploring* the space to ensure that good regions don't go overlooked, and *exploiting* its current best guess of where the best regions lie. In areas where the epistemic uncertainty is high due to lack of samples,

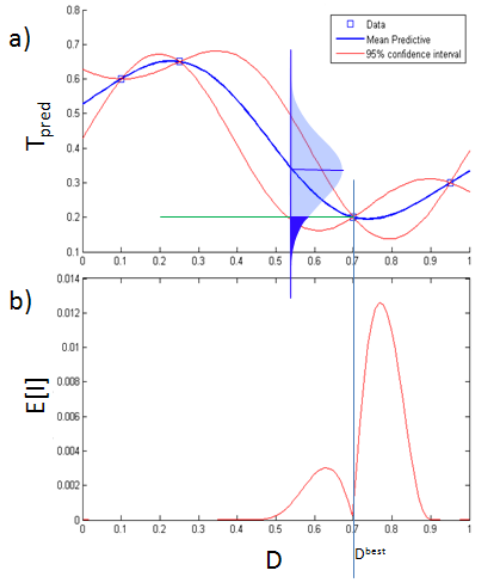


Figure 15: Statistical Improvement method, (a) Bayesian Surrogate fit to data, and the predictive distribution shown at a single point. The current best point D^{best} is marked with a vertical line, and the probability density below D^{best} is shown shaded (b) Expected Improvement for all D.

the E[I] and P(I) will be high due to the long tail of the distribution; and if expected value of a region is high, E[I] and P(I) will also be high. The method will *not* sample at existing data points, because the uncertainty is zero and therefore the probability of improving is zero. It is more efficient than a Design of Experiments global sampling approach because it takes few samples (that is, allows high uncertainty) in regions of the design space where it is confident that the performance is poor.

This method has been shown to work very well in practical design problems [57]. Sobester *et al.* compare the method with different sizes of initial DoE samples. They find that the effectiveness of the method does depend on this initial sample, and that there is an optimal value. Therefore, they implicitly find that E[I] works better than a pure design of experiments, for the functions and sample sizes they tested [105].

A point made by Sobester *et al.* is that if the true objective function is known to be well-behaved and unimodal, there is really no need to balance between exploring unknown areas and exploiting the expected optimum; a good strategy might be a simple "greedy" strategy of sampling where the surrogate thinks the function is best [105]. However, if the

function is multi-model or less predictable, this could result in finding a local optimum.

In practice, expensive simulation codes are often run on large computer clusters, and it is therefore desirable to parallelize any optimization algorithm. E[I] and P(I) are not trivial to parallelize. The problem was explored by Schonlau in his 1997 PhD thesis [99]. Ginsbourger *et al.* further discuss methods for selecting multiple designs to run simultaneously, which they refer to as q-E[I]. The challenge is essentially one of of finding a set of q points that collectively reduce uncertainty about the location of the optimum. They define the q-E[I] metric as:

$$\mathrm{E}[I](D^{n+1},\ldots,D^{N+q}) = E[\max\{(Y^{best} - Y(D^{N+1}))^+,\ldots,(Y^{best} - Y(D^{N+q}))^+\}] \quad (35)$$

Where again Y^{best} is the current best sampled point, N is the current number of samples, and the $()^+$ superscript indicates that only improvements are considered. Ginsbourger *et al.* compare several analytical approximations for finding q-E[I], as well as a direct Monte Carlo approach involving random Gaussian Process surfaces [43]. This will not be a focus of this document, but in practice the ability to parallelize the sampling process should prove useful.

5.2.3.1 Multi-Objective Statistical Improvement

There are a number of instances in the literature where statistical improvement methods have been adapted to multi-objective problems. Though each author uses a different term, they will here be collectively referred to as Multi-Objective Statistical Improvement (MOSI) methods.

Emmerich *et al.* explored and tested a number of possible criteria, including a lower-confidence-bounds based method, a method based on an expected increase in hypervolume, and a multi-dimensional expected improvement method. The multi-dimensional E[I] method required multi-dimensional integration, which they suggested could be achieved with piecewise numerical integration or Monte Carlo integration. This method, as well as the hypervolume method, required some kind of relative weighting on the objectives. The authors preferred the lower-confidence-bound method as more numerically tractable and as not requiring weights. They found that, compared to NSGA-ii and a surrogate-enhanced

NSGA-ii, the statistical improvement methods resulted in better exploration of the space and better coverage of the Pareto frontier [32]. In later works, Emmerich *et al.* developed closed-form and computationally efficient methods for computing the hypervolume-based E[I] method [30], and Emmerich provides MATLAB code on his website for a two-objective case [31]. More details on this hypervolume approach will be provided in the next chapter, and extended to the case of an uncertain Pareto set.

Keane derived a (*very* long) closed-form expression for multi-dimensional expected improvement and probability of improvement, for the special case of exactly two objectives [60]. In two engineering test cases, he found that the methods performed better than both NSGA-ii and a surrogate-enhanced version of NSGA-ii.

Another implementation of multi-objective expected improvement can be found in Bautista's PhD thesis [15], called the *EmaX* algorithm. Bautista specifies a minimax metric for expected Pareto improvement which can be implemented using Monte Carlo methods for an arbitrary number of objectives, and she provides test cases with more than two dimensions.

Knowles developed a method called "ParEGO", which on every iteration combines all objectives into a single metric according to randomized weightings. The point with maximum expected improvement in that metric is found and sampled; then the weightings are re-randomized, and the process is repeated. Over time, the frontier is expanded uniformly in all directions of improvement [65]. It should be noted that the single objective metric is chosen such that even non-convex parts of the frontier will be found. In most of the test problems explored, ParEGO outperformed NSGA-ii, though for one test function the reverse was true.

Lastly, Hawe and Sykulski consider a discrete "levels of improvement" metric based on how many existing points a candidate would dominate [48], though they do not provide details.

5.2.3.2 Multi-Objective Statistical Improvement for Robust Design

Of the five MOSI implementations presented here, only Keane presents a robust design test case. The Keane test case uses crossed arrays, with 20 Monte Carlo points sampled at

every design, and separate Kriging models fitted for aleatory mean and standard deviation. In an airfoil design test problem, Keane found that the E[I] and P(I) methods performed slightly better than a surrogate-enhanced NSGA-ii method [60]. Note that there is no reason such a crossed array method could not be used with any of the multi-objective statistical improvement methods found in the literature.

Statistical improvement methods in general have shown significant potential on engineering problems, and multi-objective versions have generally shown encouraging results as well. Note that the baselines for comparison have been NSGA-ii, the multi-objective evolutionary algorithm. There have been no comparisons with Design of Experiments.

This raises a question, however. The designer would *like* to know when to use multi-objective statistical improvement methods for robust design, and when to use design of experiments. Again, the space of possible engineering problems is too vast to make sweeping generalizations. It is reasonable, however, to assess the relative performance of the two methods at a baseline case, and to make predictions about how that relative performance changes as a function of certain important properties of the test case.

A multi-objective optimizer seeks the Pareto frontier, rather than a single point as a regular optimizer does. Conceptually, a larger "fraction" of the space can be considered *optimal*. In the limit of a Pareto frontier that occupies the *entire* design space, it is not likely that any optimizer could outperform a DoE, since a DoE is designed specifically for global accuracy.

This sort of "Pareto fraction" (it might not be unitless, if the frontier has lower dimension than the design space) will increase as the number of objectives increases and decrease as the number of design variables increases. For simplicity, only one of these will be varied, and the question will be asked:

Research Question 2: For finding mean/risk Pareto frontiers, how does the relative efficiency of design of experiments and multi-objective statistical improvement change with the number of design variables?

Again, "efficiency" will need to be defined later.

Also again, it will be impossible to prove that any one method works better than any other, even for a small subset of possible problems. However, an experiment can provide support for a more modest hypothesis. There is good reason to believe that in all but the simplest small-dimensional and predictable DoE cases, the multi-objective statistical improvement methods will achieve greater efficiency by selectively sampling near the Pareto frontier. Looking at the *relative* efficiency of the two methods, it seems likely that greater numbers of design variables will reward statistical improvement methods. An intuitive explanation follows.

Consider that the Pareto frontier has, in a sense, one fewer dimensions than the number of objectives. A one-objective Pareto frontier is a 0-dimensional point, a 2-objective Pareto frontier is a curve (a 1-dimensional path through 2-space), and a 3-objective Pareto frontier is a surface (essentially 2-dimensional). In design space, the frontier will tend to have that same degree of "dimensionality". See for example the 2-objective frontier shown in Figure 16, which is "curve-like" in both objective and design space, though it is broken into sections. If the number of design variables go up, it will still be a curve. So if the number of design variables increase but the number of objectives does not, the dimensionality of the Pareto frontier will diminish relative to the dimensionality of the space. Thanks to the curse of dimensionality, it will become more "local". Say one wished to draw a "tube" of fixed width around the curve in Figure 16. As the number of design dimensions increased, this "tube" would represent a progressively smaller fraction of the total hypervolume.

All this is to say that Multi-Objective Statistical Improvement methods are more "localized", whereas DoE methods are fully "global", and as the number of number of design variables increases the region of interest becomes more "local." This leads to a hypothesis:

Hypothesis 2: As the number of design variables increases, multi-objective statistical improvement methods will become more efficient relative to a design of experiments.

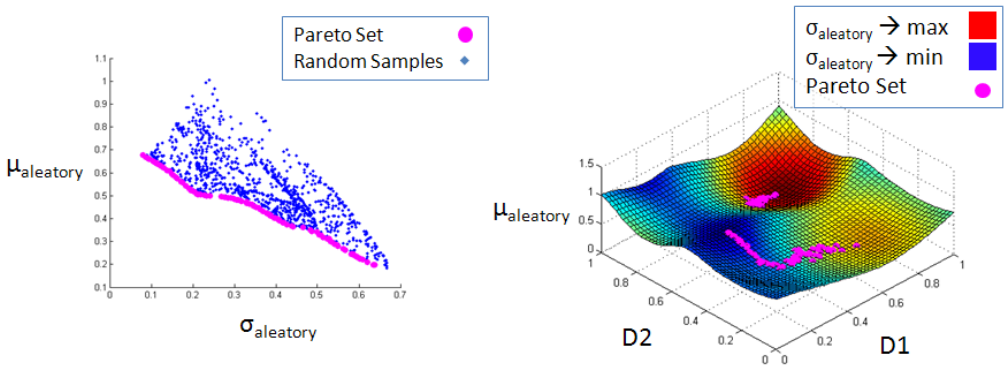


Figure 16: A 2-objective Pareto frontier. In objective space (a), the frontier is a 1-D curve through 2-D space, and will be for any 2-objective problem. In design space (b), it is still essentially a 1-D curve regardless of how many design dimensions there are, though it may have multiple discrete sections.

5.3 A Gap in the Literature

In the literature, two promising methods for Type I mean/risk robust design were identified:

- In a design of experiments approach, combined arrays have the potential to improve computational efficiency relative to crossed arrays
- In an optimization approach, a multi-objective statistical improvement method has the potential for greater computational efficiency relative to other multi-objective optimization methods

There is significant literature on both individually, though the literature on Multi-Objective Statistical Improvement is all quite recent. Both methods rely on global surrogate models of the system response. However, no paper to date, to this author's knowledge, has combined the two. This leads to a direction for research:

> **Research Objective:** Implement multi-objective statistical improvement methods using surrogate models that are functions of both design and noise variables (combined arrays).

This will lead to an investigation of possible implementation details, as well as to new research questions.

CHAPTER VI

MULTI-OBJECTIVE STATISTICAL IMPROVEMENT WITH COMBINED ARRAYS

In the previous chapter, a review of the literature found a gap, namely the use of multi-objective statistical improvement methods for Type I robust design problems where a surrogate model of the response is regressed on both design and noise variables. This led to a research objective, re-printed here for convenience:

> **Research Objective:** Implement multi-objective statistical improvement methods using surrogate models that are functions of both design and noise variables (combined arrays).

Ultimately, the real question of interest to a designer is whether such a method has any merit. First, however, the method must be implemented.

The method will require quantification of both epistemic and aleatory uncertainty, followed by the selection of sampling criteria both in design space and in noise space. Methods will be drawn from the literature where possible, and several candidate approaches will be presented at each stage.

Research questions will be raised with respect to the effectiveness and desirability of the proposed method with respect to established methods, which will lead to a set of experiments to be discussed in later chapters.

6.1 Second-Order Probability: Epistemic of Aleatory Statistics

If Multi-Objective Expected Improvement methods are to be used, it is necessary to quantify the *epistemic uncertainty* in the objective metrics. Since the objective metrics here are measures of *aleatory uncertainty* (expected performance and some measure of risk), this

means finding *epistemic uncertainty* in *aleatory* measures. This epistemic-of-aleatory uncertainty is known as *second-order probability* or SOP [29]. Notation for the various terms is shown in Table 3.

In expected improvement methods, Bayesian surrogates (typically Gaussian Process models) are fitted directly to the objective metrics; these surrogates provide Gaussian uncertainty distributions for the value of the objective at un-sampled points. Keane [60] used this method for Type I robust design, by running a fixed number of Monte Carlo samples at candidate designs, finding the aleatory statistics, and fitting separate Gaussian Process models to mean and standard deviation.

If similar methods are to be attempted with surrogates that model performance in a *combined* design/noise space, the aleatory mean and risk must be calculated indirectly from the combined surrogate. For a particular design of interest, the aleatory input noise distributions must be propagated through the surrogates to find the aleatory mean and risk.

This chapter is organized as follows. First, a two-dimensional illustrative example is shown, to clarify the concept of second-order probability in context. In the next two sections, it will be shown how Monte-Carlo based methods can be used to find SOP terms, both for linear Bayesian models and for Kriging-type Gaussian Process models. The Gaussian Process-based method will be shown to have precedence in the literature. Additionally, analytical methods for calculating SOP terms are found in the literature for a certain limiting assumptions. Finally, procedures will be proposed for sampling the simulation code. An outer search on the design variables will be based on existing multi-objective statistical improvement methods, and an inner search will attempt to efficiently estimate the statistics of interest.

6.1.1 A Two-Dimensional Illustrative Example

The next sections will illustrate the computation of epistemic uncertainty in aleatory statistics. In order to better explain the procedure, a purely illustrative example problem will be used, with a *single* design variable and a *single* noise variable. Though in practice such a simple problem could be solved with simpler methods, it will be used in the interest of

Table 3: Notation for Second Order Probability Terms

$p_e(\mu_a)$	Epistemic distribution on aleatory mean
$p_e(\rho_a)$	Epistemic distribution on a generic aleatory risk statistic
$\mu_e(\mu_a)$	Epistemic mean of the aleatory mean
$\sigma_e(\mu_a)$	Epistemic standard deviation on the aleatory mean
$\mu_e(\sigma_a)$	Epistemic mean on aleatory standard deviation; note that aleatory standard deviation *cannot* have a Gaussian distribution.
$\sigma_e(\sigma_a)$	Epistemic standard deviation on the aleatory standard deviation
$Cov[\mu_a, \sigma_a]$	Epistemic covariance between the aleatory mean and standard deviation

visualization.

A plot of the example function can be seen in Figure 17. The performance metric, T, is a function of a single design variable, D, and a single noise variable, S. Assume that the aleatory uncertainty distribution of the noise variable, $p(S)$, is known to the designer. In the example, the aleatory distribution of the noise variable is known to be Gaussian.

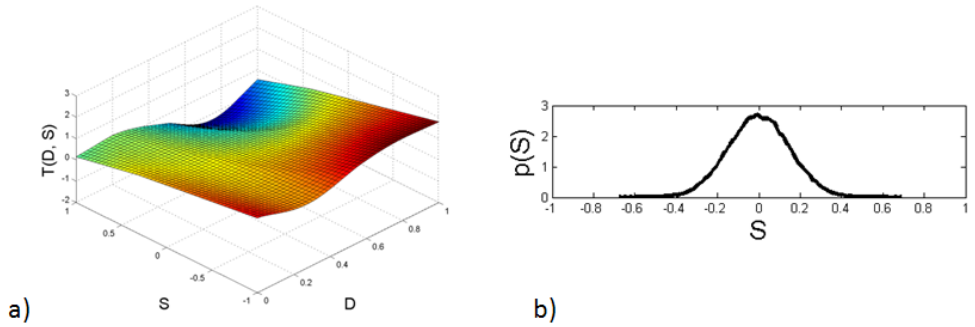


Figure 17: (a) A plot of a 2-dimensional example problem, with a single design variable D and a single noise variable S (b) The assumed aleatory distribution $p(S)$

Say there is a particular design of interest, called D^*. If the true function $T(D, S)$ is known (though it may be expensive to sample), a very good approximation of the true aleatory output distribution $p(T(D^*, S))$ can be found by exhaustively sampling from $p(S)$ and repeatedly evaluating $T(D^*, S)$. This concept is shown in Figure 18.

From this exhaustive Monte-Carlo sampling, the true aleatory statistics of mean and standard deviation can be found with high accuracy. Note that for this example problem, *standard deviation* will be assumed as the measure of risk. These two statistics at the chosen

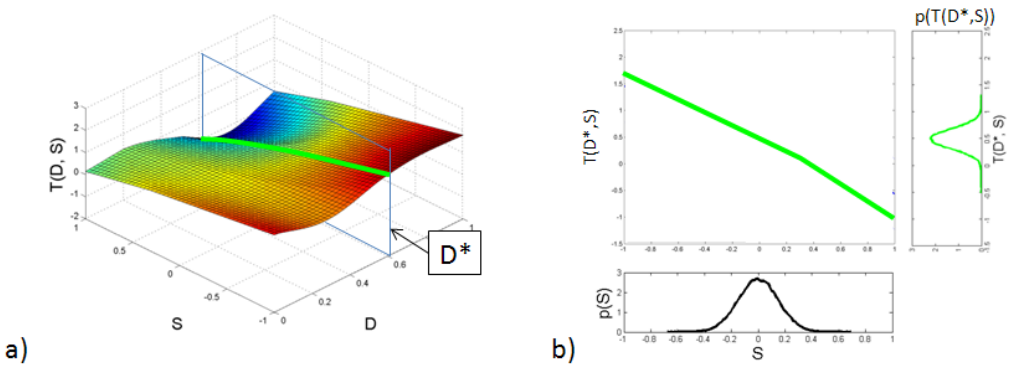


Figure 18: (a) A slice of $T(D, S)$ taken at a particular design, D^*. (b) Finding the aleatory output distribution $p(T(D^*, S))$ for a known Gaussian noise variable distribution $p(S)$.

design point are denoted $\mu_{true}(D^*)$ and $\sigma_{true}(D^*)$, respectively.

In practice, the true function $T(D, S)$ might be expensive, and this sort of exhaustive Monte Carlo sampling would be impractical. In the next two sections, it will be assumed that a Bayesian surrogate has been fitted to $T(D, S)$, and the statistics $\mu(D^*)$ and $\sigma(D^*)$ will be *estimated*, along with a measure of their *epistemic uncertainty*.

6.1.2 Linear Bayesian Models

This section will describe a method for estimating the *epistemic uncertainty*, in a Bayesian sense, on *aleatory statistics* using a Linear Bayesian Surrogate fit to both design and noise variables. This method does not explicitly appear in the literature, though it will be shown in the next section that a nearly identical method can be found for Gaussian Process models. Linear models are less frequently found in the literature in general, so it is not surprising that the method does not appear, obvious extension though it is. The case of linear Bayesian models is presented first because it is conceptually simpler.

Say that a 10-point design of experiments has been selected to choose points in a combined (D, S) space. A linear Bayesian model has been fit to the data, and is shown in Figure 19. In this example, the basis functions include linear terms as well as Gaussian radial basis functions centered on each data point.

Recall from the literature review section on Linear Bayesian Models that in a *non-Bayesian* linear model, a set of weights w are found that best describe the data, and are

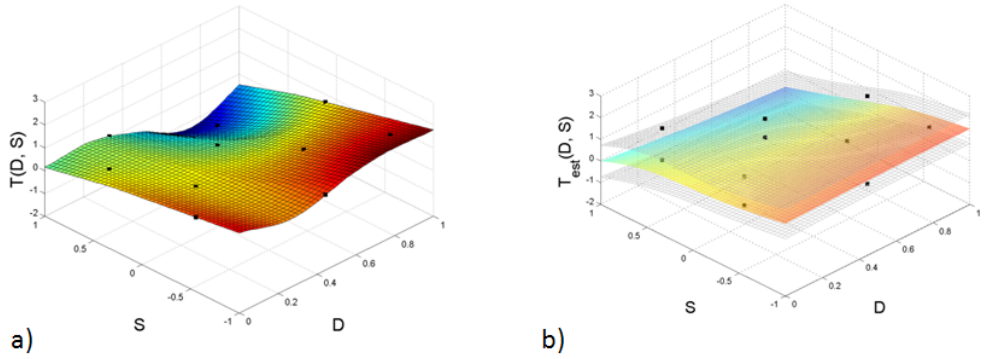


Figure 19: (a) A 10-point Latin Hypercube DoE in (D, S) space (b) A Bayesian linear model fit to the data

multiplied by a set of *basis functions*, evaluated at un-sampled points. From Equation (7):

$$\hat{T}(X) = \mathbf{w}^T \phi(X)$$

In a *Bayesian* linear model, the epistemic uncertainty is encoded in posterior distributions on the linear weights, Equation (9). This distribution is a multivariate Gaussian. Some posterior probability distributions $p(\mathbf{w}|\mathbf{D})$ given the data $\mathbf{D}$ are shown for some of the weights in the example problem in Figure 20. Note that marginals are shown; all of the weights are correlated. From now on, the joint posterior of the weights will simply be written as $p(\mathbf{w})$, and the conditional on the data $\mathbf{D}$ will be dropped from the notation. Say

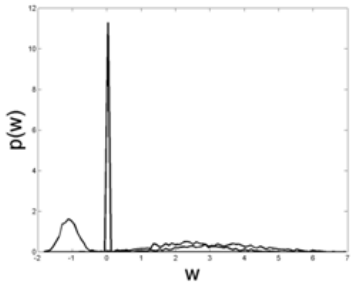


Figure 20: Some of the distributions in $p(\mathbf{w}|\mathbf{D})$. Note that these are actually multivariate Gaussian, marginals are shown.

the designer wishes to know the aleatory statistics at a particular design, D^*. Like in the previous section, a slice can be taken. Unlike before, where the true function was known, now there is epistemic uncertainty. A slice along with a 95% epistemic confidence interval,

derived from the Bayesian model, is shown in Figure 21(a). The confidence interval is found by evaluating the posterior predictive distribution of the Bayesian model, originally given in (12).

$$p(\hat{T}|X, \mathbf{T}, \mathbf{S}_0, \beta) = \mathcal{N}(\hat{T}|, \mu_{\hat{T}}, \sigma^2_{\hat{T}}(X))$$
$$\mu_{\hat{T}}(X) = \mathbf{m}_N^T \phi(X)$$
$$\sigma^2_{\hat{T}}(X) = \frac{1}{\beta} + \phi(X)^T \mathbf{S}_N \phi(X)$$

Where now $X = [D^T, S^T]^T$ is a single vector with both design and noise variables. The prediction will from here on be written as $\hat{T}(D, S)$, and all conditionals will be dropped.

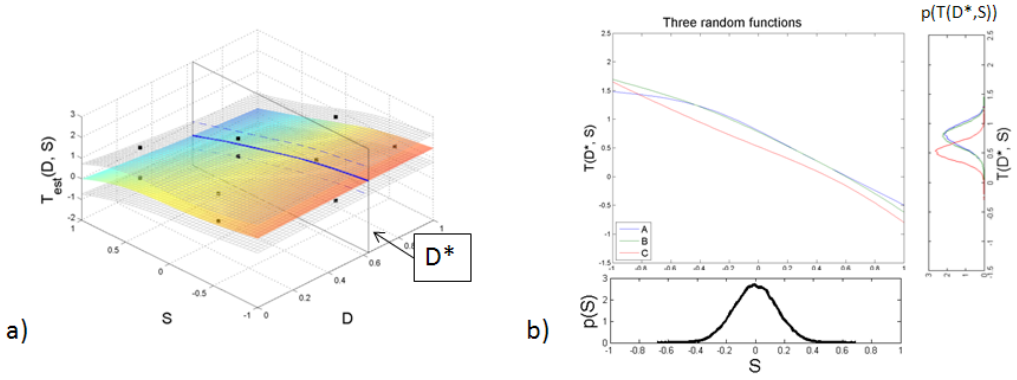


Figure 21: (a) A slice of the surrogate showing $T(D^*, S)$ at a fixed D^* (b) Three randomly generated functions $\eta_{(i)}(D, S)$ shown over the same slice

If the designer wished to find a best estimate of the aleatory statistics at a design D^*, they could do so in a manner similar to that used with the true function: with Monte Carlo sampling on S, and by using the function defined by the mean predictive value of $\hat{T}(D^*, S)$. However, the designer in this case does not simply want a best estimate; they want epistemic uncertainty.

Epistemic uncertainty is encoded in the multivariate $\mathbf{w}$ distribution $p(\mathbf{w})$, and this can be used to the designer's advantage. Say the designer samples from that multivariate distribution. Every draw i produces a vector of weights $\mathbf{w}_{(i)}$; every vector of weights represents a single random linear model, $\eta_{(i)}(D, S)$. Three such draws and the slices $\eta_{(i)}(D^*, S)$ are shown in Figure 21(b).

For each draw from $p(\mathbf{w})$, Monte Carlo sampling on the noise variable S can be used

to find an aleatory distribution on the output. From this, the aleatory statistics $\mu_{a,(i)}$ and $\rho_{a,(i)}$ may be computed *for that random function*. When this is repeated over many samples of $\mathbf{w}$, a histogram may be found for each of the aleatory statistics, and these histograms are approximations of the *epistemic distributions* of the aleatory statistics, $p(\mu_a)$ and $p(\rho_a)$. Figure 22 shows 100 random draws from $p(\mathbf{w})$ and the corresponding histograms for $\mu(D^*)$ and $\sigma(D^*)$.

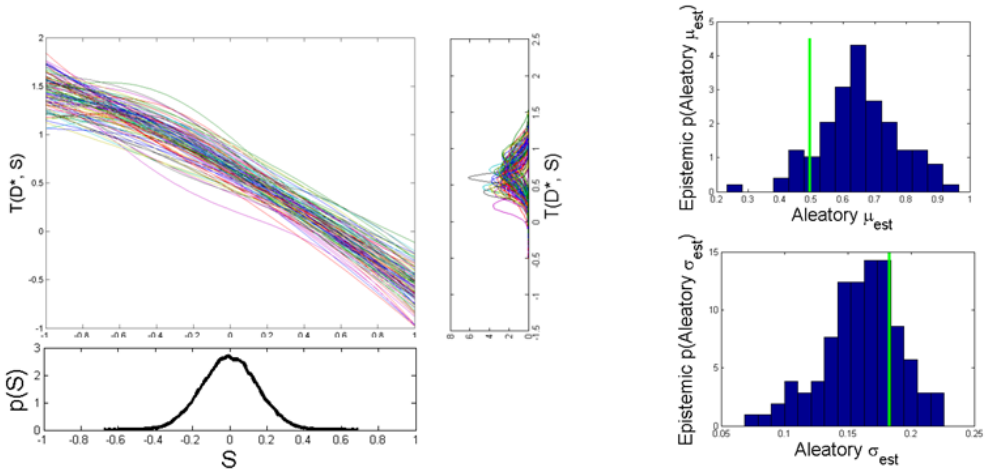


Figure 22: (a) 100 randomly generated functions shown at D^* (b) Epistemic histograms for the aleatory statistics $\mu(D^*)$ and $\sigma(D^*)$

6.1.3 Gaussian Process Models

When the surrogate is a Gaussian Process model rather than a linear model, second-order statistics can be found using a similar method, with some notable differences. This method appears in a slightly modified form in Oakley and O'Hagan [87]. The modification, along with further work by Apley *et al.* [4] is discussed at the end of the subsection.

As before, Figure 23 shows the same 10-point DoE and a Gaussian Process model fit to the same data. At the same D^* as in the previous example, the designer wishes to know the aleatory statistics, given a known noise variable distribution $p(S)$.

Unlike in a linear Bayesian model, in a Gaussian Process model the epistemic uncertainty is *not* represented through function weights. Instead, it is encoded in a correlation structure. Recall from the literature section on Gaussian Process models, that the covariance between

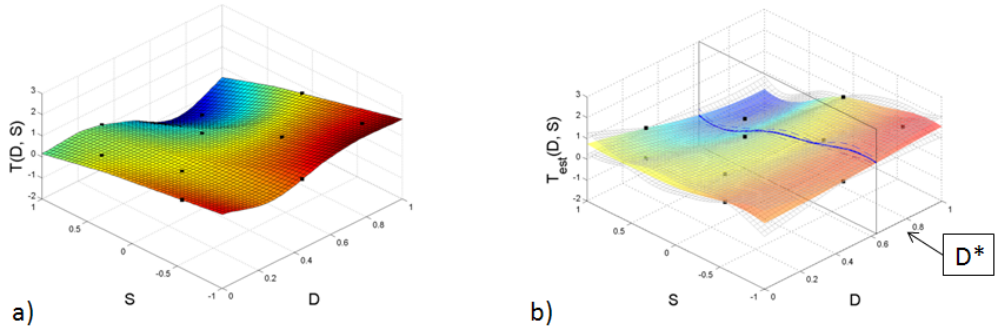


Figure 23: (a) A 10-point Latin Hypercube DoE in (D, S) space (b) A Gaussian Process Kriging model fit to the data

any two un-sampled points i and j can be found as 30:

$$\begin{aligned}\text{Cov}[\hat{T}(x^{(i)}), \hat{T}(x^{(j)})] = \hat{\sigma^2}[k(x^{(i)}, x^{(j)}) - \psi(x^{(i)})^T \Psi^{-1} \psi(x^{(j)}) \\ + \{\phi(x^{(i)}) - G^T \psi(x^{(i)})\}^T W \{\phi(x^{(j)}) - G^T \psi(x^{(j)})\}]\end{aligned} \tag{36}$$

This can be easily expressed in matrix form, to find the covariance between a set of points $\mathbf{x}$:

$$\Sigma_{\hat{T}}[\mathbf{x}] = \hat{\sigma^2}[\mathbf{\Psi}(\mathbf{X}) - \psi(\mathbf{x})^T \Psi^{-1} \psi(\mathbf{x}) + \{\phi(\mathbf{x}) - G^T \psi(\mathbf{x})\}^T W \{\phi(\mathbf{x}) - G^T \psi(\mathbf{x})\}] \tag{37}$$

Like with a linear model, what is needed is a way to sample from function space, that is, to make random Monte Carlo draws, each of which represents a possible functional form. This can be easily achieved by choosing a set of points, and sampling from their joint predictive distribution. Figure 24 shows a joint predictive distribution conceptually, for two points $X^{(A)}$ and $X^{(B)}$ in a 1-dimensional example problem.

The joint posterior is a multivariate Gaussian distribution:

$$\hat{\mathbf{T}}(\mathbf{x}) \sim \mathcal{N}(\hat{T} | \mu_{\hat{T}}, \Sigma_{\hat{T}}) \tag{38}$$

This can be used to now sample from *function space*. Rather than actually creating a random function, as was done with the linear model, instead the points in X space are selected first, and their joint posterior predictive distribution is found. Every draw from this multivariate Gaussian represents a random function, evaluated at those points. In Figure 25,

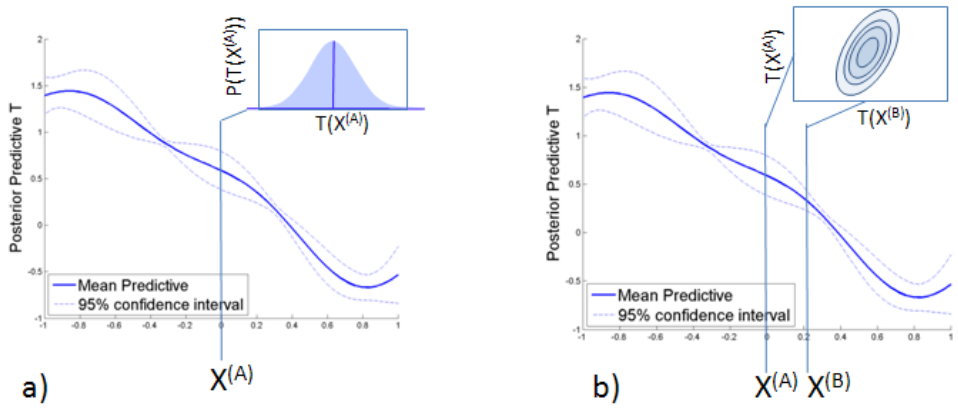


Figure 24: A 1-dimensional Gaussian Process Kriging model (a) The posterior predictive distribution at a single point (b) A joint posterior predictive distribution at two points $X^{(A)}$ and $X^{(B)}$

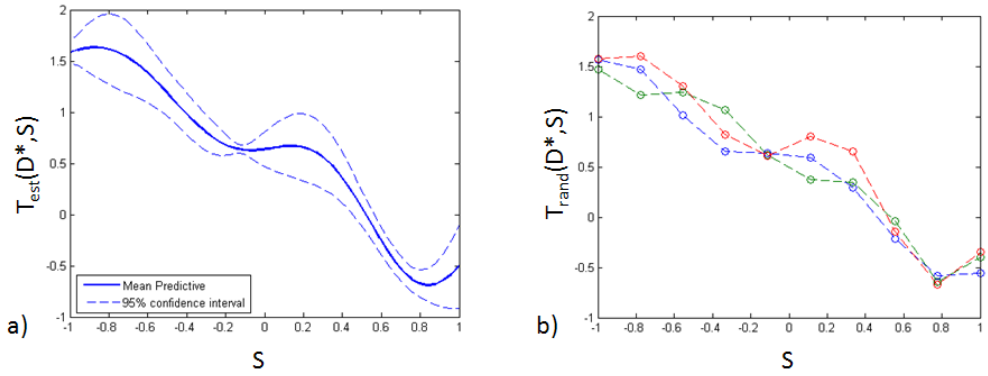


Figure 25: (a) A slice of the Gaussian Process surrogate showing $\hat{T}(D^*, S)$ at a fixed D^* (b) Three randomly generated "functions" shown over the same slice, evaluated on a set of 10 evenly-spaced points

this is shown for a slice of the 2-D example problem at D^*, with the joint posterior of ten evenly-spaced points in S evaluated three times.

To find statistics of statistics, then, the first step is now to skip ahead and generate a full set of *aleatory* Monte Carlo cases $\mathbf{S}_{MC}$, by drawing from the aleatory noise variable distribution $p(S)$, shown in the lower part of Figure 26(a). In the example, 1,000 noise cases were generated. These Monte Carlo cases are all points in noise space, and it is now possible to find their epistemic *joint posterior predictive* distribution, $\hat{\mathbf{T}}(D^*, \mathbf{S}_{MC})$.

Now a *second* set of Monte Carlo cases is created, this time by drawing from that distribution. Every random draw represents a different function, evaluated at all of the noise

points. This is shown in Figure 26(a), with 100 random functions drawn from $\hat{\mathbf{T}}(D^*, \mathbf{S}_{MC})$.

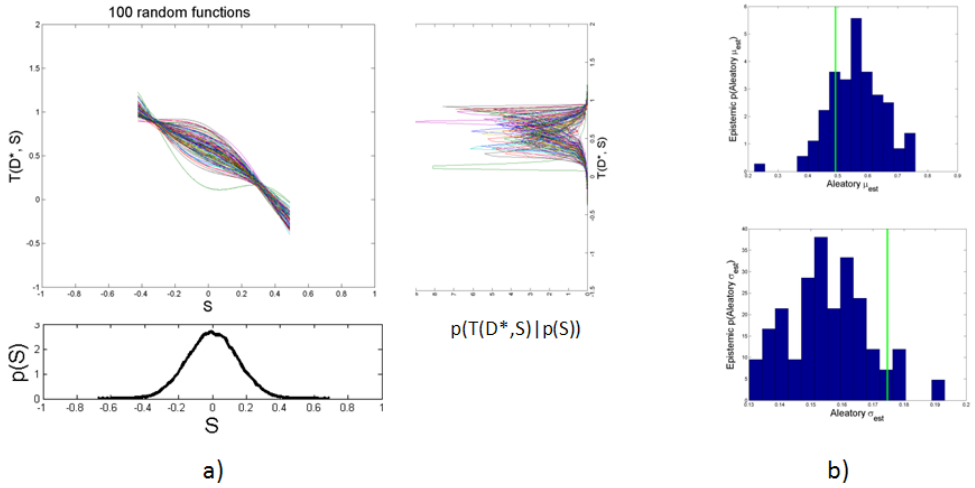


Figure 26: (a) 100 randomly generated functions shown over the same slice, evaluated on a set of 1,000 Monte Carlo points drawn from the aleatory noise distribution $p(S)$ (b) Epistemic histograms for the aleatory statistics $\mu(D^*)$ and $\sigma(D^*)$

Now the designer possesses the same information as was required in the case of a linear Bayesian model. For every random function, there is an aleatory distribution $p(\hat{T}(D^*, S)|p(S))$. The aleatory statistics μ_a and σ_a can be computed for every random function, resulting in histograms that approximate the *epistemic* distributions $p_e(\mu_a)$ and $p_e(\sigma_a)$. The results of this are shown in Figure 26(b).

6.1.3.1 Non-Gaussian Epistemic Distributions

Recall that the purpose of finding epistemic uncertainty in the aleatory statistics was to enable the use of *multi-objective statistical improvement* methods, where each aleatory statistic is treated as an objective. Previous use of multi-objective statistical improvement methods, as found in the literature, relied on the use of two Bayesian models, one each for μ_a and σ_a. In such an implementation, the epistemic uncertainty distributions $p_e(\mu_a)$ and $p_e(\sigma_a)$ will be Gaussian, because that is the form that the surrogates take.

However, in the previous section, it was shown how to indirectly approximate $p_e(\mu_a)$ and $p_e(\sigma_a)$, both derived from a single surrogate model. When finding the epistemic distribution

of an aleatory statistic, there is no longer a structural guarantee that the uncertainty distributions will be Gaussian. In the case of σ_a, for example, it cannot be Gaussian, because standard deviation is always greater than or equal to zero. Whether a given statistic is in fact Gaussian, or failing that whether it can be approximated as Gaussian, will affect the usability of off-the-shelf statistical improvement methods.

6.1.3.2 Extensions to the Method

Several extensions to this method can be found in the literature. Oakley and O'Hagan use a similar method to find statistics in a non-design scenario. They use the same concept for generating random functions, but they find that with large numbers of aleatory Monte Carlo points, the samples can become close together and lead to ill-conditioning of the correlation matrix. They solve this problem by first carefully selecting points from the space (though they do not describe their criteria). Random draws from their joint posterior are made, each corresponding to a random "function", and from this function an arbitrarily large number of points can be sampled, which will have a joint t-distribution; however, the authors note that this process is very computationally intensive. Note that Oakley and O'Hagan use a generalized Gaussian Process model with linear model terms.

Building on the work of Oakley and O'Hagan, Apley *et al.* extend the method to an engineering design application, where estimates of aleatory mean and standard deviation are combined into a single uncertain robustness objective, namely $\mu + 3\sigma$. Importantly, the authors find an analytical expression for $p_e(\mu_a)$ and $p_e(\sigma_a)$ for the specific case of Gaussian distributions on the noise variables $p(S)$. The epistemic distribution on the aleatory mean $p_e(\mu_e)$ is shown to be Gaussian. The epistemic distribution of the aleatory standard deviation $p_e(\sigma_a)$ is approximated as a Gaussian (σ is selected because it is closer to a Gaussian than σ^2), and $\mu_e(\sigma_a)$ and $\sigma_e(\sigma_a)$ are approximated. Lastly, $\mathrm{Cov}(\mu_a, \sigma_a)$ is also found analytically. Note that Apley *et al.* also use a general Gaussian Process model [4]. The emphasis of the Apley *et al.* paper is on the computation of an additional measure:

$$f(D) = \mu(D) + c \cdot \sigma(D) \tag{39}$$

For a normally distributed response, this will be equivalent to a percentile, or a Value-at-Risk. Here, c is selected for a percentile of interest; for a VaR of 95%, it will be approximately 1.64. For this reason, this metric will be referred to as a pseudo-VaR, and will be used in place of VaR in test exercises. Apley *et al.* develop expressions for the second-order probabilities with respect to this $f(D)$ metric:

$$\mu_f(D) = \mu_\mu(D) + c \cdot \mu_\sigma(D) \tag{40}$$

$$\sigma_f^2(D) = \sigma_\mu^2(D) + c^2 \cdot \sigma_\sigma^2(D) + 2c \cdot \mathrm{Cov}[\mu(D), \sigma(D)] \tag{41}$$

Apley *et al.* further argue that Monte Carlo sampling will be computationally impractical, the work of Oakley and O'Hagan notwithstanding, and encourage the use of analytic results instead. However, the authors do not provide the complete integrated results, and leave a good deal of integration as an exercise to the reader.

In an un-published technical note, O'Hagan presents similar results using clarified notation, and provides complete equations for computing the statistics [88]. O'Hagan does not provide results for $\mathrm{Cov}(\mu_a, \sigma_a)$, but this can be easily found by combining O'Hagan's expressions with those found in Apley *et al.*. The complete expressions take several pages to write out and are provided in Appendix A, along with more details on Apley's paper. Note that so far the emphasis of this chapter has been on μ_a and σ_a as aleatory statistics of interest. Other statistics, namely VaR and CVaR, might be of interest to the decision-maker. Oakley [85] presents a method similar to the method presented in Oakley and O'Hagan [87], that can be used to estimate aleatory percentiles (which are equivalent to VaR). The method is combined with a method for sampling the noise variables, which will be presented in a later section.

6.2 Sampling in Design Space: C-MOSI

The previous section showed several methods by which second-order probability (SOP) could be calculated for any design of interest. If the assumption can be made that both the mean and risk measure are Gaussian random variables, any of the five multi-objective statistical improvement methods discussed in the previous chapter can be used. Apley *et*

al. made this assumption, and their application was engineering design, but they did not specifically use a statistical improvement method.

The choice of which MOSI method to use is somewhat arbitrary. The five options are Emmerich *et al.* [30], Keane [60], Bautista [15], Knowles [65], or Hawe and Sykulski [48]. The method proposed by Keane is limited to only two dimensions without extensive re-deriving, so it can not be used if there is more than one stochastic objective; for only two objectives, it is appealing because it can be evaluated in closed form. Emmerich's method can also be found in closed-form for a two-dimensional problem, but it presents computational difficulties in the presence of more than two dimensions. For larger-dimensional problems, this leaves the methods proposed by Bautista, Knowles, or Hawe and Sykulski, any of which might be implemented. Hawe and Sykulski's paper does not provide algorithmic details, so some creativity would be required to implement it. Bautista's method can be readily calculated for an arbitrary number of objectives using Monte Carlo methods.

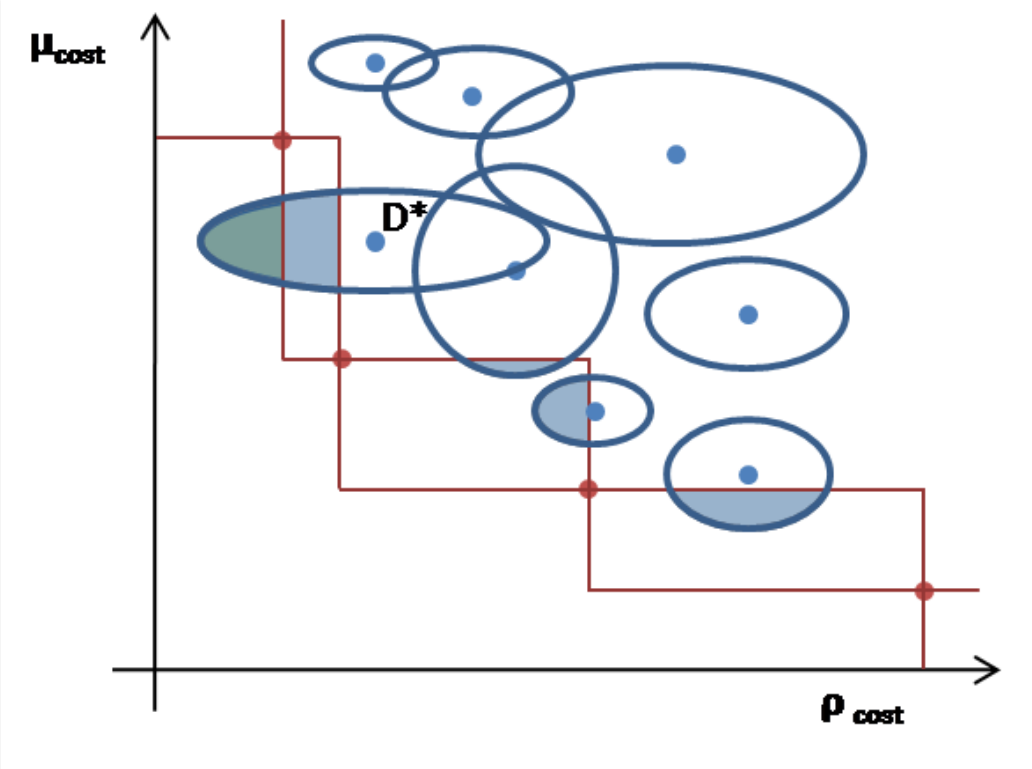


Figure 27: A notional Pareto frontier. Several candidate designs are shown, with epistemic uncertainty ellipses drawn around them. The selected design, D^*, seems to have the maximum expected Pareto improvement (or the highest probability of Pareto improvement)

According to one of these algorithms, a design point can be selected that maximizes either the probability of improvement or expected improvement, in terms of aleatory mean and risk. This process is performed with a global optimizer that searches over the design variables and performs the SOP calculation at many design points. A notional frontier plot is shown in Figure 27. The P(I) or E[I] function is usually highly multi-modal, with local maxima between already-sampled points, where the epistemic uncertainty is high.

Previously, this document has used the term Multi-Objective Statistical Improvement (MOSI) to refer generally to the five methods found in the literature. When MOSI is used in a combined design/noise array, it will be called Combined-space MOSI, or C-MOSI.

6.3 Modifying MOSI for Uncertain Pareto Sets

In existing multi-objective statistical improvement methods, it is assumed that already-sampled points are known with certainty, and the current known Pareto frontier is found from these points, as in Figure 28(a). However, in a combined-array approach, since designs are only partially sampled in noise space, there are no designs where the objective values are known for certain. It is not possible, therefore, to talk about improvement over a deterministic Pareto frontier. Instead, it is necessary to talk about improvement over a *probabilistic frontier*, as in Figure 28(b). Looking at the figure, it would be expected that ignoring uncertainty in the Pareto set would lead to an under-estimation of uncertainty, especially in regions where the frontier itself is highly uncertain. This problem was addressed in the context of a multi-objective genetic algorithm by Kumar [67].

A revised MOSI approach will be discussed only in the context of Emmerich, Deutz, and Klinkenberg's hypervolume method [30], though a similar approach could be used to modify Keane's normalized method [60].

Emmerich, Deutz, and Klinkenberg [30] propose a hypervolume-based approach to multi-objective statistical improvement, as has already been mentioned in the literature review. It divides the objective space into discrete hypervolumes, as shown in Figure 30, a reproduction of a figure found in the original paper and included for clarity. For a 2-objective problem, with independent Gaussian uncertainty on the objectives at un-sampled points,

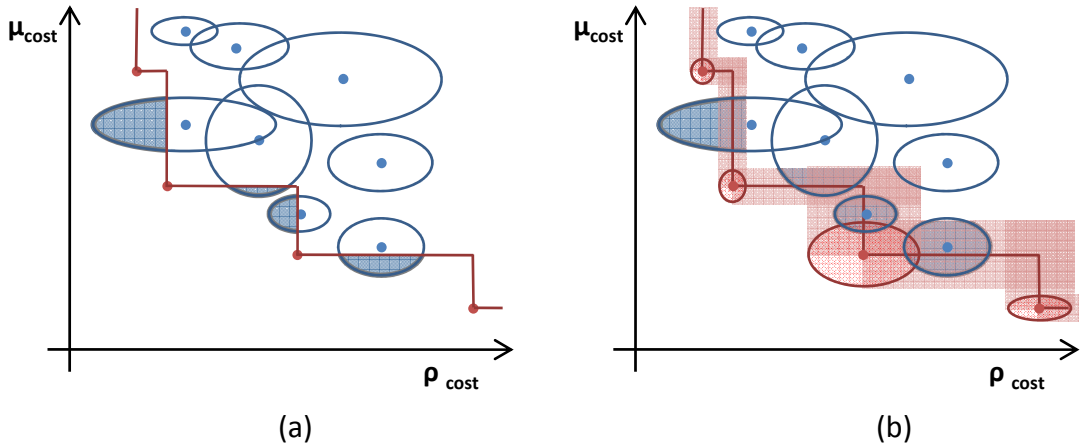


Figure 28: Multi-Objective Statistical Improvement environment, deterministic vs. probabilistic Pareto set. Blue ellipses represent uncertainty in mean/risk objective space of candidate designs. In (a), red points are deterministically known samples that form the currently understood Pareto frontier, and red lines delineate the augmenting vs. dominated regions. In (b), the current Pareto set is known only probabilistically, as in a combined-array method. The transparent red bars represent confidence regions for the Pareto frontier boundaries.

the computations can be made analytic and quite tractable. The authors even provide MATLAB code on their website [31].

Note that in a combined-space approach, two assumptions of the analytic approach are violated. The objectives at un-sampled points are not *independent*, nor are they necessarily *Gaussian*. However, it will be assumed for tractability that the objectives are both independent and Gaussian.

Since there are no already-sampled points which are known with certainty, the entire design space can theoretically be considered as part of the probabilistic frontier. If the epistemic uncertainty distributions have infinite tails, any arbitrary point has some probability of being on the frontier. If any un-sampled point is to improve over the Pareto frontier, and the Pareto frontier theoretically extends over the entire design space, then the problem becomes one of integration over the entire design space.

However, instead of treating the entire space as a frontier, a subset of points will be considered as candidates for the Pareto set. This subset could be chosen randomly from throughout the design space. Instead, it will consist of all existing design samples, since

already sampled designs will be expected to have lower uncertainty than randomly sampled designs. From this set of designs, a Pareto set will be selected based on the *expected values* of their objectives. An important assumption will be made here. Though the values of the current Pareto set will be considered uncertain, the *membership* and *ordering* will remain fixed. Thus, several important assumptions and simplifications have been made so far:

- **Assumption:** The epistemic uncertainty in the design objectives is both independent between designs and Gaussian.
- **Simplification:** The Pareto set will be chosen from the current set of (albeit incompletely) sampled designs.
- **Assumption:** Even though the objective values of the Pareto set are uncertain, the membership and order in the set will be assumed to remain fixed. Membership in the Pareto set will be determined by the expected values of the mean/risk objectives.

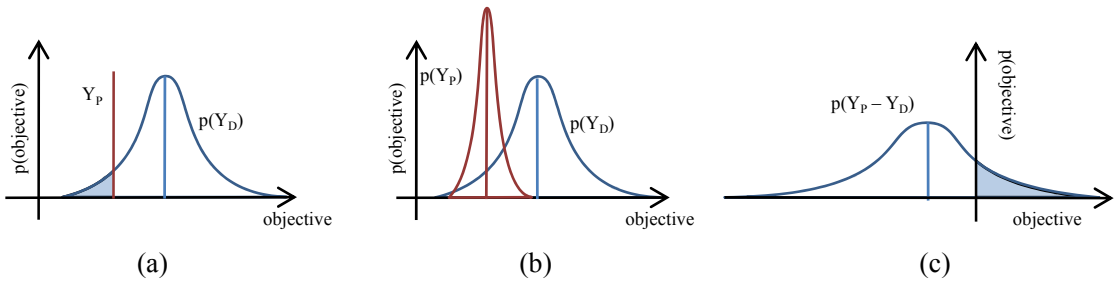


Figure 29: 1-dimensional statistical improvement of design D over a Pareto point P. In (a), P is known deterministically, and the improvement is found from the positive tail of $Y_P - p(Y_D)$. In (b), P is uncertain, and so its Expected Improvement or Probability of Improvement is found from the new distribution $p(Y_P - Y_D)$, which is shown by itself in (c).

Under these assumptions, it is possible to consider a modified version of Emmerich, Deutz, and Klinkenberg's MOSI method that applies to an uncertain Pareto set. Consider first the improvement over a fixed, totally certain baseline, as depicted in Figure 29(a), and

as assumed in the paper. With Gaussian uncertainty on un-sampled designs, the difference in objective space between a design D and a particular Pareto point P is Gaussian:

$$Y_P - Y_D \sim \mathcal{N}(Y_P - \mathrm{E}[Y_D], \mathrm{Var}[Y_D]) \tag{42}$$

Now consider the improvement in one objective over an uncertain baseline, as shown in Figure 29(b). If the values Y_D and Y_P of the un-sampled point (D) and the baseline (P) are considered to be jointly a bivariate Gaussian, their difference is also Gaussian:

$$Y_P - Y_D \sim \mathcal{N}(\mathrm{E}[Y_P] - \mathrm{E}[Y_D], \mathrm{Var}[Y_P] + \mathrm{Var}[Y_D] - 2\mathrm{Cov}[Y_P, Y_D]) \tag{43}$$

If the parameters $\mathrm{E}[Y_D]$, $\mathrm{E}[Y_P]$, $\mathrm{Var}[Y_D]$, $\mathrm{Var}[Y_P]$, and $\mathrm{Cov}[Y_D, Y_P]$ are all known, then the Expected Improvement can be easily found analytically. Monte Carlo methods can be used to find these parameters, though if Monte Carlo methods are used then the Expected Improvement can be found more directly.

Note, however, that the term $\mathrm{Cov}[Y_D, Y_P]$ represents the covariance in a single objective between the candidate design and an uncertain Pareto point. If analytical methods are used to find SOP terms, this term will not be available. Is this actually a problem? For designs that are close together in design space, covariance in the objective will be positive. So if the term is ignored, the variance of the improvement metric will tend to be over-estimated for designs close to already sampled Pareto points. This will lead to an over-estimation of the Expected Improvement and "over-sampling" in these areas.

Consider, however, that as a sampling criteria, Expected Improvement is not necessarily optimal, especially for a combined-array approach. Ideally, in addition to finding superior designs, a sampling algorithm should also improve the estimate of the current frontier. From a purely heuristic standpoint, ignoring the covariance term will lead to increased sampling near designs already thought to be on the frontier. Therefore, though somewhat unsatisfying from a theoretical standpoint, it is not certain that ignoring the $\mathrm{Cov}[Y_D, Y_P]$ term will result in degraded algorithmic performance, and there is a possible reason why it might improve performance. Due to the computational expense of testing this, the question will be left unresolved, and the covariance term will be dropped for tractability.

Assumption: $\mathrm{Cov}[Y_D, Y_P] = 0$ (the covariance in objective values between any two designs is zero)

With the assumption of independence between designs, Emmerich, Deutz, and Klinkenberg's analytical MOSI method can be easily revised through selective modification of variances. The new assumed form for the difference in an objective between two designs becomes:

$$Y_P - Y_D \sim \mathcal{N}(\mathrm{E}[Y_P] - \mathrm{E}[Y_D], \mathrm{Var}[Y_P] + \mathrm{Var}[Y_D]) \tag{44}$$

6.3.1 Emmerich's Hypervolume E[I] Method Summary

What follows is a brief description of Emmerich *et al.*'s equations. The description largely follows that paper's flow verbatim, with very minor changes to notation. Two figures from the paper are also reprinted verbatim in Figure 30 and Figure 31. Readers who are interested in a derivation may consult the very useful original paper, [30]. After the method is described, the next section will detail the changes required to adapt to uncertain Pareto sets.

It is assumed that there are K designs in the Pareto set:

$$P = \{\vec{y}^{(1)}, \ldots, \vec{y}^{(K)}\} \tag{45}$$

Each point is in M-dimensional objective space, $\mathbb{R}^M$, and has coordinates:

$$\vec{y}^{(k)} = \{b_1^{(k)}, \ldots, b_m^{(k)}, \ldots, b_M^{(k)}\} \tag{46}$$

Where m will be used to index over the dimensions of the objective space. Now, looking at a single objective m, all the Pareto points can be sorted by their mth coordinate, denoted b_m. The sorted list is written as $b_m^{(1)}, b_m^{(2)}, \ldots, b_m^{(j)}, \ldots, b_m^{(K)}$, where j is used to index over the ordered list. Note that this index does *not* always refer to the same design, since the ordering will be different for every objective. For technical reasons, the authors define $b_m^{(0)} = -\infty$, $b_m^{(K+1)} = y_m^{ref}$, and $b_m^{(K+2)} = \infty$. These sets of sorted coordinates lead to a partitioning of the objective space into grid cells. A set of grid coordinates is denoted $(i_1, \ldots, i_m, \ldots, i_M)$,

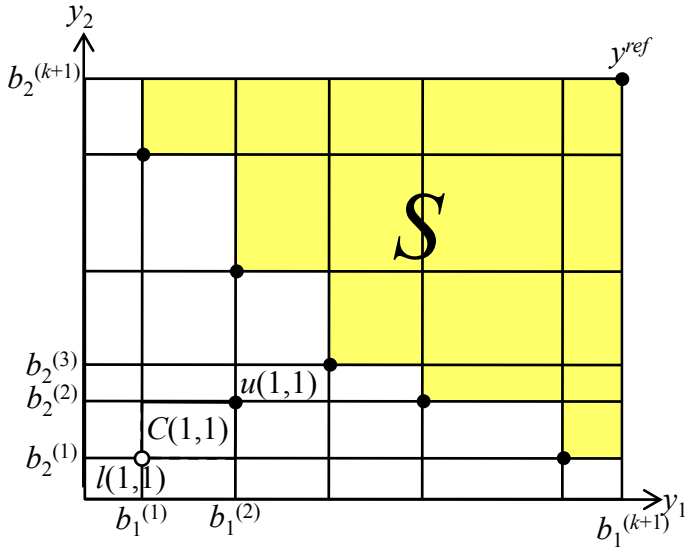


Figure 30: Reproduction of Figure 2 from Emmerich *et al.*. "Schematic drawing of a population, its hypervolume, and grid in the bi-objective case. The black points are the points of the population, except the point in the upper right corner that marks the position of the reference point for the hypervolume. The yellow region defines the measured hypervolume S. The grid coordinates are indicated by $b_1^{(i)}$ and $b_2^{(i)}$ for the first and second coordinate, respectively. Grid-cell $C(1,1)$ is highlighted by a thick black boundary."[30]

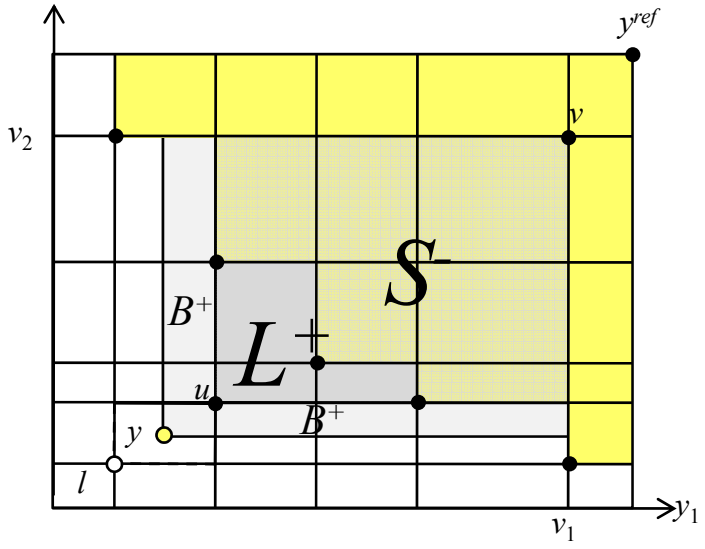


Figure 31: Reproduction of Figure 3 from Emmerich *et al.*. "Schematic drawing of the integration area and grid in the bi-objective case."[30]. The figure is reproduced here primarily as an explanation of the vector $\vec{v}$ and the region S^-.

where each $i_m \in \{0, \ldots, K+1\}$. A grid cell is denoted $C(i_1, \ldots, i_M)$, and is determined by a lower grid node and an upper grid node, as a half-open (from below) interval box. These two nodes are defined as the upper and lower bounds of C:

$$\vec{l}(i_1, \ldots, i_m, \ldots, i_M) = (b_1^{(i_1)}, \ldots, b_m^{(i_m)}, \ldots, b_M^{(i_M)}) \tag{47}$$

$$\vec{u}(i_1, \ldots, i_m, \ldots, i_M) = (b_1^{(i_1+1)}, \ldots, b_m^{(i_m+1)}, \ldots, b_M^{(i_M+1)}) \tag{48}$$

The space bounded by these two nodes will be described with the notation $(\vec{l}, \vec{u}]$. Many of these cells lie behind the Pareto frontier, and are called *inactive cells*. Those that dominate the Pareto set are called *active cells*, denoted C^+. The expected improvement of a design D with objective vector $\vec{y}$ over the Pareto set P is the sum of the integrals over each active cell,

$$\mathrm{E}[I](D) = \sum_{C(i_1,\ldots,i_M) \in C^+} \delta(i_1, \ldots, i_M) \tag{49}$$

$$\delta(i_1, \ldots, i_M) = \int_{\vec{y} \in (\vec{l}, \vec{u}]} I(\vec{y}, P) \cdot \mathrm{PDF}(\vec{y}) d\vec{y} \tag{50}$$

Emmerich *et al.* provide the following expression for the computation of the integral over an active grid cell:

$$\delta(i_1, \ldots, i_M) = \left(\prod_{j=1}^{M} \delta_j(i_1, \ldots, i_M) \right) - \mathrm{Vol}(S^-) \prod_{i=1}^{M} \left(\Phi\left(\frac{u_i - \mu_i}{\sigma_i} \right) - \Phi\left(\frac{l_i - \mu_i}{\sigma_i} \right) \right) \tag{51}$$

$$\begin{aligned} \delta_j(i_1, \ldots, i_M) =& \Psi(v_j(i_1, \ldots, i_M), u_j(i_1, \ldots, i_M), \mu_j, \sigma_j) \\ & - \Psi(v_j(i_1, \ldots, i_M), l_j(i_1, \ldots, i_M), \mu_j, \sigma_j) \end{aligned} \tag{52}$$

Where the vector $\vec{v}_j(i_1, \ldots, i_M) \in \mathbb{R}^M$ is defined as shown in Figure 31. The terms Ψ are the integrations of the marginal normal distribution:

$$\Psi(a, b, \mu, \sigma) = \sigma \cdot \phi\left(\frac{b - \mu}{\sigma} \right) + (a - \mu)\Phi\left(\frac{b - \mu}{\sigma} \right) \tag{53}$$

And ϕ and Φ are simply the PDF and CDF of the standard normal distribution. Finally, $\mathrm{Vol}(S^-)$ is a correction term for a certain hypervolume defined by a subset of P and $\vec{v}$, as shown in Figure 31.

6.3.2 Changes to Emmerich's Hypervolume E[I] Method to Deal with Uncertain Pareto Sets

The procedure can be easily modified to deal with an uncertain Pareto set. As previously stated, it is assumed that the membership and ordering of the set does not change, and covariance between designs is ignored. The only modification required, then, is to adjust the variance of the objectives to account for the uncertainty of the Pareto points. Any time a standard deviation in a dimension, σ_j, appears, it is simply replaced by a corrected standard deviation, which will include the uncertainty contributed by a particular Pareto point. The previous equation for $\delta(i_1, \ldots, i_M)$ becomes:

$$\begin{aligned}\delta(i_1, \ldots, i_M) = &\left(\prod_{j=1}^{M} \delta_j(i_1, \ldots, i_M) \right) \\ &- \mathrm{Vol}(S^-) \prod_{i=1}^{M} \left(\Phi\left(\frac{u_i - \mu_i}{\tilde{\sigma}_j^u(i_1, \ldots, i_M)} \right) - \Phi\left(\frac{l_i - \mu_i}{\tilde{\sigma}_j^l(i_1, \ldots, i_M)} \right) \right)\end{aligned} \tag{54}$$

$$\begin{aligned}\delta_j(i_1, \ldots, i_M) = &\Psi(v_j(i_1, \ldots, i_M), u_j(i_1, \ldots, i_M), \mu_j, \tilde{\sigma}_j^u(i_1, \ldots, i_M)) \\ &- \Psi(v_j(i_1, \ldots, i_M), l_j(i_1, \ldots, i_M), \mu_j, \tilde{\sigma}_j^l)\end{aligned} \tag{55}$$

Where each $\tilde{\sigma}_j$ is influenced by the variance in the j-dimension of *one* particular Pareto point:

$$\tilde{\sigma}_j^l(i_1, \ldots, i_M) = \sqrt{\sigma_j^2 + \mathrm{Var}[y_j^{(i_j)}]} \tag{56}$$

$$\tilde{\sigma}_j^u(i_1, \ldots, i_M) = \sqrt{\sigma_j^2 + \mathrm{Var}[y_j^{(i_j+1)}]} \tag{57}$$

Where σ_j^2 is the variance in objective j due to the uncertainty of the candidate design, and $\mathrm{Var}[y_j^{(i_j)}]$ is due to the Pareto point that marks the lower bound of box C in dimension j. Similarly, $\mathrm{Var}[y_j^{(i_j+1)}]$ is from the Pareto point that marks the upper bound of box C in dimension j. The mixing of σ^2 and Var[] notation is regrettable.

Computationally, this is a minor extension of Emmerich *et al.*'s method, and adds only a small amount of expense. The effect on a small example problem can be seen in Figure 32. The figure depicts a small Pareto set of 5 designs. In (a), the contours show the multi-objective E[I] using Emmerich's method, for a candidate point with a fixed variance of 0.01 in both objectives, as a function of that point's expected value. In (b), the middle Pareto

point has been made uncertain, also with a variance of 0.01, and the surface shows the *increase* in E[I] due to this change, when the modified method is used. Candidate points which lie near this uncertain Pareto point will experience an increase in E[I]. What's more, a local boost in E[I] extends outward from the uncertain point, along its gridlines.

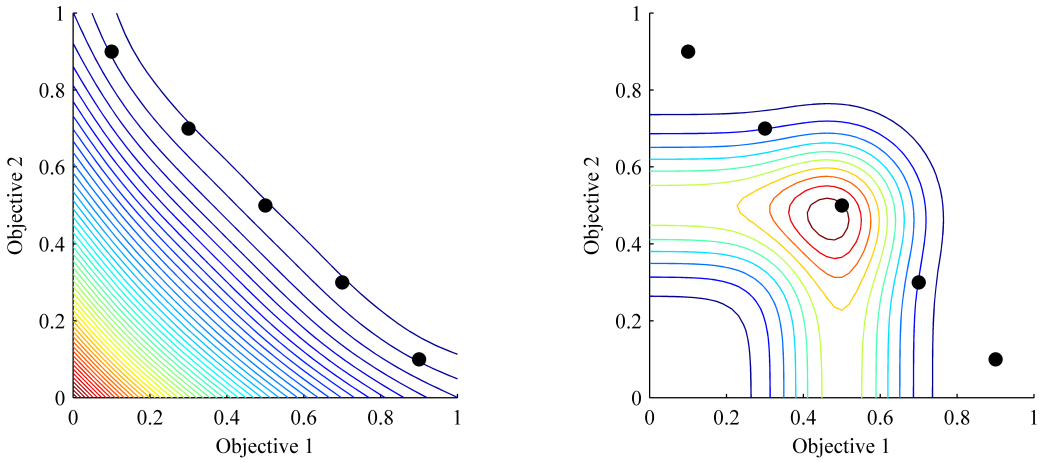


Figure 32: The effects of Pareto set uncertainty on MOSI. Both objectives are to be minimized. In (a), five deterministic Pareto points are shown, and the hypervolume-based E[I] of a candidate design centered on the corresponding point on the graph is shown, when the design has a variance 0.01 in each objective. There is some expected improvement just behind the frontier, and it eventually begins to increase linearly as the expected objective value becomes very dominant. In (b), the middle Pareto point has been given a variance of 0.01 in each objective, and the *increase* in E[I] is shown relative to (a). There is a local boost in E[I] near the uncertain Pareto point.

6.4 Sampling in Noise Space

After a design has been chosen which maximizes P(I) or E[I], a new decision must be made before the expensive simulation code can be sampled: where the next sample point should be placed in noise space. Whereas the design point was chosen to balance exploration of un-sampled regions with exploitation of areas known to be attractive, the noise variable settings can be chosen based purely on an exploration metric.

A naive sampling criteria would be to simply sample where the epistemic posterior variance of the response model is highest, as shown in Figure 33. Eventually, this will reduce the uncertainty to zero. However, this is probably not the most efficient method. Depending

on the aleatory distribution of the noise variable, there are probably areas that are more important than others. For example, if the noise variable has a Gaussian distribution centered on the noise range with a tight distribution, points at the edges of the noise space will be highly unlikely, and so the model accuracy at the edges won't strongly affect how well the aleatory mean and variance are estimated. It could also be imagined that a designer is interested in some tail-centric risk measure, like value-at-risk, in which case the accuracy at the tail of interest would disproportionately affect the accuracy of the risk metric.

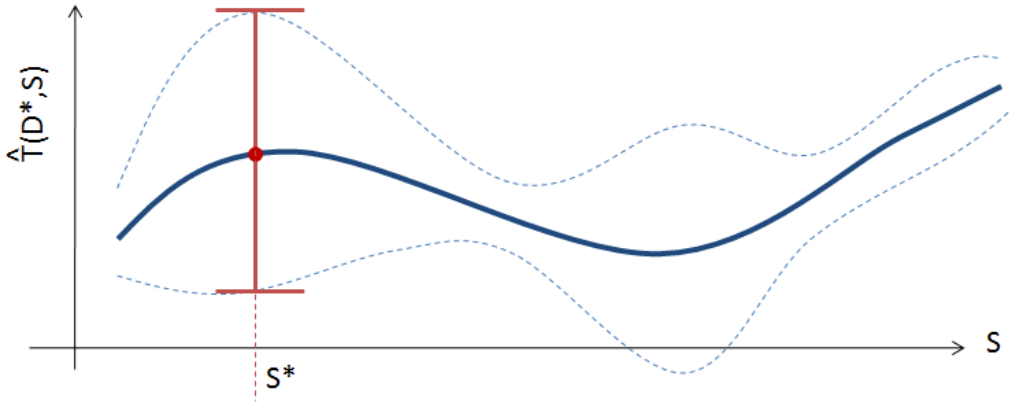


Figure 33: A simple noise sampling scheme, point of highest uncertainty

6.4.1 Oakley and O'Hagan's General Sampling Method

The problem of choosing samples in order to estimate the distribution of an output is well-studied in statistics. The specific case found here, where the response is already estimated by a Bayesian surrogate, can be found in several papers by by Oakley for Gaussian Process surrogates. For a general case, where the objective is to obtain more information about the output probability distribution, Oakley and O'Hagan employ a "simple greedy algorithm" similar to the "naive" algorithm suggested above, and shown in Figure 34. The set of Monte Carlo points (from the SOP-finding step) are used as candidate points; over this set, the point of maximum posterior variance is selected. If additional points are to be selected, the first point is added to the data set and the procedure is repeated [87]. For an aleatory noise distribution that spans the input space and an infinite number of Monte Carlo samples, this is the same as the naive strategy suggested above; but if the Monte Carlo set is smaller, the

strategy will favor points with higher aleatory density.

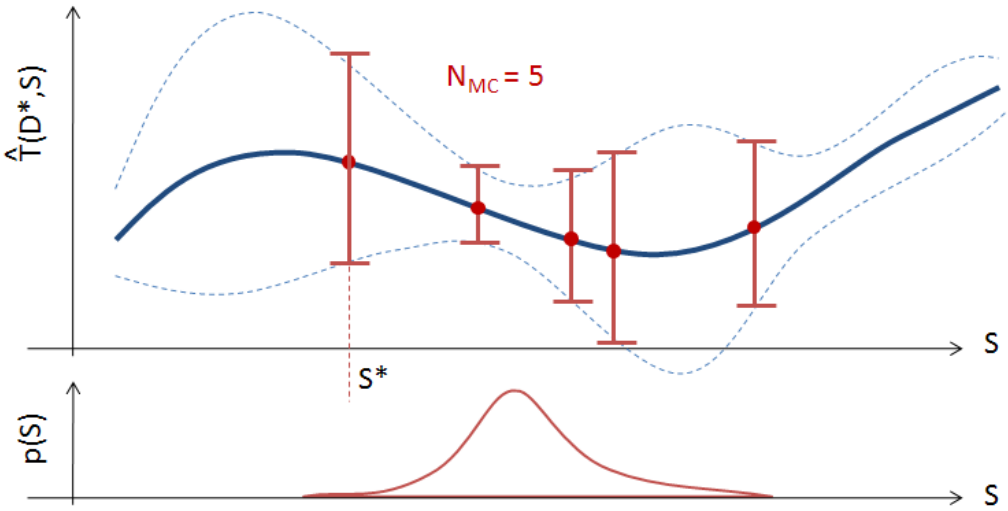


Figure 34: Oakley and O'Hagan's general method. From the already-existing Monte Carlo sample (from calculating SOP), select the point of highest variance.

6.4.2 Oakley's Method for Percentiles

In a later paper, also for Gaussian Process model surrogates, Oakley deals with the case where the objective is to estimate a percentile (which is the same as value-at-risk) [85]. The method is shown graphically in Figure 35. His objective is to identify a region **R** in which to concentrate the next set of samples. First, he generates a random function $\eta_{(i)}(\cdot)$, just as was done during SOP quantification. He generates a series of J Monte Carlo samples $\{s_1^*, s_2^*, ..., s_J^*\}$ by sampling from the aleatory noise distribution $p(S)$, and using the random function he creates a set of outputs. From this output set, he find the single point $\nu_{(i)}$ that is the best estimate of the percentile of interest. He repeats the procedure K times to create a set of "random" percentile values, $\nu = \{\nu_{(1)}, \nu_{(2)}, ..., \nu_{(K)}\}$. These percentile values are all candidate locations of the true percentile in S space, and should be clustered in a suitable region R where the true percentile is likely to be found. However, that does not mean that the points $\nu_{(i)}$ themselves are good sample locations, because there may already be a real sampled data point nearby. The task then is to find good sample locations that collectively reduce the uncertainty in the region **R**. To accomplish this, Oakley uses a maximum entropy Latin Hypercube approach. He first finds a weighting function $w(S)$

that approximates the density of ν over R. Details are not given on how, but the possible multi-region nature of the problem is discussed, as is illustrated in Figure 35. From this weighting function, a Latin Hypercube sample is generated, equal in size to the number of desired samples. This Latin Hypercube design will cover the space R with a density that approximates the density of ν, but if it has points close to existing samples it will not reduce uncertainty much. A way of quantifying the degree to which it reduces uncertainty is to find the *entropy*, which will be denoted $\mathbb{S}$. For a set of data Q with a covariance matrix Ψ_Q, entropy is proportional to the determinant:

$$\mathbb{S} \propto |\Psi_Q| \tag{58}$$

The covariance matrix Ψ_Q comes from the joint posterior of the Gaussian Process model. A large number of random Latin Hypercube designs are created and tested, and the design with the largest entropy is chosen as the next set of sample points. The method assumes multiple samples are taken in each step. If a single sample point is desired the method would need to be modified slightly because a Latin Hypercube sample could not be generated. A simple method would be to pick a probability contour from $w(S)$ and choose the point within that contour of maximum uncertainty.

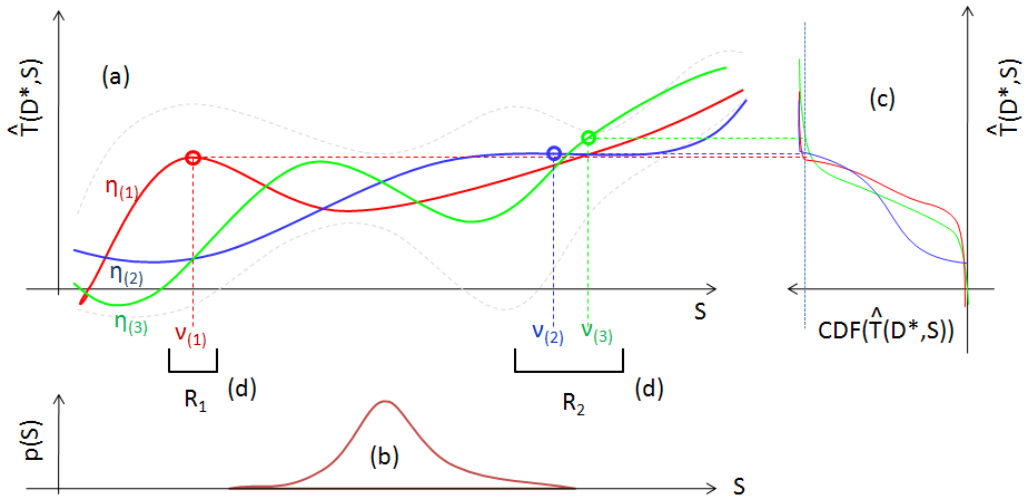


Figure 35: Oakley's method for sampling to improve percentile estimates [85]. (a) Generate random functions $\eta_{(i)}$ (b) Densely sample from $p(S)$ (c) Estimate the percentile $\nu_{(i)}$ for each random function (d) Define the new sampling region $\mathbf{R}$, which may be discontinuous.

The Oakley method for percentiles should work well, but it will only reduce the uncertainty of the percentile estimate. It will not improve the estimate of the mean or variance very well, because the samples will all be clustered around one percentile. This leads to a general sampling strategy, outlined next.

6.4.3 A General Noise Sampling Strategy

For the sake of discussion, assume for a moment that there are exactly two global objectives, to reduce the mean μ_a and to reduce risk ρ_a. These objectives are the basis on which the current design point D^* was chosen. For the problem of sampling in noise space, then, it is assumed that there are two sub-objectives: reduce the epistemic uncertainty in each of the aleatory statistics.

A multi-objective problem framework could be adopted, to work towards both sub-objectives simultaneously. However, unlike in the case of the global objectives, there is not a need to ultimately select a single point. Multiple points can be selected sequentially, so there is no need to trade between the two objectives. A sensible strategy, then, is to adopt two independent sampling methods, one for each objective.

So the problem can be stated as two sequential optimization challenges:

$$\arg\min_{S^*} \text{Var}_e(\mu_a|S^*) \tag{59}$$

$$\arg\min_{S^*} \text{Var}_e(\rho_a|S^*) \tag{60}$$

As before, μ_a and ρ_a represent the aleatory mean and risk statistics, and Var_e represents the epistemic variance. S^* is a candidate sample in noise space.

Under this strategy, an ideal sampling method can be found independently for each of the statistics of interest. For value-at-risk, for example, Oakley's method above [85] can be used. For mean and variance, a different method is required.

6.4.3.1 Proposed Noise Sampling Method: I-SOP

In a later paper, Oakley uses techniques similar to his previous papers to find the Expected Value of Sample Information (EVSI) [86]. However, the application and form of the problem is different from this application. Inspired by that work and by Oakley and O'Hagan [87], a

method is proposed for selecting noise samples in order to improve estimates of the aleatory response statistics. It largely mirrors the procedure used to select the design sample point D^*, and it involves a similar level of computational effort. The approach is to sample at points that reduce best-guess epistemic uncertainty in the aleatory statistics. It relies on imputation of the candidate data point to estimate the SOP terms; for that reason, and to have a convenient name for it, it will be called I-SOP.

Recall that the problem has been decomposed into sub-objectives. Assume for a moment that the current sub-objective is to improve the estimate of the aleatory mean.

D^* has been fixed, and the task is to select a sample point (or points) in S space. The next points will be selected one at a time, through an optimization strategy. Starting by improving the estimate of the aleatory mean, a sensible objective would be:

$$\arg\min_{S^*} E[\ \mathrm{Var}_e(\mu_a|S^*)] \tag{61}$$

Note that here the objective is the expected value of the objective from (59), given a new sample S^*. The expression $E[\ \mathrm{Var}_e(\mu_a|S^*)]$ is the expected epistemic uncertainty in the aleatory mean estimate given that S^* will be sampled. This is an expectation on a second-order probability; it could be called a third-order probability. It would be possible to again estimate it numerically, by doing a three-level nested Monte Carlo, but this is really not necessary. Instead, the strategy employed will be to *impute* an imaginary data point at S^*: the response at that point is assumed equal to the epistemic mean prediction given by the response surrogate. The surrogate is partially updated; the tuning parameters are not re-optimized, but S^* is added to the data set and the covariance matrix is re-computed. With the data point imputed, the surrogate will be referred to as $\hat{t}(D^*, S|S^*)$.

The posterior prediction of $\hat{t}(D^*, S|S^*)$ can be used to generate random functions, and these can be used to find SOP terms just as before. Now the epistemic statistic of interest is simply the variance in the aleatory mean, which has been calculated assuming that S^* is set to its predicted mean value. So the optimization problem has been changed to:

$$\arg\min_{S^*} \mathrm{Var}_e(\mu_a|\hat{T}(S^*) = \mu_{\hat{T}}(S^*)) \tag{62}$$

Where here the outer expectation has been removed, and it it is assumed that $\hat{T}(S^*)$ is set to its mean predictive value from the Bayesian surrogate, $\mu_{\hat{T}}(S^*)$. This is the *most likely* value of the function, and S^* is said to be imputed. This simplified formulation is not equivalent to the real expectation equation shown in (61), but it is used as a best guess, a common practice in statistics.

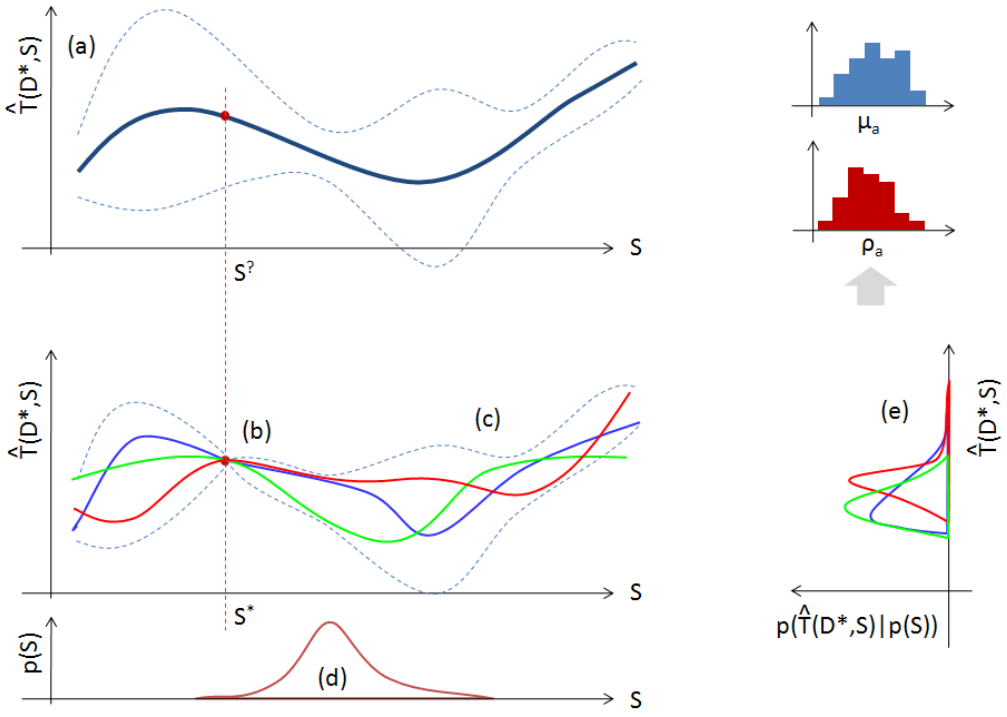


Figure 36: I-SOP method for finding "most likely" epistemic uncertainty, given a candidate sample S^*. (a) Bayesian surrogate, given the data (b) Candidate point S^* is imputed (c) Generate random functions (d) Sample aleatory noise $p(S)$ (e) Calculate aleatory statistics for each random function, and find the epistemic variance in the aleatory statistic

Note that once the point S^* has been imputed, since the procedure is the same as for finding SOP terms, if an analytic formulation could be used for the SOP terms, it can be used here.

Using a single-objective optimization scheme, such as a genetic algorithm, the point in noise space that results in the lowest epistemic variance on the aleatory mean is selected. This is called S^*_μ.

The exact same procedure can be used to find the point S^*_ρ that minimizes the epistemic variance in the aleatory risk statistic. The selected S^*_μ is kept imputed (or sampled before

the next step), and the objective now is:

$$\arg\min_{S^*} \text{Var}_e(\rho_a | \hat{T}(S^*_\mu) = \mu_{\hat{T}}(S^*_\mu), \hat{T}(S^*) = \mu_{\hat{T}}(S^*)) \tag{63}$$

If the risk statistic is a percentile, this method could still be used, or alternately the method proposed by Oakley can be used [85]. The selected point is called S^*_ρ.

Note that the procedure described assumes that a Bayesian surrogate is fitted to both design and noise variables. However, there is no reason it cannot be used with a surrogate fitted only to noise variables. Indeed, the papers by Oakley that inspired the method do just that. Therefore, though it is presented here in the context of a combined space method, it can also be used for crossed array noise sampling. The only extra step required is to run an initial sparse DoE on only noise variables, to train the noise surrogate.

6.5 Pseudocode for Proposed Method

The entire method as proposed, from start to finish, is given here as pseudocode. Through most of this chapter, concepts have been presented in the context of a single stochastic objective T, but the method is easily extensible to a vector of J objectives $\{T_j\}$, and the notation below reflects that. The flowchart in Figure 37 presents the same process assuming only a single stochastic objective.

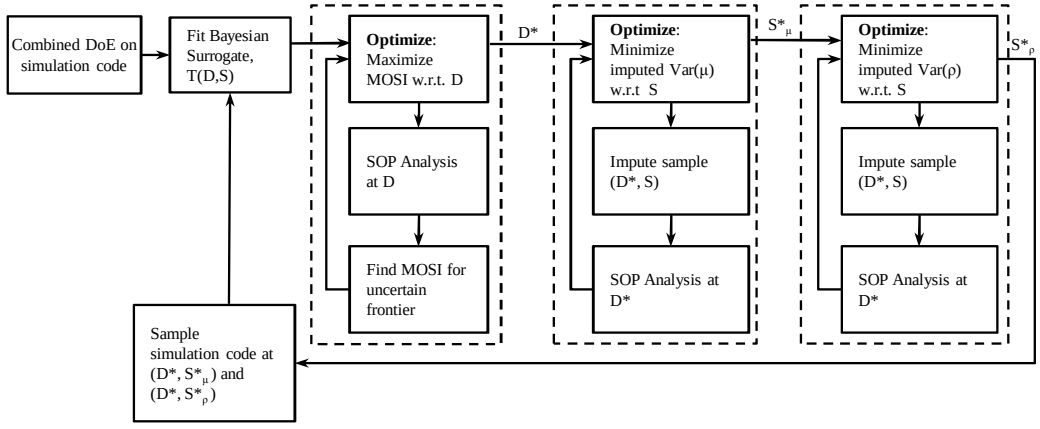


Figure 37: Flowchart of C-MOSI method. This assumes a single stochastic objective $\hat{T}$.

- Construct a sparse DoE using a space-filling design in combined design and noise space (D,S), and evaluate the expensive code at all points.

- Fit a Bayesian surrogate $\hat{T}_j(D, S)$ to all J stochastic responses of interest j.
- Loop:
  - Optimize: $\arg\max_{D^*}$MOSI
    - SOP analysis: for each objective j find epistemic distributions $p_e(\cdot)$ for aleatory mean μ_a and risk ρ_a measures
    - Feed all SOP moments $\{\mathrm{E}[\mu_e], \mathrm{Var}(\mu_e), \mathrm{E}[\rho_e], \mathrm{Var}(\rho_e)\}_j$ into a MOSI algorithm
  - Optimize: At D^*, find $2J$ points that reduce epistemic uncertainty
    - $S^*_{\mu,j}$ minimizes imputed epistemic uncertainty in the aleatory mean

$$S^*_{\mu,j} = \arg\min_{S^*} \mathrm{Var}_e(\mu_a | \hat{T}(S^*) = \mu_{\hat{T}}(S^*))$$

    - $S^*_{\rho,j}$ minimizes imputed epistemic uncertainty in the aleatory risk

$$S^*_{\rho,j} = \arg\min_{S^*} \mathrm{Var}_e(\rho_a | \hat{T}(S^*_\mu) = \mu_{\hat{T}}(S^*_\mu), \hat{T}(S^*) = \mu_{\hat{T}}(S^*))$$

  - Sample the expensive code at all $2J$ new data points $(D^*, S^*_{\mu,j})$, $(D^*, S^*_{\rho,j})$
  - Update all J surrogates $\hat{T}_j(D, S)$
  - If the sample budget has been exceeded, or the epistemic uncertainty of the Pareto set has stopped improving, stop
- Armed with inexpensive surrogates $\hat{T}_j(D, S)$, use a normal multi-objective optimizer to find the best estimate of the multi-objective mean/risk Pareto frontier.

Now that a method has been proposed, it can be compared to other methods. To a designer interested in using such a method, one of the principal questions is whether the method will be able to find the Pareto frontier with greater efficiency than other methods. As with any set of methods, the answer will be highly dependent on the problem and the implementation specifics. However, the first question is whether it is *ever* better:

Research Question 3: Is there a design scenario where a combined array Multi-Objective Statistical Improvement method out-performs both crossed-array and design of experiments methods in terms of efficiency?

Again, as with research questions 1 and 2, efficiency will need to be defined.

6.6 A Note on SOP Computational Cost

The analytic SOP calculations provided by O'Hagan [88] and proposed for use in both the C-MOSI and I-SOP steps are more computationally efficient than running nested Monte Carlo. However, that is not to say that they are inexpensive. In O'Hagan's expressions, there is one particular term, buried in the expression for the Variance of the Variance, that dominates the computational expense of the SOP calculations. It is the term $A^{-1}R_{tt}$, found in the term which O'Hagan calls I_3. A^{-1}, the inverse of the GP covariance matrix, is of size $n \times n$ (where n is the number of samples). R_{tt} is also of size $n \times n$, so the whole operation takes a number of floating point operations (FLOPs) given by:

$$\text{FLOPs} = n \cdot n(2 \cdot n - 1) \approx O(n^3) \tag{64}$$

This is on the same order as taking the inverse of A. The SOP step itself is buried in three separate optimization loops: the C-MOSI step to select D^* and the two I-SOP steps to select S^*_μ and S^*_ρ. In each of these three optimization loops, the SOP terms must be calculated many times. Therefore, for every iteration where a new design or noise point must be selected through optimization, the operation scales as:

$$\text{FLOPs} \propto N_{\text{opt}} \cdot O(n^3) \tag{65}$$

where N_{opt} is just the number of optimization function calls to the SOP analysis. The number of function calls required to optimize D^* will itself depend on the dimensionality of the design space, and the number to select the S^* points will depend on the dimensionality of the noise space. As the number of samples reaches the order of hundreds, this one term comes to completely dominate the calculation time. In testing, by the time the sample size

reaches about a thousand, selecting a single pair of samples took on the order of an hour on an Intel i7 Sandy Bridge.

CHAPTER VII

ELECTRIC POWER GENERATION TEST PROBLEM

Since this work is motivated by an electric power portfolio selection problem, an electric power simulation tool will be needed for testing. The tool will be used in two ways. It will be used to demonstrate that the method can be used to solve a problem similar to the motivating problem. Before that, however, it will be used to characterize the design space. In the next chapter, a generic and scalable test problem will be developed whose characteristics are similar to that of the simulation test problem.

7.1 Power Portfolio Components

The test problem need not be capable of simulating every possible electric power generation scenario. However, it should be capable of modeling a subset of scenarios such that the most important problem characteristics are captured. Characteristics of electric power generating portfolios may include:

- **Demand** for electric power exhibits regular daily and annual fluctuations that are partially predictable, and short-term fluctuations that are of lower magnitude and also lower predictability.
- **Baseload plants** have high capital costs but low operating costs, and usually cannot be throttled up and down very quickly. They are therefore mostly run at fixed or slowly varying power outputs. Coal and nuclear are the most common baseload plants.
- **Peaking plants** have low capital costs but higher operating costs, and are designed to be throttled up and down quickly to meet demand fluctuations. These are most commonly natural gas plants. Some amount of **spinning reserve** must be kept online to deal with demand fluctuations, that is, more plants must be kept running than are currently needed, so they can throttle up to meet spikes in demand. This reduces efficiency.

- Most baseload and peaking plants are powered by **fossil fuels** such as coal and natural gas, which are subject to price uncertainty and potentially carbon costs.

- Peaking plants and most baseload plants are also **dispatchable**, in that they can be turned on and off at will (though perhaps not instantaneously).

- **Renewable energy** sources such as wind and solar power are *not* dispatchable, that is their power outputs cannot be adjusted to meet varying demand. They are subject to fluctuations that may be partially (though never fully) predicted. Their power is random, and different sources will not be very correlated, so **diversification** reduces uncertainty. Non-dispatchable supply uncertainty increases required fossil spinning reserve.

- **Energy storage** systems such as pumped hydroelectric storage and flow batteries are net consumers of energy, but can quickly absorb and release energy to smooth out fluctuations in demand and non-dispatchable energy supply. They are not common, as they are usually quite expensive, but their importance may increase in the future as renewable energy penetrations increase (and potentially as their costs come down). Storage reduces the need for peaking plants.

- **Demand-side management** allows an electric utility to have some degree of control over demand, for example by adjusting consumers' thermostats at times of peak load.

- Utilities may also subsidize or otherwise encourage **efficiency measures**, to reduce the load demand. This would present itself as a change in the magnitude or statistical properties of the demand, and would be subject to uncertainty in the degree of adoption and effectiveness.

- **Market purchases** may be made from other utilities to make up for power shortfalls, at prices that depend on time of day and day of week. They are negotiated much like financial options, with an up-front price for the option and an agreed-upon "strike" price at the time the power is needed [90].

Table 4: Power Portfolio Options from Selected Utilities' Integrated Resource Plans

Utility	Natural Gas	Coal	Wind	Solar	Hydroelectric	Biomass	DSM	Nuclear	Market	Storage	Other	Year	ref
Ameren Missouri	✓	✓	✓	✓			✓	✓	✓	✓		2011	[2]
Puget Sound Energy	✓		✓		✓	✓	✓		✓		✓	2011	[95]
Xcel Energy	✓	✓	✓	✓	✓	✓	✓	✓	✓			2010	[116]
Northwestern	✓	✓	✓			✓	✓					2009	[84]
PacifiCorp	✓	✓	✓	✓			✓	✓	✓	✓	✓	2011	[90]
Entergy Louisiana	✓	✓	✓	✓		✓	✓	✓	✓			2010	[35]
Avista	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	2009	[8]
Progress Energy Carolinas	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	2009	[93]
Idaho Power	✓		✓	✓			✓	✓	✓	✓	✓	2011	[53]
Florida Power and Light	✓			✓			✓		✓		✓	2010	[37]

Table 4 shows the portfolio options investigated in several utilities' Integrated Resource Plans. The amount of detail in the plans was far greater than is obvious from the table. For example, PacifiCorp considered a wide array of solar power options, including rooftop PV, hot water installations, and combined solar thermal/natural gas systems [90]. Natural gas options almost always included both combined-cycle combustion turbines (CCCT's) and simple cycle combustion turbines (SCCT's). Carbon Capture and Sequestration was considered often, both for natural gas and coal plants. The "other" category included such diverse options as geothermal power, distributed generation [53], wave energy [90], and fuel cells [93][90], though in many cases some of the portfolio options were modeled to a far smaller degree than others. Indeed, not all of these IRPs even used simulation.

7.2 Test Case Characteristics

There exist commercial electric power simulation codes, such as GenTrader [92] and Ventyx System Optimizer [111]. Ultimately, if the methods being developed here are actually to be useful in a utility planning context, they will have to be demonstrated on such codes. However, a lower-fidelity custom simulation was produced instead. The simulation ran relatively quickly, so it could be tested extensively; the execution time depended on the size of the portfolio being considered, but at the sizes considered for the test cases it ran

in a few seconds per simulation on an Intel i7 Sandy Bridge. Additionally, it was coded in MATLAB, which allowed easier integration with the C-MOSI test code. This is seen as a first test, a proof of concept to demonstrate that the method can work on a related problem; future testing on real portfolio simulation codes will be needed in the next round of testing.

The test simulation had the following features:

- **Demand** can be generated with deterministic daily and annual fluctuations plus short-term fluctuations.
- **Natural gas plants** are dispatchable, but subject to fuel price uncertainty
- **Wind farms** are not dispatchable, and individual turbines are partially (though not perfectly) correlated in their power output
- **Solar PV arrays** provide peak power in the middle of the day, somewhat before daily peak demand. Individual PV arrays are more correlated than wind turbines, but still not perfectly correlated.
- Generic **energy storage** can smooth out short-term uncertainty in both supply and demand
- **Market purchases** can be used to make up for any shortfall, but at higher cost than self-generating
- **Demand Side Management** programs can be implemented to reduce peak loads (through load management) and overall energy consumption (through efficiency measures), with some increase in baseload (due to displaced usage).

This subset is not complete. Most notably, coal plants were not modeled, because at the level of fidelity where the system was modeled, they were indistinguishable from natural gas plants on a technical level. There is also no consideration of nuclear plants, biomass, or hydroelectric power. Those options which are modeled are represented at a quite low level of fidelity. The model is probably insufficient for an actual utility portfolio planning

exercise, but it is hoped that the test case is similar enough in functional "shape" that methods which work to solve the test problem can be realistically applied to a real power portfolio problem. The model components are described in detail in the next section, along with relevant noise sensitivity variables, and the last section characterizes the "shape" of the space.

7.3 Model Description

The model is implemented in MATLAB. It discretizes a one-year period into 8760 hours. For each hour, it attempts to satisfy load demand with non-dispatchable distributed wind and solar inputs, energy storage, and dispatchable fossil supply. Any left-over unsatisfied demand is met with market purchases. The geographical extent is not explicit, and physical power flows are not modeled explicitly, nor is there any consideration to reactive power or power quality.

The tool uses randomly generated input time series data, but does not use multiple stochastic Monte Carlo runs for every portfolio. Instead, a set of Gaussian white noise time series are generated during an initialization step, and these are used for all candidate portfolios, properly transformed to have whatever statistical properties are necessary. Thus, though sensitivity assumptions may be changed, and thus the statistical properties of the input time series may change from run to run, this occurs in a smooth and continuous manner, and it is still possible to make fair comparisons between portfolios and assumptions. This approach is the same as that used by NREL's HOMER micro-power simulation tool. Indeed the underlying algorithms for generating wind speed and insolation time series were drawn from that simulation tool's very helpful documentation [70], and their original provenance is also noted below as appropriate.

7.3.1 Load Demand

Load demand is generated as a combination of a fixed baseload, a deterministic annual cosine wave, a deterministic daily cosine wave, and autocorrelated Gaussian noise. The Gaussian noise is generated with a very simple autoregressive model. First, uncorrelated standard

Gaussian white noise is generated, $\beta_{\text{demand}}(1, \ldots, i, \ldots, 8760)$. Then, given a supplied lag-1 autocorrelation value α_{demand} and variance σ^2_{demand}, the autocorrelated noise $\hat{\beta}_{\text{demand}}$ is simply generated with:

$$\hat{\beta}_{\text{demand}}(i) = \sigma_{\text{demand}} \left(\alpha_{\text{demand}} \cdot \hat{\beta}_{\text{demand}}(i-1) + \beta_{\text{demand}}(i) \sqrt{1 - \alpha^2_{\text{demand}}} \right) \tag{66}$$

This noise is added to the mean load, along with daily and annual periodic fluctuations of amplitudes P_{daily} and P_{annual}, to generate the demand P_{demand}:

$$P_{\text{demand}}(i) = P_{\text{mean load}} + P_{\text{daily}} \cdot \cos\left(\frac{2\pi(i-3)}{24}\right) + P_{\text{annual}} \cdot \cos\left(\frac{2\pi i}{8760}\right) + \hat{\beta}_{\text{demand}}(i) \tag{67}$$

No real power demand will be truly sinusoidal. However, the model is not intended to model *accurately*, merely to provide something which is grossly similar in terms of behavior.

7.3.1.1 Load Demand Sensitivity

The random Gaussian noise series used to generate the load is generated once. Any of the other parameters, including base load, annual variation, daily variation, noise variance, and autocorrelation, can potentially be manipulated as noise variables.

7.3.1.2 Test Case Assumptions

In the test cases, it was assumed that $P_{\text{mean load}} = 1000\text{MW}$, $P_{\text{annual}} = 200\text{MW}$, and $P_{\text{daily}} = 500\text{MW}$. The hourly noise was assumed to have a standard deviation $\sigma_{\text{demand}} = 50\text{MW}$ and autocorrelation $\alpha_{\text{demand}} = 0.8$. In the final demonstration case, all assumptions in units of MW were reduced, so that $P_{\text{mean load}} = 100\text{MW}$, $P_{\text{annual}} = 20\text{MW}$, $P_{\text{daily}} = 50\text{MW}$, and $\sigma_{\text{demand}} = 10\text{MW}$.

7.3.2 Demand Side Management

Demand side management is a broad term, and may include both infrastructure that is controlled by the utility (for example a device that can override thermostats) and passive programs (for example subsidized home weatherization). No attempt was made to model any of this explicitly. Instead, DSM was simply modeled in terms of aggregated effects on

demand, by adjusting $P_{\text{mean load}}$, P_{annual}, and P_{daily}. A DSM "unit" was said to consist of some pre-specified change to each of these, at some pre-specified cost.

This is not a very realistic way to model DSM. Not only does it fail to model specific changes due to particular technologies, but it uses a linear cost relationship, which is unrealistic; there will probably be "low-hanging fruit" DSM measures that will be more cost-effective, and implemented first.

7.3.2.1 Test Case Assumptions

In the tests performed in this thesis, one DSM "unit" was made to reduce mean load by half a Megawatt, and to reduce daily amplitude by one Megawatt, with no change to annual amplitude. Due to the method used to generate demand (where daily fluctuations caused both negative and positive deviations from the mean load), the reduction in daily amplitude actually caused an increase in baseload along with the decrease in peak load.

7.3.3 Wind Farms

Since there is no geographic information in the model, wind farms are treated as generically as possible, using an approach in some ways conceptually similar to that employed by the Energy Information Administration's National Energy Modeling System (NEMS), though with a simulation approach where NEMS uses a purely statistical model [33]. As an initialization step, an upper bound $N_{\text{max turb}}$ is set on the number of possible wind turbines under consideration, and a Gaussian white noise series $\beta^{(j)}_{\text{wind}}(1, \ldots, i, \ldots, 8760)$ is generated for each possible turbine $j = (1, \ldots, N_{\text{max turb}})$, plus an extra "dummy" series ($j = 0$). As before, autocorrelated standard Gaussian noise is then generated, but this time there are $N_{\text{max turb}} + 1$ series. For autocorrelation factor α_{wind}, the series for a turbine j is:

$$\hat{\beta}^{(j)}_{\text{wind}}(i) = \alpha_{\text{wind}} \cdot \hat{\beta}_{\text{wind}}(i-1)^{(j)} + \beta_{\text{wind}}(i)^{(j)} \sqrt{1 - \alpha^2_{\text{wind}}} \tag{68}$$

Then, rather than dividing the turbines into discrete farms or arranging them geographically, every pair of standard Gaussian time series $(\tilde{beta}^{(j)}_{\text{wind}}, \tilde{\beta}^{(k)}_{\text{wind}})$ is made to have the same spatial correlation δ_{wind}. To achieve this, they are all cross-correlated with the same dummy

time series, $\hat{\beta}^{(0)}_{\text{wind}}$:

$$\tilde{\beta}^{(j)}_{\text{wind}}(i) = \sqrt{\delta_{\text{wind}}} \cdot \hat{\beta}^{(0)}_{\text{wind}}(i) + \sqrt{1 - \delta_{\text{wind}}} \cdot \hat{\beta}^{(j)}_{\text{wind}}(i) \tag{69}$$

The Gaussian noise series are now autocorrelated with coefficient α_{wind} and cross-correlated with coefficient δ_{wind}. Since all pairs of turbines have identical cross-correlation, this is a non-physically realizable set-up; in reality, closer turbines should be more strongly correlated. However, the approach has the advantage of not requiring any geographical information beyond a general idea of average proximity. This constant cross-correlation is used by NEMS, and the EIA has compiled a table of average cross-correlation parameters to use for different geographical regions of the U.S. [33]. One effect that is masked by the approach is that in reality, more desirable lands will likely be built on first, resulting in diminishing marginal cost effectiveness for later turbines. This effect would be difficult to estimate without more specific wind farm planning, or at least very extensive historical data.

For a given mean wind speed $\bar{w}$, the Gaussian noise is transformed into Rayleigh-distributed wind speeds w with the inverse Rayleigh distribution and the standard Gaussian CDF:

$$w = F^{-1}_{\text{Rayleigh}}(\Phi(\tilde{\beta}_{\text{wind}}), \bar{w}) \tag{70}$$

Wind speeds are often modeled with a Weibull rather than a Rayleigh distribution if more information is available, but in the absence of real wind data a Rayleigh (Weibull with shape factor of 2.0) can be assumed. The wind speeds are then transformed through the wind turbine power curve to yield power output. All wind turbines are assumed to have identical normalized wind power curves $\Pi_{\text{wind}}(w)$, supplied by the modeler. The power curve that was used is shown in Figure 38. It is assumed that their power output is constant over the course of an hour, a not entirely realistic assumption that is nonetheless used by programs such as HOMER [70]. Transmission losses are not modeled explicitly, though they may be treated as a uniform de-rating of the wind turbines.

As a final step, for a given electric generation portfolio with a desired installed wind power capacity, an appropriate number of wind turbines N_{turb} are selected, always in the

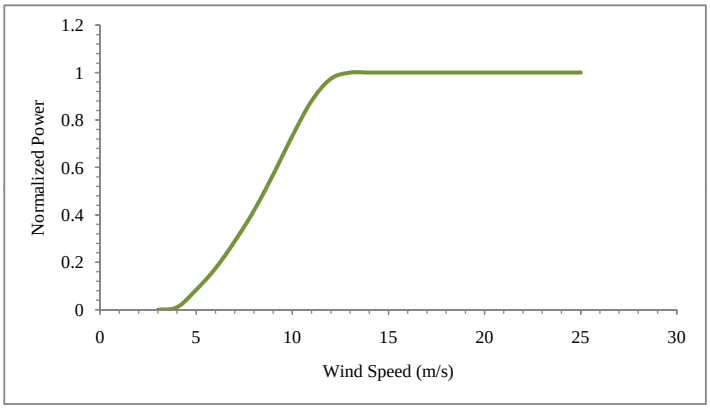


Figure 38: Wind turbine steady-state power curve.

same order. Since installed capacity treated is a continuous rather than discrete input, this will result in some "fractional" wind turbine. This last wind turbine is simply given an appropriately small rated power, and the wind power curve is scaled appropriately. Though this is not physically realistic, it allows the input and output to be smooth and continuous. The total wind power at time t, then, is just:

$$P_{\text{wind},t} = \sum_{j=1}^{N_{\text{turb}}} P_{\text{turb,rated}}^{(j)} \cdot \Pi_{\text{wind}}(w_t^{(j)}) \tag{71}$$

Where $P_{\text{turb,rated}}^{(j)}$ is the rated power of turbine j, and is the same value for all but the last "fractional" turbine. $\Pi_{\text{wind}}(\cdot)$ is the normalized power curve as a function of wind speed, and $w_t^{(j)}$ is the wind speed experienced by turbine j at time step t.

7.3.3.1 Average Wind Speed Sensitivity

Average wind speed can be adjusted as a noise sensitivity factor. Since the wind series are stored as Gaussian noise prior to being transformed into Rayleigh noise, this is a trivial matter of adjusting the transformation CDF. Thus as the average wind speed is adjusted, the simulation results will shift smoothly and continuously, without chatter.

7.3.3.2 Test Case Assumptions

In the test cases, each wind turbine was assumed to have a rated power of 2.7 MW, with the power curve given in Figure 38. The mean wind speed was assumed to be 8.0 m/s, and the spatial correlation was set to 0.5.

7.3.4 Photovoltaic Arrays

Photovoltaic arrays are treated in a manner similar to the treatment of wind farms. It is assumed that a number of PV arrays are distributed throughout a geographic area, such that they all experience cross-correlated clearness time series. Though this is a non-physical assumption, it removes the need for explicit geographic modeling.

The technical modeling is based on the work of Graham, Hollands, and Huget, who over the course of several papers in the 1980's and early 1990's developed a complete method for generating synthetic hourly solar insolation time series [46][45][50]. Credit should also be given to the Lilienthal, Gilman, and Lambert, whose documentation for the HOMER provides a procedural guide to the method [70]. As an input, the model requires monthly average clearness values $\bar{k}_t$, (atmospheric transmittance index or clearness index) which can be obtained from a database such as NREL's Typical Meteorological Year database [83].

As with wind turbines, as an initialization step, an upper limit is set on the maximum number of Photovoltaic installations, $N_{\max\ \mathrm{PV}}$. All installations are assumed to be of equal pre-determined size. Separate hourly Gaussian white noise series are generated for each possible installation, and these will be autocorrelated, cross-correlated, and used later to generate clearness time series, in basically the same procedure as was used for wind turbines.

$$\hat{\beta}^{(j)}_{\mathrm{solar}}(i) = \alpha_{\mathrm{solar}} \cdot \hat{\beta}_{\mathrm{solar}}(i-1)^{(j)} + \beta_{\mathrm{solar}}(i)^{(j)} \sqrt{1-\alpha^2_{\mathrm{solar}}} \tag{72}$$

$$\tilde{\beta}^{(j)}_{\mathrm{solar}}(i) = \sqrt{\delta_{\mathrm{solar}}} \cdot \hat{\beta}^{(0)}_{\mathrm{solar}}(i) + \sqrt{1-\delta_{\mathrm{solar}}} \cdot \hat{\beta}^{(j)}_{\mathrm{solar}}(i) \tag{73}$$

In addition to the set of hourly time series, a single daily time series with an autocorrelation of 0.29 [46] is generated, and used for all PV arrays. From Gaussian white noise $\beta_{\mathrm{daily}}(1,\ldots,d,\ldots,365)$:

$$\hat{\beta}_{daily}(d) = (0.29)\hat{\beta}_{\mathrm{daily}}(d-1) + \sqrt{1-(0.29)^2}\beta_{\mathrm{daily}}(d) \tag{74}$$

From these Gaussian time series, the methods of Graham, Hollands, and Huget can be used to find the insolation incident on a PV array, and from there the power can be easily found. Details are not provided here, but the reader is directed to consult the original source material [46][45][50] and the HOMER documentation [70]. The power is calculated for each PV array, and summed:

$$P_{\mathrm{PV},t} = \sum_{j=1}^{N_{\mathrm{PV}}} P_{\mathrm{PV,rated}}^{(j)} \cdot \Pi_{\mathrm{PV},t}^{(j)} \tag{75}$$

Where N_{PV} is the number of PV installations, $P_{\mathrm{PV,rated}}^{(j)}$ is the rated power of each installation (assumed the same for all but one fractional array), and $\Pi_{\mathrm{PV},t}^{(j)}$ is the normalized power for installation j at time t.

7.3.4.1 Atmospheric Clearness Sensitivity

The average monthly clearness values $\bar{k}_t$ can be modified as a sensitivity variable. However, there are physical limits on their values. Average monthly clearness should go below 0.3 or above 0.7, since these are the ranges over which data exist [50]. Furthermore, monthly values will already exist from data. A method was implemented to smoothly shift these values collectively without ever violating the upper or lower limits.

A noise sensitivity variable was used, called κ, that was allowed to vary between -0.3 and 0.3. At values very close to zero, it served to directly modify the monthly clearness values $\bar{k}_t$. However, at the extremes, the modified $\bar{k}_t$ values asymptotically approached 0.3 or 0.7.

The modification to $\bar{k}_t$ was as follows. First, $\bar{k}_t$ and κ are normalized,

$$K = \frac{\bar{k}_t - 0.3}{0.7 - 0.3} \tag{76}$$

$$\gamma = \frac{\kappa}{0.7 - 0.3} \tag{77}$$

Then, for negative values of κ, the normalized value is adjusted:

$$\rho^{-} = 2K \tag{78}$$

$$K'^{-} = \frac{\rho^{-}}{1 + \exp -4\gamma(1/\rho^{-})} \tag{79}$$

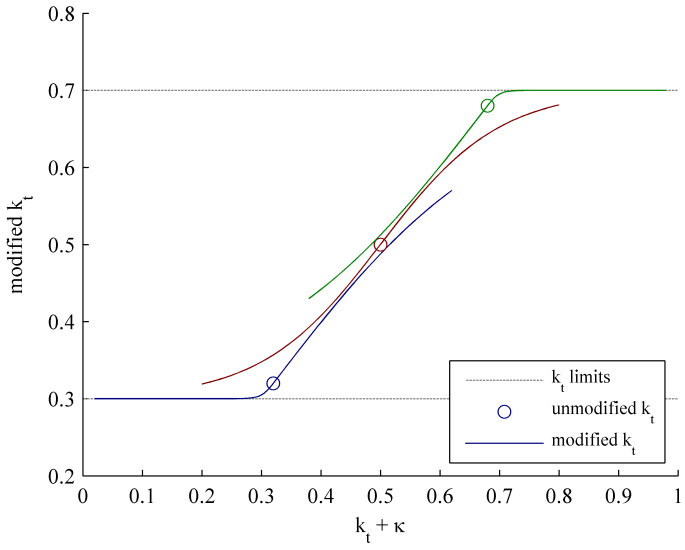


Figure 39: Sensitivity modifications to mean monthly insolation $\bar{k}_t$. At small levels of modification, the effects are linear, but near the limits of 0.3 and 0.7, the changes smoothly approach zero.

For positive values of κ, the adjustment is:

$$\rho^{+} = 2(1 - K) \tag{80}$$

$$K'^{+} = K + \rho\left(\frac{1}{1 + \exp -4\gamma(1/\rho^{+})} - \frac{1}{2}\right) \tag{81}$$

Finally, it is de-normalized, to yield an adjusted $\bar{k}_t$:

$$\bar{k}'_t = 0.3 + K' \cdot (0.7 - 0.3) \tag{82}$$

The nature of the modification is shown in Figure 39. When κ is close to zero (the lines near the original values), it acts as a direct modifier, in units of $\bar{k}_t$. However, this is prevented from moving the value past its limits. When the original value of $\bar{k}_t$ is close to the limits, it is allowed to move linearly away from the limit, but in the other direction it is immediately prevented from going past the limit.

7.3.4.2 Test Case Assumptions

In the test cases, a location of Atlanta, GA was assumed for the solar calculations (though the previous assumption of a mean wind speed of 7.5m/s is highly inconsistent with this location). The monthly clearness factors used can be seen in Table 5, taken from NASA's

Table 5: Monthly Average Clearness used in Model

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0.504	0.517	0.545	0.575	0.541	0.552	0.532	0.523	0.548	0.576	0.522	0.493

Surface Meteorology and Solar Energy database [7], and accessed using the HOMER model [70].

7.3.5 Energy Storage

Energy storage is implemented as generic and technology-independent, at a very low level of detail. Storage equipment is specified by a one-way efficiency η_{1way}, an internal energy capacity E_C, and a capacity-to-power factor ω (in units of time).

At any time step, if the storage is to be charged at a given power P_{charge}, the internal energy E simply changes by:

$$E_t = E_{t-1} + \eta_{1way} \cdot P_{charge} \cdot dt \tag{83}$$

Where t is the new time step, $t-1$ is the previous time step, dt is the length of the time step, and η_{1way} is the one-way efficiency. If it is to be discharged at a power $P_{discharge}$, the internal energy changes by:

$$E_t = E_{t-1} - \frac{P_{discharge} \cdot dt}{\eta_{1way}} \tag{84}$$

Capacity and power limits are implemented as simple constraints. There are no additional modeling details specific to any technology.

7.3.5.1 Storage Policy Optimization

It is assumed that the primary use of the storage is for removing unpredictable variability from load demand and non-dispatchable sources. Demand is subtracted from the non-dispatchable power from wind and PV, to find the "surplus".

$$P_{\text{surplus}} = (P_{\text{wind}} + P_{\text{PV}}) - P_{\text{demand}} \tag{85}$$

The result will probably usually be negative, demand exceeds the renewable energy supply. The remaining demand will have a certain degree of unpredictability. It is assumed that

energy storage will be used to reduce the unpredictability as much as possible. To that end, an optimization routine is used to determine storage charge/discharge policy.

The surplus signal is filtered with a simple exponential moving average high-pass filter, with coefficient α_{storage}:

$$P_{\text{low},t} = \alpha_{\text{storage}} \cdot P_{\text{surplus},t} + (1 - \alpha_{\text{storage}}) \cdot P'_{\text{low},t-1} \tag{86}$$

Where P'_{low} is a modified surplus signal with some high-frequency noise removed. The removed high-frequency noise is zero-mean, and is used as the charge/discharge command for the storage.

$$P_{\text{high}} = P_{\text{surplus}} - P_{\text{low},t} \tag{87}$$

$$P_{\text{C/D}} = f(P_{\text{high}}) \tag{88}$$

Where $P_{\text{C/D}}$ is the actual charge/discharge power of the energy storage device (+ is charge). A for a perfect storage device with infinite power and capacity, and unity efficiency, the charge/discharge power would equal the high-frequency noise exactly. However, the storage is limited by power and capacity, so it will not necessarily be able to follow this signal exactly. Whatever it *can* do modifies the surplus signal further:

$$P'_{\text{surplus}} = P_{\text{surplus}} - P_{\text{C/D}} \tag{89}$$

And the demand is now modified, taking into account renewables and storage:

$$P'_{\text{demand}} = max(-P'_{\text{surplus}}, 0) \tag{90}$$

This new demand will on average be higher, since the storage is not perfectly efficient. However, it will have lower autocorrelation, since some of its high-frequency variability has been removed by the storage. The degree to which the autocorrelation has been reduced will be a function of the storage power and capacity, but also of the filter parameter α_{storage}. As an iterative step, α_{storage} is optimized to result in the greatest reduction in autocorrelation. This will result in the lowest fossil plant spinning reserve requirement, as calculated in the next step.

7.3.5.2 Test Case Assumptions

In the test cases, it was assumed that the one-way efficiency $\eta_{1\text{way}} = 0.837$, and that the capacity-to-power ratio $\omega = 2.0\text{h}$.

7.3.6 Fossil Plant Spinning Reserve

After demand has been reduced by non-dispatchable sources and smoothed out with storage, what remains will need to be met with dispatchable energy sources, in this model consisting primarily of fossil plants. To be able to meet short-term fluctuations in demand, utilities must keep some fossil plants online but producing at below capacity, so that they can be throttled up to meet an increase in demand. The difference between a plant's output and its maximum output is called its *spinning reserve*.

The more predictable the demand, the lower the required spinning reserve. Since plants that operate below their peak capacity are less efficient, utilities do not want to have more spinning reserve than needed.

In this tool, required spinning reserve is calculated as:

$$P_{\text{reserve}} = c_{\text{reserve}} \cdot \sqrt{2 - 2 \cdot R_{\text{demand}}} \cdot \sigma_{\text{demand}} \tag{91}$$

Where σ_{demand} is the standard deviation of the modified demand signal, R_{demand} is the (unitless) autocorrelation of the modified demand signal, and c_{reserve} is a coefficient that depends on how reliable the designer wants the system to be. In the simple case of demand that consists of autocorrelated Gaussian noise, a reserve coefficient of 2.33 would mean that the designer wants to have enough reserve to deal with a 99th percentile hourly spike in demand. Thus lower demand variance and higher demand autocorrelation result in a lower spinning reserve requirement. In the model, c_{reserve} is kept constant throughout the year; in reality, it will change over time, as demand volatility is not constant, and in fact much more advanced methods would be used to calculate the reserve requirement.

7.3.6.1 Test Case Assumptions

In the test cases, a reserve reliability factor of 0.9999 was used, which meant $c_{\text{reserve}} = 3.72$. In the tests found later in this chapter, this equation was calculated incorrectly, and this

likely affected the results somewhat. It was corrected for the final demonstration case.

7.3.7 Natural Gas Plants

Natural gas plants are represented with a simple linear fuel consumption relationship:

$$\mathcal{F} = a_{\text{demand}} \cdot P'_{\text{demand}} + a_{\text{reserve}} \cdot P'_{\text{reserve}} \tag{92}$$

That is, there is a fuel rate per unit of produced power, and a (smaller) fuel rate per unit of spinning reserve. Rather than model plants as discrete units, instead they are modeled as a single aggregate plant. The spinning reserve requirement is taken form the previous section, and an additional half plant is added:

$$P'_{\text{reserve}} = P_{\text{reserve}} + \frac{1}{2} P_{\text{gas,rated}} \tag{93}$$

Where $P_{\text{gas,rated}}$ is the rated power of an individual plant, assumed to be the same for all plants. The important assumption is that, since only integer plants can be brought online, rarely will the required amount of spinning reserve be met exactly; and since the spinning reserve must always be greater than or equal to the requirement, on average half of a plant's worth of extra spinning reserve will be online.

If the desired reserve cannot be met, because there are not enough plants available, there will be a capacity shortage:

$$P_{\text{capacity shortage}} = max(0, P_{\text{reserve}} + P'_{\text{demand}} - \sum_{(j=1)}^{N_{\text{gas}}} P^{(j)}_{\text{gas,rated}}) \tag{94}$$

The last term, the total plant capacity, is simply specified by the designer, so though (as with wind and solar power) there might be some "fractional" plant, there is no need to calculate what it is or deal with it explicitly.

7.3.7.1 Natural Gas Price Sensitivity

In reality, natural gas price is volatile, and fluctuates with some degree of unpredictability. In the model, it is not treated as a stochastic time series, but instead as a constant multiplier on fuel price, and as such can be treated as a noise variable. As with the overall approach of averaging costs over a full year, this tends to mask cost volatility and is not ideal.

7.3.7.2 Test Case Assumptions

For the test cases in this chapter, the plant installed power was $P_{\text{gas,rated}} = 200\text{MW}$, and in the final demonstration case it was set to 10MW. Spinning reserve was set to consume fuel at 20% the rate of produced power.

7.3.8 Market Purchases

Whatever power cannot be supplied by all sources in the portfolio is assumed to be obtained through market purchases from other regions. The specific pricing structure for this is not modeled; instead, a uniform high cost is assumed.

7.3.8.1 Market Power Price Sensitivity

Though market price volatility is not modeled with a time series, it can be adjusted as a noise variable.

7.3.9 Calculating Cost of Energy

For a particular portfolio, the cost of energy was found as:

$$\text{COE} = \frac{C_{(\text{fuel})} + C_{(\text{market purchases})} + \sum(C_{(\text{capital})})}{\sum P_{\text{demand}} \cdot dt} \tag{95}$$

Where the C's are just individual costs. The capital costs were given by the designer in units of annualized costs, the fuel was assumed to have a fixed price per unit, and market purchases were assumed to have a fixed price per unit energy. In the examples and tests in the remainder of this document, cost of energy will be given in units of \$/MWh.

7.4 Characterizing the Output Space

In order to characterize the output space, a very large (N = 8000) Latin Hypercube design of experiments was created, along with a large set of random validation cases (N = 2000). Seven of the model's inputs were varied, over ranges shown in Table 6.

Average wind speed and average clearness were the only two noise variables studied. The reason for this was not that they were the only variables of interest, but because the effects of other noise variables could be studied without running simulations. Natural gas price

Table 6: Design of Experiments Ranges for Simulation Model Testing

Type	Variable	Units	Min	Max
Design	Wind turbines	MW rated capacity	0	2000
	PV installations	MW rated capacity	0	2000
	Energy storage	MWh capacity	0	2000
	Natural gas plants	MW rated capacity	0	2000
	Demand Side Management	"units"	0	100
Noise	Wind speed	m/s	6	9
	Average clearness	κ	-0.12	0.12

Table 7: Neural Network Surrogate Model Fits

Response	R^2	RMSE* (training data)	RMSE* (holdout data)
Natural Gas Fuel Used	0.998	0.0083	0.0080
Purchased Energy	0.999	0.0051	0.0054

*Error was normalized over the range of the responses

and market energy cost had *purely linear* effects; if fuel usage and market energy usage were tracked as outputs of the simulation model, there was no need to run simulations for them. Total demand sensitivity, while not strictly linear, could be approximated by re-scaling the entire system.

Running 10,000 simulations did not take very long; thanks to the simplicity of the model, it completed in under an hour. Once the data was collected, neural network surrogate models were fit to two of the simulation outputs, namely fuel used and purchased energy. A summary of their fits is shown in Table 7.

Once these two responses had been regressed, and with the additional knowledge that adding DSM reduced the total demand by a linear factor, the energy cost could be very inexpensively found as a function of design and noise variables. Cost was calculated with the surrogates, and the variables of demand sensitivity and market energy price sensitivity could now be adjusted. Capital costs could also be adjusted, but were not treated as noise variables. For simplicity, all noise variables were assumed to have triangular distributions, with minimum, maximum, and most likely values, and in some cases their ranges were restricted compared to their regression ranges. The full list of variables are shown in Table 8.

This allowed visualization and testing, in order to characterize the behavior of the model

Table 8: Full List of Input Variables

Type	Variable	Units	Min	Max	Most Likely
Design	Wind turbines	MW rated capacity	0	2000	
	PV installations	MW rated capacity	0	2000	
	Energy storage	MWh capacity	0	2000	
	Natural gas plants	MW rated capacity	0	2000	
	Demand Side Management	"units"	0	100	
Noise, internal	Wind speed	m/s	6	9	7.5
	Average clearness	κ	-0.08	0.08	0.0
Noise, external*	Natural Gas Price	\$/MWh	24	60	30
	Demand	(mult)	0.8	1.2	1.0
	Market Transaction Price	\$/MWh	100	300	200

*The effects of these variables could be found by linear re-scaling of the surrogates

and (in the next section) to develop a fully analytic test problem with similar properties.

7.4.1 Visualizing the Output Space

After surrogates had been regressed, the design space could be visualized. A series of slices are shown in Figure 40. The subspaces appear quite smooth. The first two slices have single local minima; the last appears largely linear. Bear in mind that these are just slices. Capital costs have been fixed at values which are described in a later subsection.

A set of slices through noise space are shown in Figure 41. All noise subspaces, at least for these design variable settings, are monotonic. The market transaction price and gas price subspaces are perfectly linear, because that is how they are constructed.

For all the subspaces visualized, the response is smooth, and at least in these slices, it appears unimodal. The noise sensitivity variables have close to linear effects, even the ones which are not linear by construction. With the exception of Figure 40(b), all of these subspaces could probably be fit well by polynomials. Some observations, then:

Observation: The cost response appears smooth over the design variables, and possibly unimodal.

Observation: The cost response appears monotonic and close to linear over the noise variables.

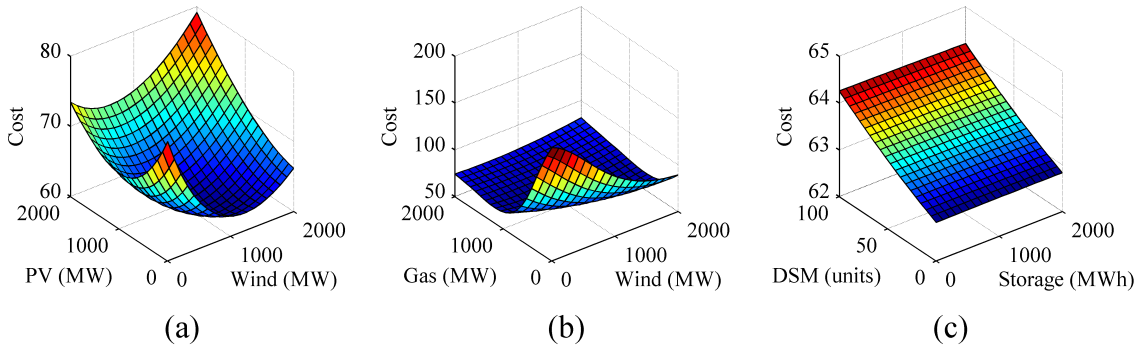


Figure 40: Slices of energy cost as a function of design variables. In all plots, all other inputs have been set to their midpoints. From L tor R, (a) shows installed wind and PV, (b) shows installed wind and natural gas plant capacity, and (c) shows energy storage and DSM. Both (a) and (b) have minima, and (c) is monotonic in both dimensions. The effects of DSM and storage are milder, at least for these noise variable settings, but the effects are non-zero. All costs are in $/MWh.

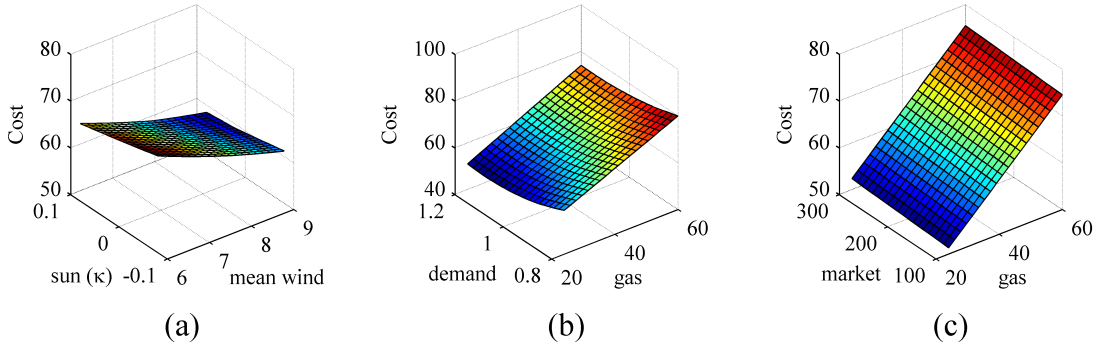


Figure 41: Slices of energy cost as a function of noise variables. In all plots, all other inputs have been set to their midpoints. From L to R, (a) shows cost declining with increased mean insolation and wind speed, (b) shows costs increasing with natural gas price, and (c) shows only a small variation with market transaction price, but only because this portfolio does not require many outside energy purchases. All subspaces are relatively linear. All costs are in $/MWh.

It is possible, however, that some features are being masked by the neural network surrogates used in this exploration, and it is difficult to fully generalize based on a few slices. The next subsection will explore the surrogates using optimization.

7.4.2 Multi-Objective Optimization of the Model

With the fast-running neural network surrogates, exhaustive techniques could be used to optimize the design space. For every design of interest, 10,000 Monte Carlo samples were used to estimate the mean Cost of Energy, and the 95% Value-at-Risk. The capital costs were (somewhat arbitrarily) set as shown in Table 9, and the NSGA-ii Multi-Objective Genetic Algorithm with a population of 5,000 was run for 1,000 generations. The resulting Pareto frontier is shown in Figure 42. In objective space, it can be seen that the range of the frontier is very small, even though the range over all possible portfolios is quite large, as can be seen from a scatterplot of random designs in Figure 43. Relative to the output range, then, high accuracy will be required in order to distinguish the frontier from the rest of the objective space:

Observation: The range of the Pareto frontier is small relative to the total range of the objectives.

Observation: The Pareto frontier is (mostly) concave.

In such an extreme case, it could be argued that the designer does not really care about the whole frontier, since in absolute terms any portfolio on such a small frontier is essentially indistinguishable from another. However, it will be shown later that different problem assumptions result in a larger frontier, but which is still small relative to the entire objective space. Looking at electric power utility IRPs, NorthWestern's frontier has a mean energy cost range of $69-85/MWh, and a risk range of approximately $80-90/MWh [84]. This is still significantly smaller than the several hundred $/MWh range of the entire portfolio space.

The normalized design variable settings along the frontier are shown in Figure 44, using a plot type after Daskilewicz [25]. The position along the x-axis shows the position along

Table 9: Capital Costs, Scenario 1

Equipment	Annualized Capital Cost	Units
Wind Turbines	100,000	$/MW
Photovoltaic Arrays	150,000	$/MW
Energy Storage	100,000	$/MWh
Natural Gas Plants	60,000	$/MW
Demand Side Management	50,000	$/unit

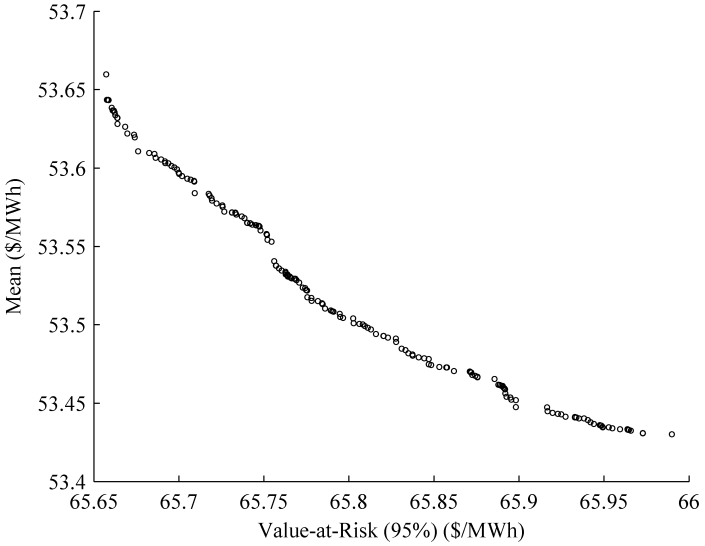


Figure 42: Pareto frontier for Scenario 1. Note the very small range on both mean and VaR.

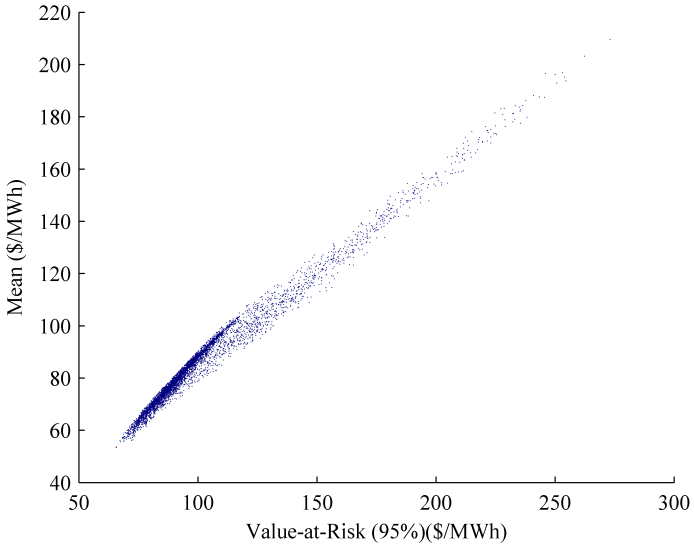


Figure 43: Scatter plot of random designs, Scenario 1. Note that the Pareto frontier is barely a speck in the lower left-hand corner. Under this set of assumptions, the Pareto frontier is a very small fraction of objective space.

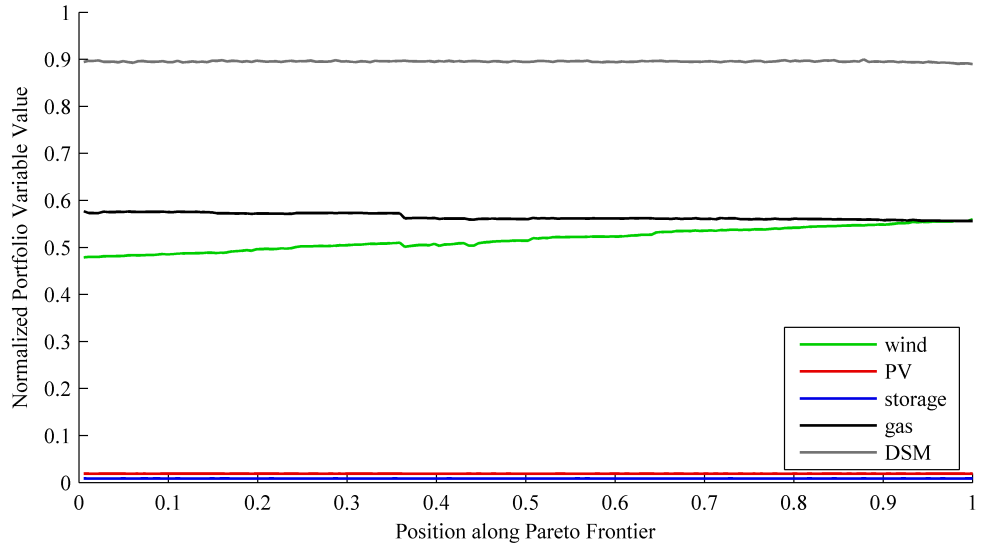


Figure 44: Daskilewicz-style [25] plot of design variable values over a normalized frontier. From L to R, the mean increases from low to high.

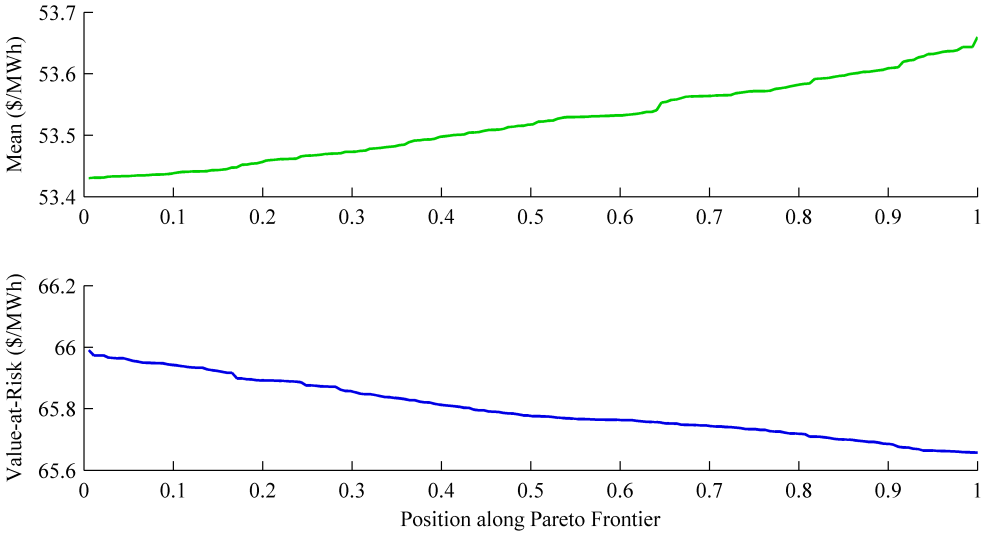


Figure 45: Mean and value-at-risk along the normalized frontier. The x-axis is the same as in the previous plot.

Table 10: Capital Costs, Scenario 2

Equipment	Annualized Capital Cost	Units
Wind Turbines	110,000	$/MW
Photovoltaic Arrays	95,000	$/MW
Energy Storage	7,500	$/MWh
Natural Gas Plants	85,000	$/MW
Demand Side Management	120,000	$/unit

the Pareto frontier from Figure 42, from low mean to high mean. Figure 45 shows the mean and VaR values along the frontier. From looking at Figure 44, it can be seen that the design variables vary smoothly. PV and storage remain near zero, unsurprising due to their high cost.

Under the assumptions used so far, the Pareto frontier appears quite simple. In design space, it is a continuous path. This need not always be the case, however. It was found that the nature of the frontier was sensitive to assumptions. In Table 10, a second set of capital costs is shown. Though this set is further from reality than the previous one (note the very low PV and storage cost, for example), it results in a drastically different frontier. Though the shape in objective space appears similar, and the objective range is still small, in design space it is clear from Figure 47 that there is a large jump in DSM values at around 0.6, and several smaller jumps elsewhere. The optimal amount of DSM moves quickly from the low end of its range to the high range. Other variables also appear to "jump" at the same points, though over smaller ranges.

This scenario was found, it should be noted, essentially by "optimizing" the capital costs to result in more interesting behavior. A design problem with a multi-part frontier is harder to solve, since it requires locating two or more parts of the space; it is the multi-objective equivalent of a multi-modal optimization problem. If such a scenario can be artificially created by adjusting cost assumptions, then it is reasonable to assume that it *could* happen in a real problem, and it is reasonable to test methods on problems with this more challenging characteristic.

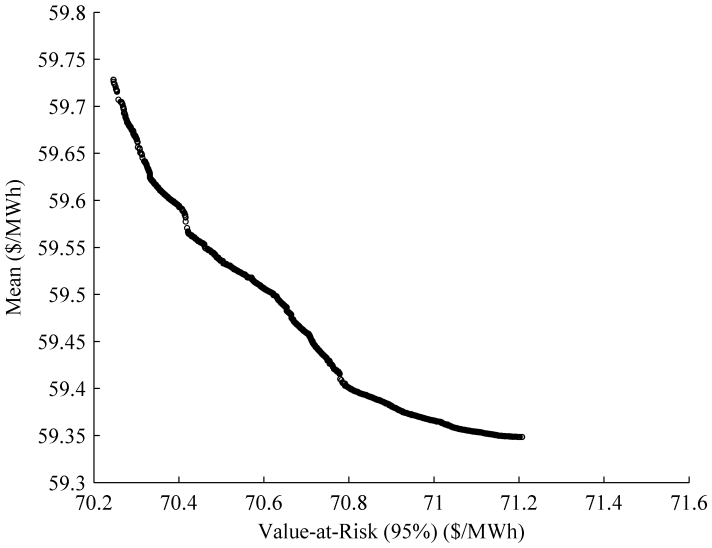


Figure 46: Pareto frontier, Scenario 2. The general shape of the frontier appears similar to in Scenario 1, though with different values because the cost assumptions have been changed.

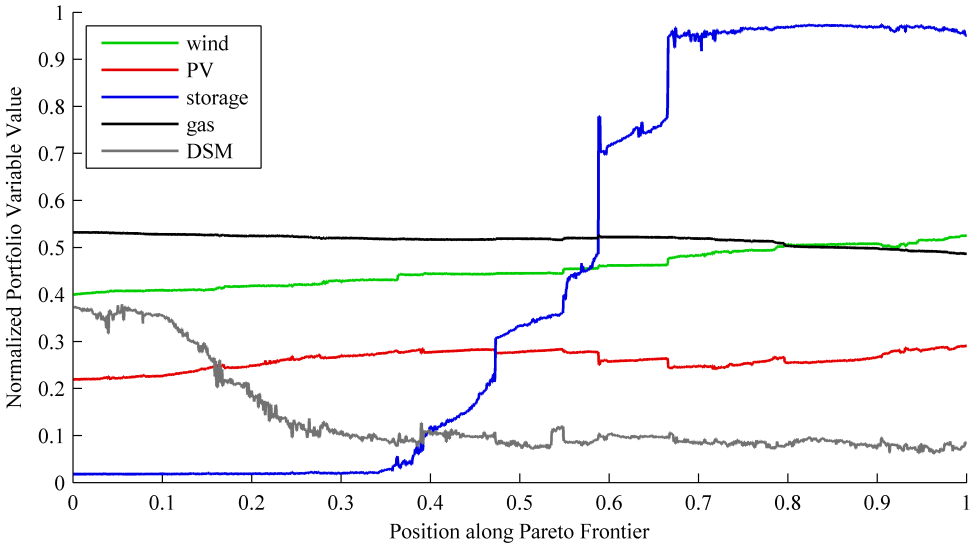


Figure 47: Daskilewicz-style [25] plot of design variable values over a normalized frontier. From L to R, the mean increases from low to high.

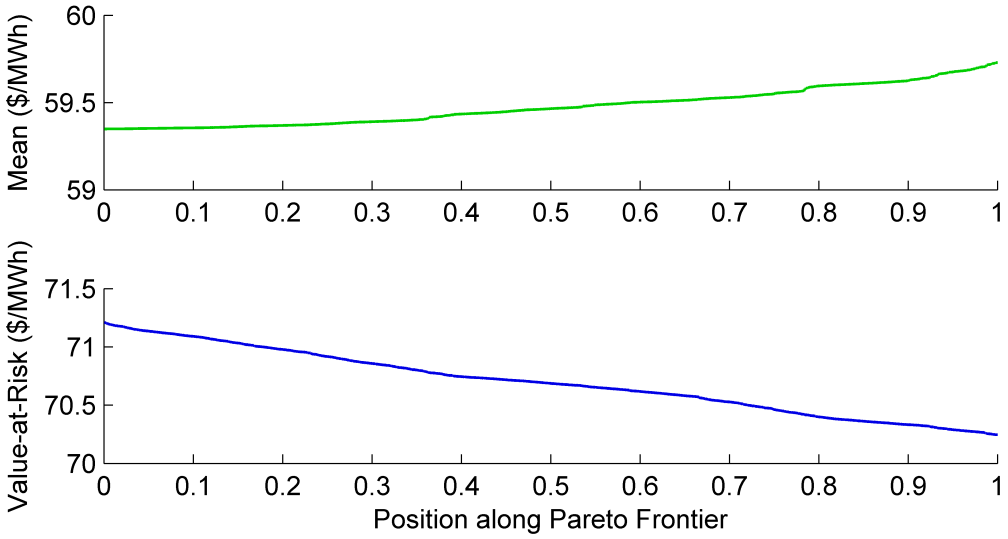


Figure 48: Mean and value-at-risk along the normalized frontier. The x-axis is the same as in the previous plot.

Observation: Under certain sets of assumptions, the frontier consists of multiple discrete sections.

Based on the set of observations about the design problem, the next chapter will develop an analytic test problem that shares those characteristics, as well as other characteristics which make it useful for experimentation.

CHAPTER VIII

SCALABLE TEST PROBLEM

In the previous chapter, a low-fidelity electric portfolio test problem was developed and characterized. In a later chapter, it will be used to demonstrate that the method can be used to efficiently find the mean/risk Pareto frontier for such a problem. However, recall that two of the research questions pertained to the sensitivity of the method to the numbers of design and noise variables:

> **Research Question 1:** For finding mean/risk Pareto frontiers, how does the relative efficiency of combined and crossed arrays depend on the number of noise variables?

> **Research Question 2:** For finding mean/risk Pareto frontiers, how does the relative efficiency of design of experiments and multi-objective statistical improvement change with the number of design variables?

Answering these questions will help inform a designer as to when the combined-array multi-objective statistical improvement method should be used, as opposed to another method. In order to test the relative sensitivity of the methods, it will be desirable to change the number of design and noise variables without changing any other characteristics of the problem. To that end, a scalable test problem is developed here.

8.1 Test Problem Characteristics

The test problem should be as similar as possible to an electric power portfolio problem. Recall the observations from the last chapter:

- The cost response appears smooth over the design variables, and possibly unimodal.
- The cost response appears monotonic and close to linear over the noise variables.

- The range of the Pareto frontier is small relative to the total range of the objectives.
- The Pareto frontier is (mostly) concave.
- Under certain sets of assumptions, the frontier consists of multiple discrete sections.

In addition to matching these characteristics, there are several other characteristics that are desirable for a test function:

- **Scalable**: As the number of design and noise variables is changed, the problem should not change fundamentally in any other way
- **Analytically known frontier**: In order to judge the effectiveness of the method, the true Pareto frontier should be known exactly, as well as the mean and risk values along it
- **Fast to evaluate**: The function should evaluate very quickly, to allow as many tests as possible to be run

8.2 Test Problem Description

A test problem was created to meet the previously described characteristics.

8.2.1 Analytical Pareto Frontier

In order to better control the nature of the Pareto frontier, and so that the frontier would be known analytically, the frontier itself was first described functionally in objective space. A dummy variable, X_D, was allowed to vary from 0 to 1, and mean and Value-at-Risk were defined in terms of it. First, after a series of constants were defined, a function $Y(X_D)$ was

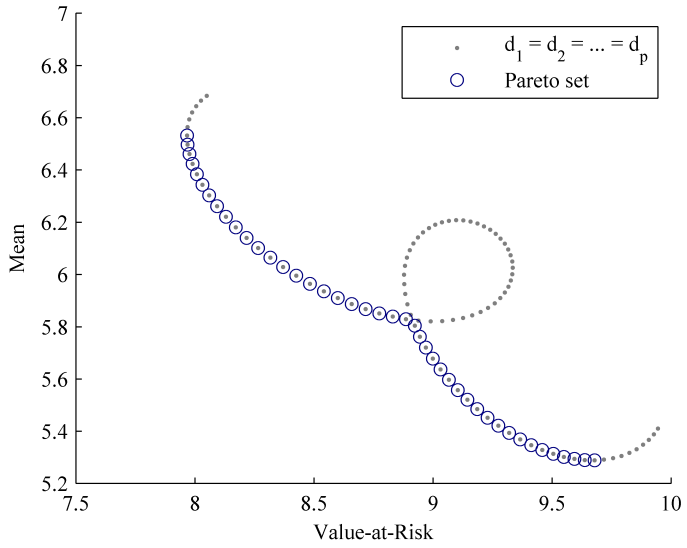


Figure 49: The Pareto frontier of the scalable test problem. Gray points are those along the dummy variable X_D from 0 to 1, blue circles are those points that lie on the frontier.

defined:

$$a = \arcsin\left(\frac{-4}{5\pi}\right) \tag{96}$$

$$b = \frac{-2a}{5\pi - 4a} \tag{97}$$

$$C_0 = \frac{1}{4}\cos\left(\frac{5\pi b}{4b-2}\right) + \frac{b}{2-4b} = 0.2582 \tag{98}$$

$$C_1 = 1/\left(C_0 - \frac{\cos(2.5\pi(1-b))}{4(1-2b)} + \frac{1-b}{2(1-2b)}\right) = 1.1930 \tag{99}$$

$$Y(X_D) = C_1\left(\frac{(X_D - 0.1)/0.8 - b}{2(1-2b)} - \frac{5\pi}{8}\cos\left(\frac{(X_D-0.1)/0.8 - b}{1-2b}\right)\right) \tag{100}$$

If mean and risk along the Pareto frontier are defined in terms of this function as follows,

$$\mu_P = Y(X_D) + \mu_0 \tag{101}$$

$$\rho_P = Y(1 - X_D) + \rho_0 \tag{102}$$

then the set of points along X_D will trace a curve in objective space, shown in Figure 49. The mean and risk as a function of X_D are shown in Figure 50.

This resulted in a Pareto frontier with two discrete sections and a small slightly convex region in the center. Note that as will be shown in a later section, the exact values of mean and value-at-risk will differ slightly from these equations, but not enough to drastically

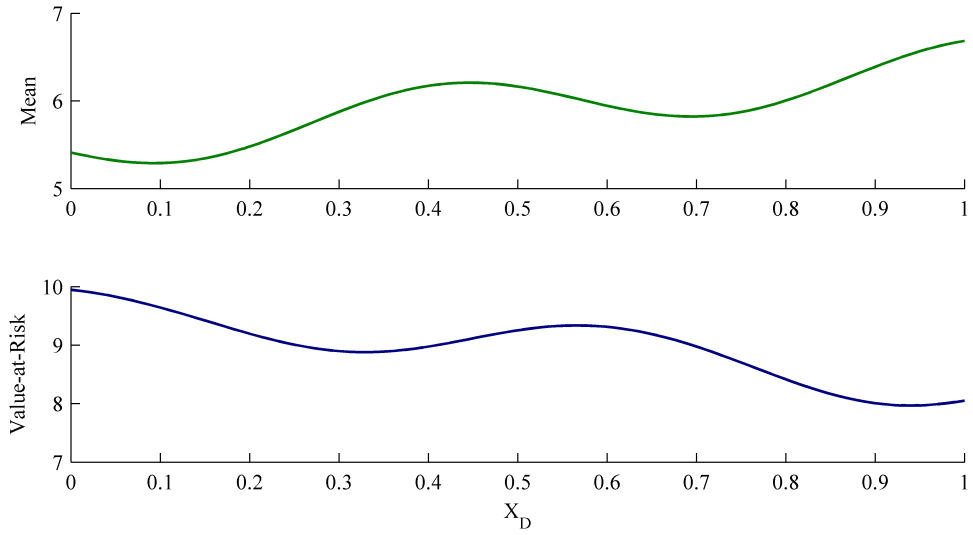


Figure 50: The objectives of mean and Value-at-Risk as functions of the dummy variable X_D.

change the characteristics of the frontier.

8.2.2 Design Space

In order to make the test function fully scalable in terms of design variables, the frontier was defined as always occurring along the line ($d_1 = d_2 = d_3 = ... = d_{p_D}$), where p_D is the number of design variables. All input variables were confined to the range [0,1].

To transform the design vector D into the dummy variable X_D, the design variables were simply summed and normalized:

$$X_D = \frac{\sum_{i=1}^{p_D}(d_i)}{p_D} \tag{103}$$

To ensure that the Pareto frontier occurred only along the line ($d_1 = d_2 = d_3 = ... = d_{p_D}$), a penalty function was implemented,

$$\mathcal{B} = 15 \cdot \left(\frac{\sum_m |X_D - D_m|}{p_D} \right)^2 \tag{104}$$

which is simply the square of the 1-norm distance to the point nearest in Euclidean space. The idealized mean (μ^*) and value-at-risk VaR* were then determined as:

$$\mu^* = Y(X_D) + 5 + \mathcal{B}(X_D) \tag{105}$$

$$\text{VaR}^* = Y(1 - X_D) + 7 + \mathcal{B}(X_D) \tag{106}$$

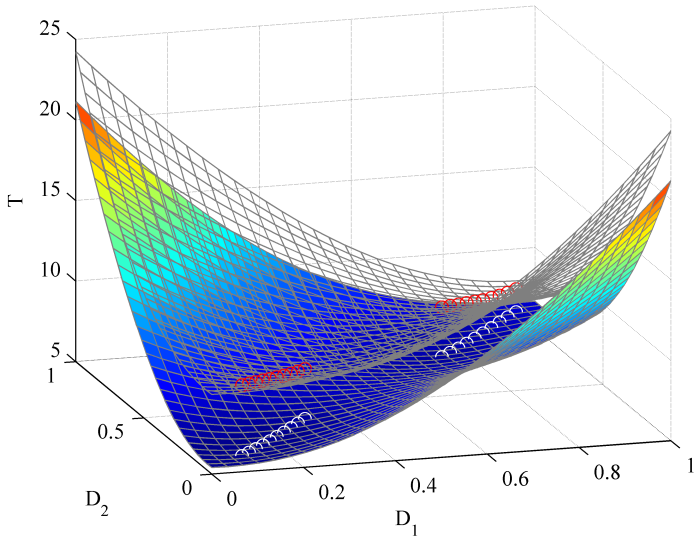


Figure 51: An implementation of the test function with two design variables. The colored surface is the mean, and the gray mesh is the Value-at-Risk. The frontier lies in the trough along the line ($d_1 = d_2$), and is represented by white circles (mean) and red circles (VaR).

For a two-objective problem, the mean and VaR are shown as a function of design space in Figure 51. A scatter plot of the mean and VaR objective values for randomly selected designs is shown in Figure 52, showing that they are spread out over a range of about 20, and the Pareto frontier is a relatively small 1x1 box in the lower left corner.

Thus for any design D, a value for mean and Value-at-Risk are determined. However, as will be seen next, these values are merely approximate.

The frontier can be seen to consist of two discrete sections, one approximately between $0.1 < X < 0.29$, the other between approximately $0.71 < X < 0.9$. The exact extent will be defined later.

8.2.3 Noise Space

It was assumed that all noise variables were independent and standard normal:

$$S_i \sim \mathcal{N}(0, 1) \tag{107}$$

$$\mathbf{S} \sim \text{MVN}(\mathbf{0}, \mathbf{I}_{p_S}) \tag{108}$$

Where p_S is the number of noise variables.

A perfectly linear noise space with slope dT/dS would result in an output distribution

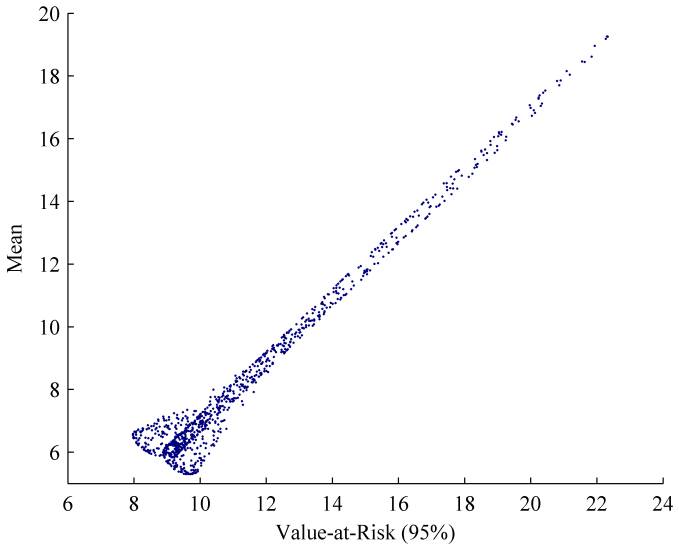


Figure 52: Scatterplot of objective values for randomly selected designs. This is the equivalent of Figure 43 from Chapter 7. The Pareto frontier represents a small fraction of the objective space, occupying a 1x1 box when both objectives range up to about 20.

that is also normal, with a variance easily found analytically:

$$(\sigma^2)^* = \sum_i^{p_S} \sigma_i \left(\frac{dT}{dS}\right)_i^2 = \sum_i^{p_S} \left(\frac{dT}{dS}\right)_i^2 \tag{109}$$

Where each σ_i^2 is the variance of noise dimension i, and these are all assumed to be equal to 1.0. Furthermore, this would result in an analytically known Value-at-Risk:

$$\text{VaR}_\alpha = \mu + \Phi^{-1}(\alpha) \cdot \sigma \tag{110}$$

where α is the confidence level, Φ^{-1} is the standard inverse normal CDF, and the term $\Phi^{-1}(\alpha)$ is referred to as c.

For a moment, if a perfectly linear noise space is assumed, and if it is furthermore assumed that the slope is equal in all dimensions, then the already-determined Value-at-Risk can be used to define the noise slope for each dimension:

$$\frac{dT}{dS}_i = \frac{dT}{dS}_j = \sqrt{\frac{(\text{VaR} - \mu)^2}{c^2 p_S}} \tag{111}$$

$$a^* \tag{112}$$

where the label (*) refers to a term which has been prescribed.

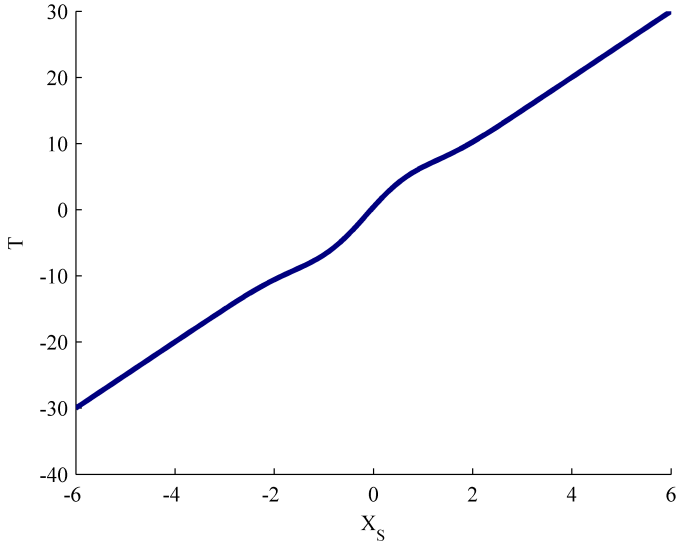


Figure 53: Response of test function as a function of noise dummy variable X_S.

However, even though the portfolio simulation model was shown to have a *roughly* linear noise space (and was in fact perfectly linear in several dimensions), a *perfectly* linear noise space will be "too easy" to model, unrealistically so. A crossed-array DoE with only $p_S + 1$ points would give perfectly accurate estimates of all statistics (where p_S is the number of noise variables). Instead, a function was implemented that was monotonic and close to linear in every dimension, but not *perfectly* linear.

As with the design space component of the test function, in noise space the function depends on a dummy variable, X_S, which is a sum of the noise variables, this time weighted to adjust the noise sensitivity.

$$X_S = \sum_i a^* S_i \tag{113}$$

This dummy variable is fed into a function which is approximately, but not precisely, linear, as shown in Figure 53. The function is a linear term plus a sum of radial basis functions:

$$Y_S(X_S) = X_S + \sum_{j=1}^{q} a_Y^j \exp(-\frac{1}{6}(X_S - a_X^j)^2) \tag{114}$$

where the summation is over basis vectors j, each of which has coefficients a_Y^j and a_X^j. Three basis functions are used, and a table of the values used for the coefficients is provided

Table 11: Coefficients used to Compute Y_S

Basis no (j)	a_X	a_Y
1	-1.20735021812485	-9.20988520187053
2	0.751898663420942	17.6118822181628
3	1.45605761640806	-8.58052401975203

in Table 11. To evaluate the test function, then, the response is simply:

$$T = \mu_{input} + Y_S(X_S) \tag{115}$$

where μ_{input} is the mean value found from the analytic frontier and penalty function.

Because the noise space is approximately linear, but not exactly so, the *true* statistics will not match the idealized ones and must be re-calculated using numerical integration. However, regardless of how many noise variables there are, the dummy variable X_S will always be normally distributed:

$$X_S \sim \mathcal{N}(0, (\sigma^2)^*) \tag{116}$$

$$(\sigma^2)^* = p_S(a^*)^2 \tag{117}$$

This $(\sigma^2)^*$ is the same variance that *would* be seen in the output for a perfectly linear noise space, and it is set according to the analytically determined VaR. Since X_S is Gaussian, it is an easy matter to perform a 1-dimensional numerical integration and find the true mean and variance:

$$\mu = \int_{-\infty}^{\infty} Y_S(X_S) p(X_S) dX_S \tag{118}$$

$$\sigma^2 = \int_{-\infty}^{\infty} (Y_S(X_S) - \mu)^2 p(X_S) dX_S \tag{119}$$

and the true Value-at-Risk can be even more easily found:

$$\text{VaR} = Y_S(\text{VaR}^* - \mu^*) \tag{120}$$

where again μ^* and VaR^* are those that were determined by the analytical frontier function.

8.3 Summary of Test Function

The test function developed here is designed to generally match the characteristics of the motivating energy portfolio problem. Its value for any point in design and noise space is

known analytically. Moreover, the mean, variance, and value-at-risk for any design can be computed using simple 1-dimensional numerical integration, and its Pareto frontier is always known exactly. The problem can be scaled to have any number of design and noise variables without any other substantial changes, and so is suitable for studying the sensitivity of methods to problem dimensionality. A summary table of the relevant equations is shown below.

First, find the dummy variable value:

$$X_D = \frac{\sum_{i=1}^{p_D}(d_i)}{p_D}$$

Then, evaluate the Pareto function:

$$\begin{aligned}
a &= \arcsin\left(\frac{-4}{5\pi}\right) \\
b &= \frac{-2a}{5\pi - 4a} \\
C_0 &= \frac{1}{4}\cos(\frac{5\pi b}{4b-2}) + \frac{b}{2-4b} \\
C_1 &= 1/\left(C_0 - \frac{\cos(2.5\pi(1-b))}{4(1-2b)} + \frac{1-b}{2(1-2b)}\right) \\
Y(X_D) &= C_1\left(\frac{(X_D - 0.1)/0.8 - b}{2(1-2b)} - \frac{5\pi}{8}\cos(\frac{(X_D-0.1)/0.8-b}{1-2b})\right)
\end{aligned}$$

Next, calculate the off-Pareto penalty:

$$\mathcal{B} = 15 \cdot \left(\frac{\sum_m |X_D - D_m|}{p_D}\right)^2$$

and find the *idealized* (*) mean and Value-at-Risk, as well as the idealized standard deviation:

$$\mu^* = Y(X_D) + 5 + \mathcal{B}(X_D)$$

$$\begin{aligned}
\text{VaR}^* &= Y(1 - X_D) + 7 + \mathcal{B}(X_D) \\
\sigma^* &= \frac{(\text{VaR}^* - \mu^*)}{c}
\end{aligned}$$

If the goal is to evaluate the test function on design/noise values, use the noise sub-space

directly:

$$a^* = \frac{\sigma^*}{\sqrt{p_S}}$$
$$X_S = \sum_i a^* S_i$$
$$Y_S(X_S) = X_S + \sum_{j=1}^{q} a_Y^j \exp(-\frac{1}{6}(X_S - a_X^j)^2)$$
$$T = \mu^* + Y_S(X_S)$$

If the goal is to find the (true) mean and Value-at-Risk, integrate and evaluate:

$$\mu = \int_{-\infty}^{\infty} Y_S(X_S) p(X_S) dX_S$$
$$\sigma^2 = \int_{-\infty}^{\infty} (Y_S(X_S) - \mu)^2 p(X_S) dX_S X_S \qquad \sim \mathcal{N}(0, (\sigma^2)^*)$$
$$\text{VaR} = Y_S(\text{VaR}^* - \mu^*)$$

Since the problem is designed to be similar to an energy portfolio problem, it may not be perfectly suitable as a surrogate for all engineering problems. Its noise spaces are all monotonic, a characteristic found in the energy portfolio test problem but not to be assumed generally. Further, it is relatively smooth, and though it is multimodal it is not excessively so. The Pareto frontier occurs in two sections in design space, and has a small concavity in one part of the objective space. Lastly, in order to make the problem scalable, it depends functionally on a sum of all noise variables, and on an only slightly more complex function of design variables.

Now that a test problem has been developed that is scalable, Research Questions 1 and 2 with regard to the scalability of methods can be answered; and since the test problem runs in negligible time, all computational resources can be devoted to the methods themselves.

CHAPTER IX

EXPERIMENTS: WARM-START SIZE AND EFFICIENCY

In previous chapters, several general classes of methods were discussed for solving robust design problems. Of principal interest were two classifications: combined-array vs. crossed-array methods, and design of experiments vs. multi-objective statistical improvement methods. This leads to a classification of methods shown in Table 12. The shorthand acronyms found in the table will be used through the rest of this document. The lower right box, combined-array statistical improvement methods (C-MOSI), was identified as a gap in the literature.

In this taxonomy, the two groups of "combined-array methods" (C) and "crossed-array methods" (X) can each be thought of as continuums. A MOSI method begins with a "warm-start" DoE to fit an initial Bayesian surrogate; this warm-start could represent any fraction of the total samples, from a few percent to nearly the entire set.

Say the designer has some desired level of accuracy along the true Pareto frontier. They can achieve that level of accuracy by starting with a warm-start DoE, and running a MOSI method until the surrogate is sufficiently accurate. A set of possible "paths" are shown notionally in Figure 54, each starting at a warm-start DoE and increasing in accuracy until a threshold is reached.

Increasing the DoE size should result in higher initial accuracy, shown conceptually in

Table 12: Taxonomy of Methods

	Crossed Array	Combined Array
Design of Experiments	X-DoE	C-DoE
Multi-Objective Statistical Improvement	X-MOSI	C-MOSI

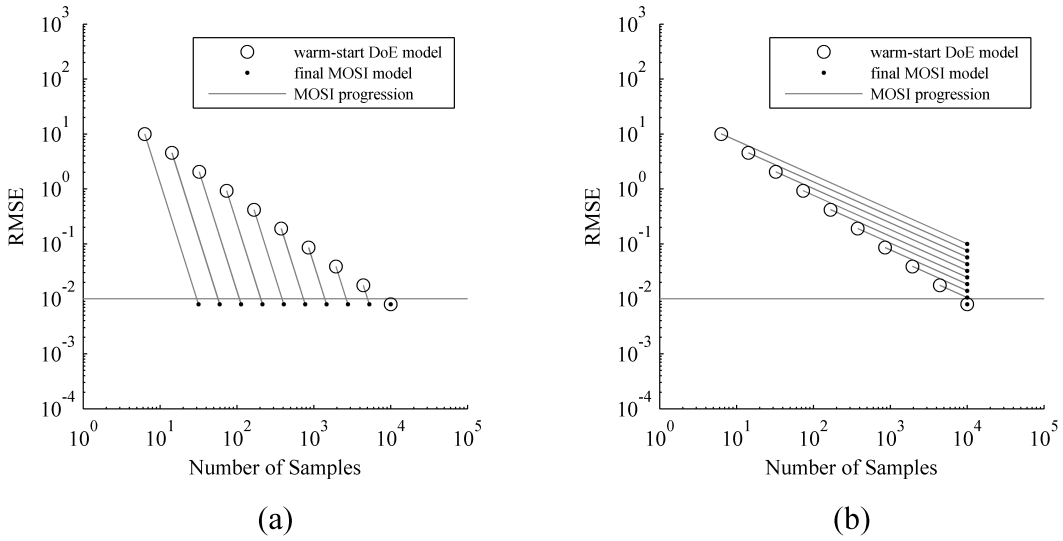


Figure 54: Notional warm-start DoEs (circles) and MOSI paths (ending in dots). In (a), MOSI sampling reduces error faster than increasing DoE size. In (b), it would be better simply to use a larger DoE rather than run MOSI.

Figure 54 by the reduced error of the larger DoE sizes. If the MOSI paths reduce error faster than is achieved by increasing DoE size, then MOSI can be shown to be more efficient, as in Figure 54(a). If, on the other hand, the paths remain "above" the DoEs as in (b), then the MOSI approach is a waste of samples.

Another way of thinking about the different warm-start sizes is in terms of initial number of samples and final number of samples. This is shown conceptually in Figure 55. Each dot in the figure represents a full execution of an adaptive sampling method, from a warm-start DoE until a satisfactory level of accuracy is reached. Presumably, there will be some warm-start size that is "optimal", that results in the fewest total samples in order to reach the desired level of accuracy. However, the process is somewhat stochastic; a Latin Hypercube DoE is randomly generated, and there is likely to be some degree of randomness and imperfection in the optimization processes used to select subsequent points. In Figure 55(a), the optimal warm-start size lies somewhere between A and B. A warm-start size of A samples has the potential for the lowest number of total samples, but is risky. A choice of B would reliably result in fewer samples than a pure DoE approach. By point C, a DoE by itself reliably provides sufficient accuracy. Because the upper bound at B is lower

than C, a MOSI method can be expected to be more efficient with high confidence.

A second possible result is in Figure 55(b), where the warm-start size is not reliably large enough until it reaches the size necessary for a DoE-only approach. In such a case, using a MOSI approach offers no benefit; in half the of attempts, it reaches the accuracy threshold in fewer than C runs, but in the other half it requires more than C runs.

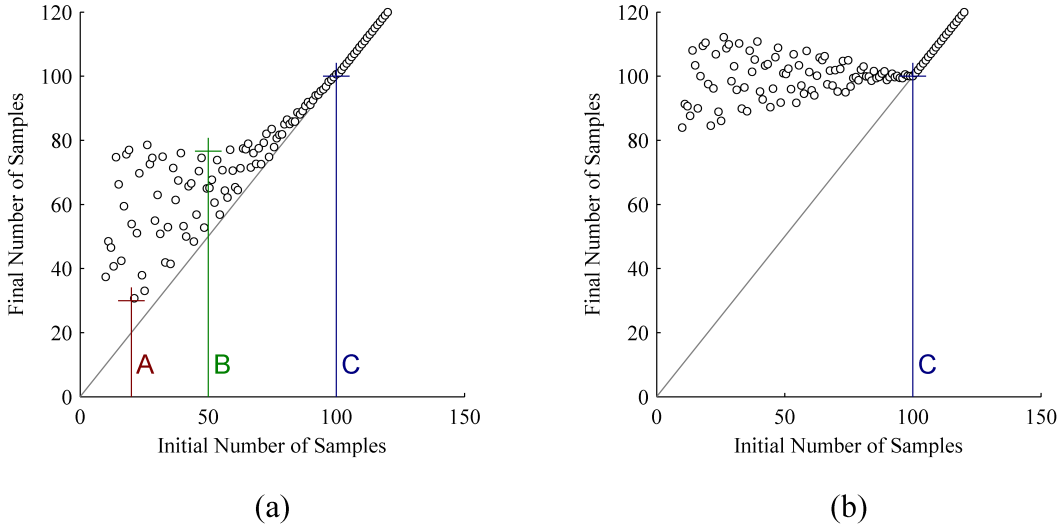


Figure 55: Notional initial and final samples for a MOSI method. Here it is assumed that a MOSI method is run until the error drops below a threshold. In (a), MOSI is more efficient: A is the risky minimal warm-start size, B is the safe warm-start size, and C is the size where a DoE is sufficient. In (b), any warm-start smaller than C risks using more samples than the safe DoE size.

In an actual design exercise, there is no way of knowing the true accuracy of the surrogate along the (also unknown) true Pareto frontier. However, with a test function, the true frontier is known. The first experiment aims to find the optimal warm-start size, first for combined-array and then for crossed-array MOSI, for the test problem with 2 design and 2 noise variables. If the optimal size is found to be smaller than would be required for a purely DoE-based approach, as shown conceptually in Figure 55(a), this will also demonstrate that MOSI adaptive sampling methods are more efficient than pure DoE methods. Further, if the samples required for a C-MOSI method are lower than for the other three methods, this will answer Research Question 3, re-printed here:

Research Question 3: Is there a design scenario where a combined array Multi-Objective Statistical Improvement method (C-MOSI) out-performs both crossed-array and design of experiments methods in terms of efficiency?

As will be shown, C-MOSI was found to be encouragingly efficient for the test problem, but with important limitations regarding ill-conditioning of Gaussian Process surrogate models.

The same tests can be run for different numbers of design and noise variables, and the sensitivity of the four methods found from the resulting data. This will comprise the second experiment.

9.1 Experimental Assumptions and Details

Before the experiments themselves are discussed, this section will present the experimental details and assumptions common to both experiments.

9.1.1 Gaussian Process Model Simplification

Gaussian Process models were used, because they allowed Second Order Probabilities (SOPs) to be calculated analytically [4][88]. In order to allow for greater numbers of test cases, the Gaussian Process model was "pre-tuned". The correlation vector θ is normally optimized, so that each θ_i best represents the degree of correlation along dimension i. Since the test problem is already well-known, some degree of pre-optimization is possible, to reduce the computational effort required in finding this optimal θ vector. It is known that all design dimensions are identical to each other, as are all noise dimensions; the values of θ_i were therefore constrained to be equal to a single value θ_D for all design dimensions, and also to be equal to a single value θ_S for all noise dimensions. Furthermore, the relative values of θ_D and θ_S were found to largely obey the relationship:

$$\frac{1}{\sqrt{\theta_S}} = 11.25\frac{1}{\sqrt{\theta_D}} \tag{121}$$

Where the term $1/\sqrt{\theta_i}$ has the same units as the input space, and can be understood as a sort of "width" parameter. Once this relationship had been fixed, the optimization of θ

became a single-objective optimization exercise, and could be done quickly with a golden-section line search.

Additionally, rather than optimizing θ_D directly, the optimization was performed on a transformed variable w:

$$w = ln\left(\frac{1}{\sqrt{\theta_D}}\right) \tag{122}$$

which can be understood as the logarithm of a "width" parameter. In the optimization step, this is allowed to vary between -3 and 3.

9.1.2 Gaussian Process Model Initialization

As is described in Appendix A, some of the SOP calculations do not depend on the design variables, and can be computed once when the GP is fitted. These are performed as an initialization step. The final SOP calculations are performed within already expensive optimization loops, so it is important not to perform unnecessary computations.

9.1.3 Pseudo-VaR

In order to allow for greater numbers of test cases, second-order probabilities were computed using O'Hagan's analytical method [88]. Because analytical expressions have not been derived for percentile directly, Apley *et al.*'s metric was used instead, and is referred to here as *pseudo-VaR* or *pVaR*. The statistics of interest, then, computed using a combination of O'Hagan's and Apley *et al.*'s analytical second order probability equations, were aleatory mean:

$$\mu_a \sim \mathcal{N}(E[\mu], Var[\mu]) \tag{123}$$

and pseudo-value-at-risk:

$$\text{pVaR} = \mu + c \cdot \sigma \tag{124}$$

$$\text{pVaR} \sim \mathcal{N}(E[\text{pVaR}], \text{Var}[\text{pVaR}]) \tag{125}$$

$$E[\text{pVaR}] = E[\mu] + c \cdot E[\sigma] \tag{126}$$

$$\text{Var}[\text{pVaR}] = \text{Var}[\mu] + c^2 \cdot \text{Var}[\sigma] + 2 * c \cdot \text{Cov}[\mu, \sigma] \tag{127}$$

Where the values $E[\mu]$, $\mathrm{Var}[\mu]$, $E[\sigma]$, $\mathrm{Var}[\sigma]$, and $\mathrm{Cov}[\mu, \sigma]$ are all second-order probabilities computed using O'Hagan's equations [88] and Apley *et al.*'s equations [4]. Details can be found in Appendix **??**. The value of c was set to that corresponding to a 95% VaR, assuming a standard normal distribution, or approximately 1.645.

9.1.4 Definition of Error

The goal of a Pareto-frontier-based robust design method is twofold:

- Predict with accuracy which designs lie on the Pareto frontier
- Predict with accuracy the mean and risk of designs that are on the Pareto frontier

The first of these requires that the predicted frontier lie close to the true frontier in design space. To quantify the first goal, a predicted frontier would need to be determined through optimization, and this frontier would need to be compared to the true frontier. This is not an easy comparison to make, since the frontiers are multidimensional surfaces or curves passing through higher-dimensional spaces, and it requires defining the true frontier as continuous rather than merely identifying discrete points. The second goal is easier to quantify, and is the chosen metric of goodness.

The true Pareto frontier in mean/pVaR space is known analytically. To quantify the accuracy of the model, a set of 20 design points were selected, evenly spaced along the two sections of the frontier. The test function's true aleatory mean ($\mu_{\text{true}}^{(i)}$) and pseudo-VaR ($\text{pVaR}_{\text{true}}^{(i)}$) at each point i was found and stored.

The surrogate's predicted μ_μ and μ_{pVaR} were found at these points, and the root mean square error was found:

$$RMSE = \sqrt{\frac{1}{2}(MSE_\mu + MSE_{\text{pVaR}})} \tag{128}$$

$$MSE_\mu = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mu_{\text{true}}^{(i)} - E[\mu^{(i)}]}{R_\mu}\right)^2 \tag{129}$$

$$MSE_{\text{pVaR}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\text{pVaR}_{\text{true}}^{(i)} - E[\text{pVaR}^{(i)}]}{R_{\text{pVaR}}}\right)^2 \tag{130}$$

Where R_μ and R_{pVaR} are the range of each statistic in objective space, used to normalize the response. These were taken to be the range on the Pareto frontier, rather than the

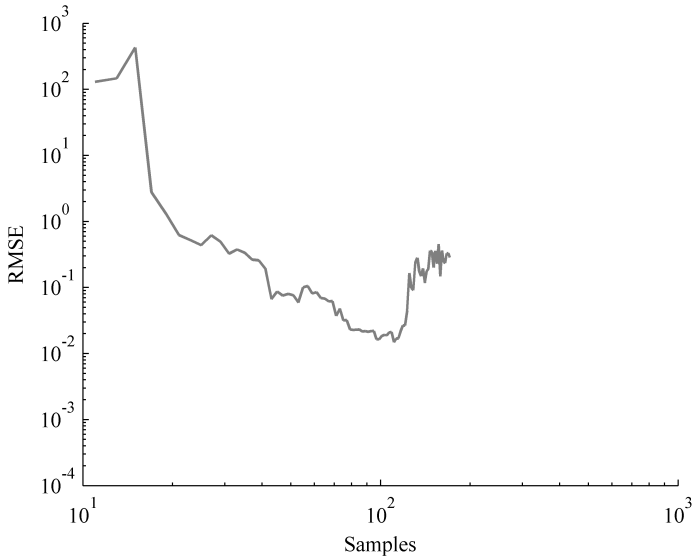


Figure 56: A C-MOSI run's error progression, test problem with $p_D = p_S = 2$. The accuracy of the model progressively improves until around 100 samples are reached, after which it quickly gets worse. This is due to ill-conditioning effects in the GP model.

range over all of objective space, since the Pareto frontier was the range of interest. For the test problem, this range was 1.0 in both cases.

9.1.5 A Note on Ill-Conditioning of the Covariance Matrix

For many of the combined-array MOSI runs, the RMSE along the frontier declined as samples were gradually added to the surrogate, but then began to increase again. A sample path for such a run is shown in Figure 56.

Why should the surrogate's accuracy get worse as information is added? It is because the covariance matrix becomes ill-conditioned. The condition numbers for even small numbers of samples are on the order of 10^6, and as the condition number approaches 10^{10} the error quite suddenly stops getting lower and instead spikes up. The ill-conditioning problem is a well-known issue with Gaussian Process models [57][60].

The exact condition number at which error began to rise was not always predictable, however. The condition number at the point of minimum error varied from 10^7 all the way up to around 10^{11}. Because of the wide range of "critical" condition numbers, condition number by itself was not a useful stopping criteria. Stopping criteria will be discussed in a later section.

9.1.6 Algorithm Details, Combined (C) Arrays

This subsection will provide the algorithmic details necessary to replicate the experiments. The details here pertain to the combined-space models, C-DoE and C-MOSI.

9.1.6.1 Warm-Start Design of Experiments

An initial Latin Hypercube Sampling (LHS) Design of Experiments was created using the built-in MATLAB function *lhsdesign*, with the "correlation" criteria (which minimizes the correlation between the columns of the design). There are algorithms available to create more optimal LHS designs for Gaussian Process models, such as presented by Forrester *et al.* [39][38], but due to the high computational cost and the large number of experiments required, the relatively inexpensive MATLAB function was used instead. This might result in some degradation of performance relative to what is possible, and therefore any declarations with regard to the relative merits must be qualified. The *lhsdesign* function is random; it generates random designs, and then selects the one that best meets the desired criterium. The default number of iterations, 5, was used. The design variable columns were normalized over the range [0,1], and the noise variables over the range [-3,3].

In the case of the C-MOSI runs, this initial sample served as the warm-start design; in the C-DoE runs, it constituted the whole sample set. Once the initial points were selected, a GP model was fit according to the procedure described previously. The quality of the initial fit, and how well that initial fit allows accurate prediction along the true Pareto frontier, was highly dependent on the initial sample, so the RMSE of the warm-start DoE varied a great deal from one LHS sample to the next. Repeated trials were therefore very important.

9.1.6.2 Optimization of Design Sample, C-MOSI

Once a GP had been fit to the initial warm-start LHS design and the SOP calculations have been initialized, optimization was used to find the design D^* with the greatest expected Pareto improvement.

First, the current Pareto set was found. The current set of sampled designs were used as

candidates, as described in section 6.3, and SOP analysis was performed at all designs. The epistemic expected value of the mean and pseudo-VaR were found, as well as the epistemic standard deviation for both parameters, using the procedure described in Appendix A

Then, a genetic algorithm was used to find a candidate design with the greatest exected improvement over the Pareto set. The built-in MATLAB genetic algorithm function *ga* was used, with a population of 20, for a maximum of 100 generations, but otherwise at default settings. At each design, the expected Pareto improvement was calculated with the modified Emmerich *et al.* method described in section 6.3. Note that the computation time here was dominated by a single term in the computation of SOPs, as described in section 6.6, which scaled as $O(n^3)$.

9.1.6.3 Optimization of Noise Samples, I-SOP

Once the design D^* was found, optimization over the noise space was used to find two samples, S^*_μ and S^*_ρ. For this, the I-SOP method was used, as described in section 6.4.3.1. For every candidate noise sample, the epistemic mean predictive value was found from the GP model, and this is point was imputed. The GP θ correlation parameters were *not* re-computed, but the covariance matrix A *was*, as well as those parts of the SOP initialization step that depended on it. Then the SOP analysis was performed at D^*, and the epistemic variance on the aleatory mean (σ^2_μ or Var$[M]$) was found. A genetic algorithm with a population of 20 and a maximum of 100 generations was used, again with the built-in MATLAB function *ga*. It was important in this case to provide the function with a random population over the search area of interest (which was [-3,3] for each noise dimension), or else the function would generate its own population over the range [0,1]. All other settings for *ga* were used at their defaults.

The point S^*_μ was selected which minimized the imputed σ^2_μ. This point $[D^*, S^*_\mu]$ was then sampled, and added to the GP, which was completely re-fit.

The procedure was then repeated to find the sample $[D^*, S^*_\rho]$ that minimized the epistemic variance of the pseudo-VaR, a term which is referred to as $\sigma^2_f(d)$ in Appendix A. Once this point was found, it was sampled, the GP was completely re-fit, and the algorithm

returned to selecting a new D^*.

9.1.7 Algorithm Details, Crossed (X) Arrays

For the crossed-array methods, X-MOSI and X-DoE, many of the algorithmic details were similar to those used in the combined-array methods.

9.1.7.1 Warm-Start Design of Experiments

As with the combined-array methods, an initial LHS warm-start DoE was generated using MATLAB's built-in *lhsdesign* function, with the selection criteria set to "correlation" and 5 iterations. However, this design was only over the design variables.

9.1.7.2 Sampling in Noise Space, I-SOP

For every design, a separate sampling method was used over noise space. First, another warm-start LHS DoE was used. For a given number of noise variables, this DoE was of fixed size, $p_S + 7$. Additionally, the same initial warm-start population was used for every design, to make the noise sampling more consistent, and this initial warm-start population was chosen with some care. Rather than using the built-in MATLAB *lhsdesign* function, Keane and Forrester's code for generating optimized LHS designs was used [**?**].

A GP model was fit to the noise space, and I-SOP was used to find successive S^*_μ and S^*_ρ samples. These were sampled and added to the GP model, exactly as in the C-MOSI procedure, with the same genetic algorithm procedure. Unlike in C-MOSI, however, the design varibale settings did not change. I-SOP was continued until one of the following conditions were met:

- The epistemic standard deviation of both the mean and pseudo-VaR were below 0.005
- The condition number of the GP covariance array A climbed above 10^{10}
- Adding a new data point to the GP failed

At this point, the epistemic means of the aleatory mean and aleatory pseudo-VaR were returned. The number of samples required to estimate both statistics was tracked.

9.1.7.3 Fitting the Gaussian Process Model, Crossed (X) Arrays

Once I-SOP had been used to estimate aleatory mean and pseudo-VaR for all designs, two separate GP models were fit, one to each. In the case of X-DoE, this was the last step. In the case of X-MOSI, the two GP models were updated as additional designs were sampled.

9.1.7.4 Sampling in Design Space, X-MOSI

After the initial two GP models had been fit, one to mean and one to pseudo-VaR, additional samples were selected using the expected Pareto improvement criteria. Emmerich *et al.*'s formula was used, in this case without modification, as the results from the I-SOP step were assumed to be free of epistemic uncertainty. This was an approximation; in reality, there was still epistemic uncertainty present, even though it was low (less than 0.005 for both mean and pseudo-VaR, when the Pareto frontier had a range of 1.0 over both). However, there was no purpose to considering epistemic uncertainty when sampling in design space; previously sampled designs were completely independent from any new samples, so no further reduction in epistemic uncertainty was possible.

The point of greatest expected Pareto improvement was again found using the built-in MATLAB *ga* function, again with a population of 20 for a maximum of 100 generations.

9.2 Experiment: Warm-Start Size

In the first experiment, the number of design and noise variables were both fixed at two. The warm-start DoE sizes were swept from low to high. The lower limit on DoE size was set by the ability to calculate Second-Order Probabilities, which required at least $p+6$ samples (where p is the number of input dimensions). The upper limit was set as somewhat past the point where ill-conditioning of the GP covariance matrix caused poor accuracy for the warm-start DoEs.

9.2.1 Combined Array DoE and MOSI

A set of fifty C-MOSI runs were executed for the test problem with two design and two noise variables. The progression of their model errors is shown in Figure 57, and their final vs. initial sample sizes are shown in Figure 58.

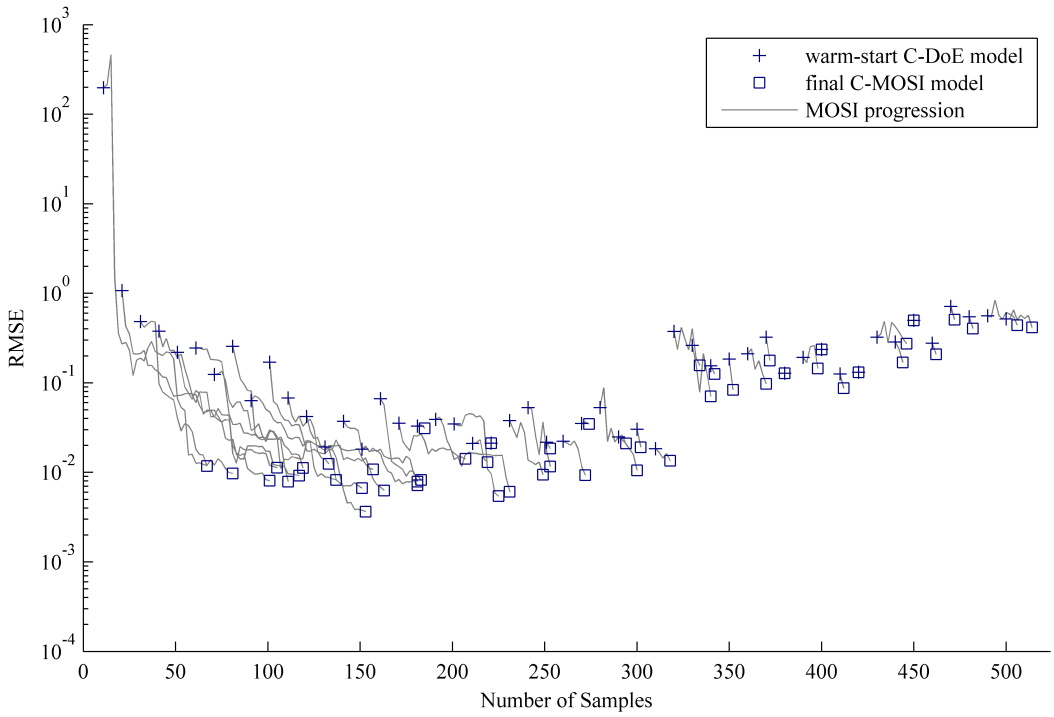


Figure 57: A set of C-MOSI paths, from initial warm-start DoEs (+) to the point of minimum error (□). Past about 150 to 200 samples, ill-conditioning effects take hold. Prior to that, both the endpoints and paths of the C-MOSI runs tend to dominate the DoE samples in terms of error and number of samples. Several of the C-MOSI runs do degrade in accuracy initially before beginning to improve.

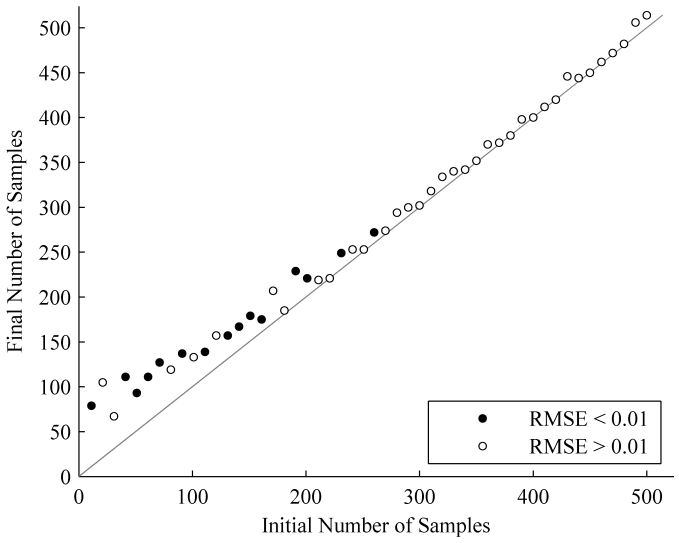


Figure 58: The initial and final sample sizes for the same set of runs as is shown in 57. Dark circles achieved a target RMSE of 0.01 or below, while white circles did not. There is a significant amount of randomness as to whether the runs reached the target or not, but there does not seem to be an "optimum" warm-start size other than the minimum allowable.

In 57, runs are shown in terms of number of samples and RMSE. Warm-start DoEs are shown as '+' markers, and the progression of error with samples is shown as a path. The square markers represent the point in the run of minimum RMSE. Note at the smallest warm-start size, the error increased for a few samples before decreasing. This effect will be more pronounced for higher-dimensional problems, and will be discussed in a later section. Note also that above about 200 samples, C-MOSI began to do almost as poorly as C-DoE, and above about 300 samples, both the DoE and MOSI methods performed quite poorly. This was due to the conditioning problems discussed earlier.

In the same figure, the initial DoE-based models and the final MOSI models can be viewed as separate populations, in terms of their number of samples and RMSE. Over all sample sizes, the MOSI method results in lower RMSE, with a small amount of overlap. Thus, at least in this test case, C-MOSI is usually (though not always) more efficient than C-DoE.

This provides a partial answer to Research Question #3:

Between Combined-array Design of Experiments (C-DoE) and Combined-array Multi-Objective Expected Improvement (C-MOSI), for the scalable test problem with 2 design and 2 noise variables, C-MOSI is usually more efficient.

In Figure 58, it has been assumed that the designer had a target RMSE of 0.01. For cases that reached the target (dark circles), the vertical axis represents the number of samples required. In cases that did not reach an RMSE of 0.01 (white circles), the vertical axis represents the number of samples when minimum error was reached. From this figure, it should be possible to determine the optimum warm-start size, as was supposed in Figure 55 from the beginning of this chapter. Perhaps surprisingly, the optimum appears to occur at the minimum warm-start size. Even with an initial DoE with only 10 samples, the C-MOSI algorithm was able to efficiently model the true Pareto frontier. Note that Sobester *et al.* found that optimal warm-start size was problem-dependent, and in their test cases the optimal warm-start size was occasionally larger than the minimum.

9.2.1.1 A Stopping Criteria

Due to the ill-conditioning problems discussed previously, if the C-MOSI method is run long enough, the error will eventually rise. The onset of this error rise is relatively sudden, and if left to run the MOSI method will quickly end up performing more poorly than a DoE method with a comparable number of samples. In these tests, the point of minimum error or a target error is used, but in practice error is not known exogenously. It becomes a very real practical concern to determine at which point the run should be stopped. Fortunately, as true error rises, the epistemic uncertainty estimate also begins to rise. Similar to the RMSE measure, a measure of total epistemic uncertainty can be adopted, among the designs

thought to be on the frontier:

$$\sigma_{\text{total}} = \sqrt{\frac{1}{2}(\sigma_{\mu}^2 + \sigma_{\text{pVaR}}^2)} \tag{131}$$

$$\sigma_{\mu}^2 = \frac{1}{N_P} \sum_{i \in P} \left(\frac{\text{Var}[\mu]^{(i)}}{R_{\mu}^2} \right) \tag{132}$$

$$\sigma_{\text{pVaR}}^2 = \frac{1}{N_P} \sum_{i \in P} \left(\frac{\text{Var}[\text{pVaR}]^{(i)}}{R_{\text{pVaR}}^2} \right) \tag{133}$$

Where here σ_{μ}^2 and σ_{pVaR}^2 refer to the *total* epistemic variance, and i indexes along the N_P designs currently thought to make up the Pareto set P. The ranges R_{μ} and R_{pVaR} are, as before, taken to be the ranges of the true Pareto set, and for this test problem are both 1.0.

Figure 59 shows how σ_{total} predicts RMSE. For lower numbers of samples ($N < 100$), it is a very good predictor, with an $R^2 = 0.9$. After the ill-conditioning problems arise, σ_{total} tends to over-predict the RMSE. It is unsurprising that the model's prediction of its own error becomes less accurate as the model itself becomes less accurate. What is perhaps surprising is that the model predicts an increase in error at all. That is does could be considered more a numerical artifact than anything. The epistemic variance terms are of higher order than the expectation terms, and are constrained to be positive. Numerical noise will tend to push them larger, and will change them faster than the expectation terms. Thus, it is not that the model is "correctly" predicting that its error is higher, it is simply that the errors in the model cause the *estimates* of the error to diverge... and this divergence will tend to be positive.

Figure 60 shows a scatter plot of when these two metrics are minimized, in terms of number of samples. The x-axis shows the number of samples where the minimum RMSE occurred, and the y-axis shows the number of samples where the minimum σ_{total} occurred. The two appear to be in reasonable agreement, though obviously not in perfect agreement. Points below the line represent runs that would be stopped early with the σ_{total} criteria, whereas points above the line would be stopped late. The criteria appears to lead to early stopping more often than late stopping.

In Figure 61, the RMSE vs. samples plot of Figure 57 has been re-plotted, this time using the MOSI cases with lowest epistemic uncertainty, rather than lowest true error. The

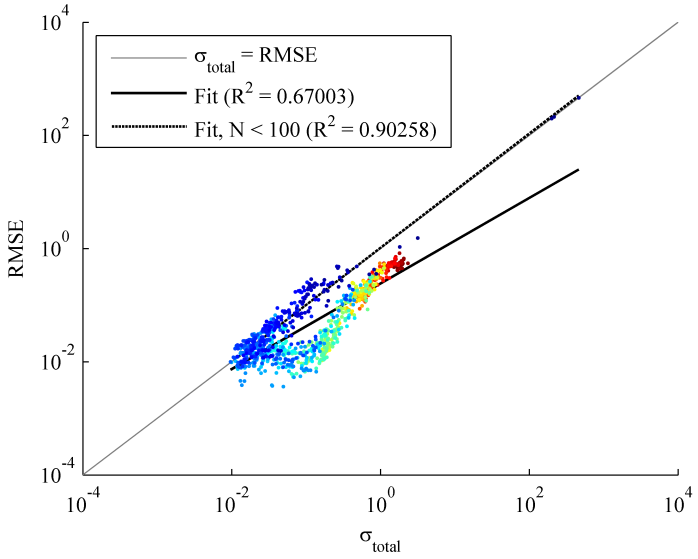


Figure 59: RMSE as predicted by σ_{total}. Colors correspond to number of samples, from low blue to high red. σ_{total} appears to be a very good predictor before the model becomes ill-conditioned ($R^2 = 0.9$), but is biased high over the whole dataset. One point in the graph represents one sample. All C-MOSI runs from this experiment are plotted together.

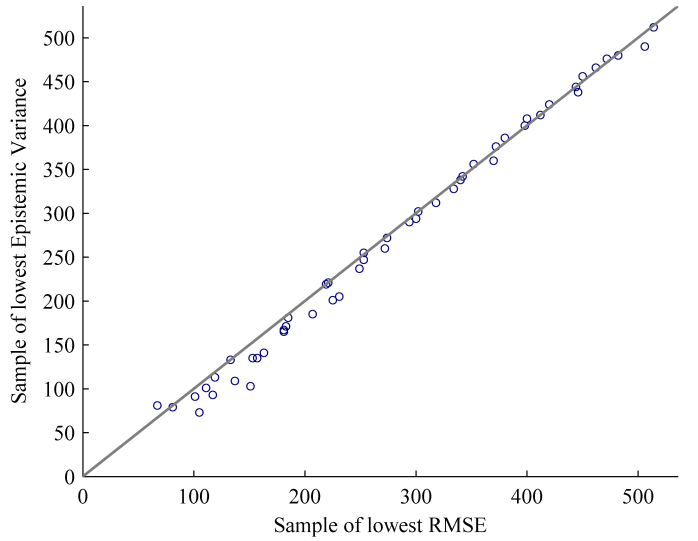


Figure 60: Number of samples to reach min $\mathbf{sigma}_{\text{total}}$ vs. samples to reach min RMSE. The two are strongly, though not perfectly, correlated.

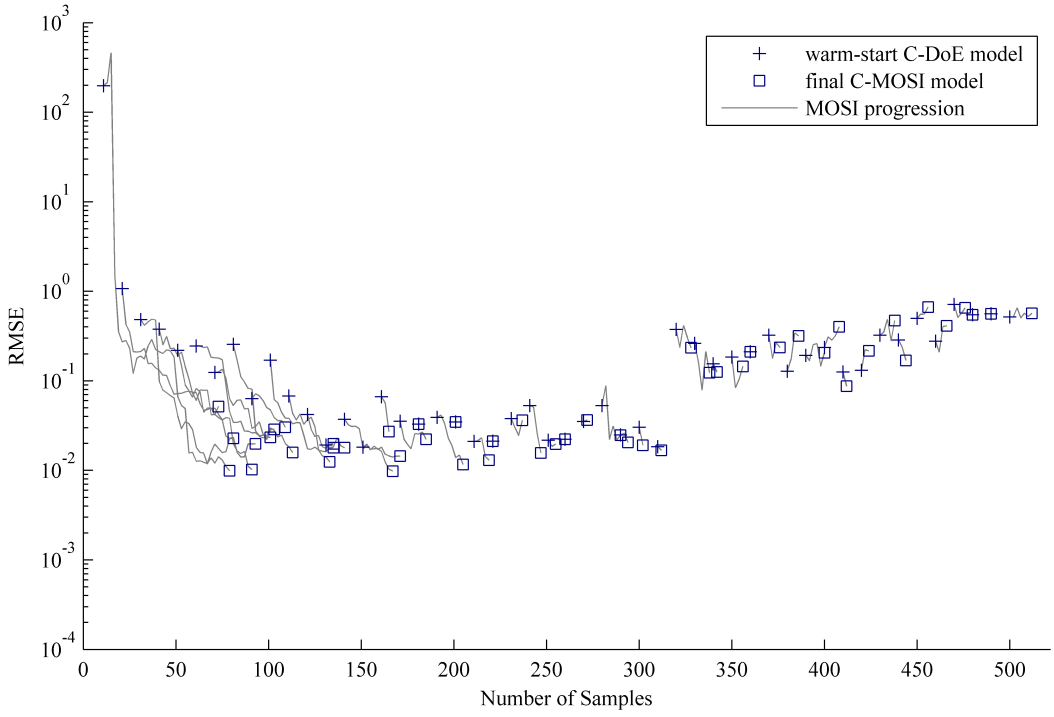


Figure 61: Re-plot of Figure 57, but with runs stopping at point of minimum σ_{total} rather than minimum RMSE. The C-MOSI runs still dominate the DoE runs, though no longer by as much.

C-MOSI runs still dominate the C-DoE runs, though not by nearly as much. The point when σ_{total} stops improving, then, is a workable stopping criteria, though it leaves something to be desired.

9.2.2 Crossed Array DoE and MOSI

In a crossed-array method, there are two separate arrays, one for the noise space and one for the design space. An "inner" array is made in design space. For every design, a separate "outer" array in noise space is used to determine the mean and value-at-risk. The sizing of these arrays were investigated separately at first.

9.2.2.1 Noise Array Warm-start Size

The noise space can be sampled adaptively using the same method as was used to select samples in noise space in the combined-array method, by imputing candidate points and finding the sample that gives the greatest most-likely reduction in epistemic uncertainty (the I-SOP method described in Chapter 6). Again, this requires a warm-start DoE, this time only over noise space, and this warm-start DoE can consist of any fraction of the final set of samples. A sweep of noise space warm-start size was performed for one single design ($D = (0.5, 0.5)$), and the results are shown in Figure 62. In this case, RMSE was measured as simply:

$$\text{RMSE} = \sqrt{\frac{1}{2}\left((E[\mu] - \mu_{true})^2 + (E[\text{pVaR}] - \text{pVaR}_{true})^2\right)} \tag{134}$$

Where again pVaR is the pseudo-value-at-risk ($\mu + c \cdot \sigma$), and the expected values of statistics are found by finding the second-order-probabilities analytically from a Gaussian Process model.

A warm-start DoE was required to contain at least $p_S + 6 = 8$ samples, because the variance of the variance is un-defined with fewer samples. At least for the (admittedly low-dimensional) case of 2 noise variables, using adaptive sampling did not appear to offer much advantage over a fixed DoE. For a target RMSE of 0.007, a warm-start of only 10 samples was sufficient without any further sampling.

However, since true error is well-predicted by second-order probabilities (as shown in

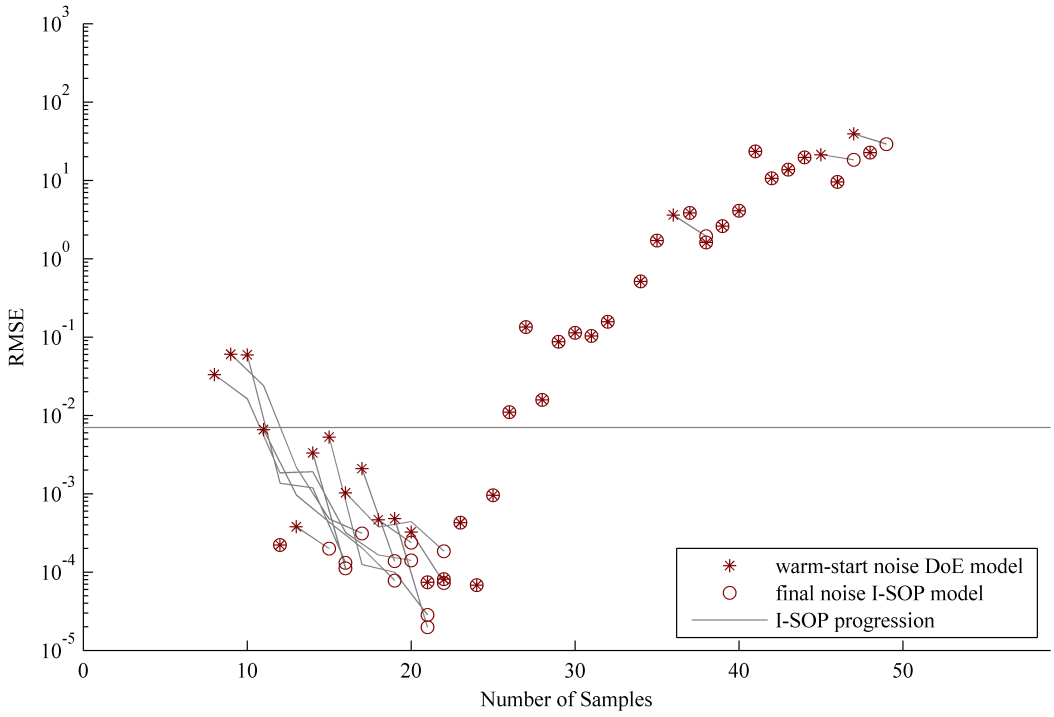


Figure 62: Imputation-based Crossed-array Second Order Probability sampling in noise space (X-I-SOP). This is only a two-dimensional space, and ill-conditioning leads to inaccuracy after only 20 or so samples. Up until that point, additional DoE and I-SOP samples seem to improve the accuracy by about the same amount.

the previous section), using I-SOP provides the opportunity to refine a model, if the initial DoE is estimated to not be sufficiently accurate. In economic terms, I-SOP provides value by providing an option to refine the model. This means that the designer does not need to spend more samples than necessary at any given design, and in the end this could lead to greater efficiency. In practice, it was found that using a fixed noise DoE rather than I-SOP did not improve the *initial* accuracy of the model (see Figure 63), but that it *did* result in occasional instances of very inaccurate statistics at individual designs. Since the design-space surrogate assumed perfect accuracy in the noise-space models, this meant that from then on the model was "cursed" with a bad data point, and could never improve beyond a certain level of accuracy. This is actually a fundamental issue with combining crossed arrays and adaptive sampling. It was addressed by Kumar [67] through the use of GP models with a "nugget" that allow for non-zero uncertainty at sampled data points. If this approach were adopted, the sub-space models could be saved and periodically re-sampled to improve their accuracy, at the expense of slightly higher overhead and storage requirements.

For the next test, the noise subspace warm-start size was set at the minimum, and adaptive sampling was used to find the mean and value-at-risk for every design.

9.2.2.2 Design Array Warm-start Size

The effects of varying warm-start size of the design-space surrogate were then investigated. A Latin Hypercube DoE was constructed in design space. At every design point, another Latin Hypercube of 9 points was created in noise space, and adaptively sampled until epistemic uncertainty in both mean and value-at-risk fell below 0.005. Note that at this point, this was a *hybrid method*, combining adaptive sampling in noise space with a fixed DoE in design space.

Once the mean and VaR had been estimated to satisfactory accuracy at all designs, two GP surrogates were fit, one to mean and one to VaR. These were used in combination with Emmerich *et al.*'s hypervolume-based MOSI method to select a new design. This design was then sampled in noise space until the accuracy of the mean and VaR were acceptable, and the process was repeated.

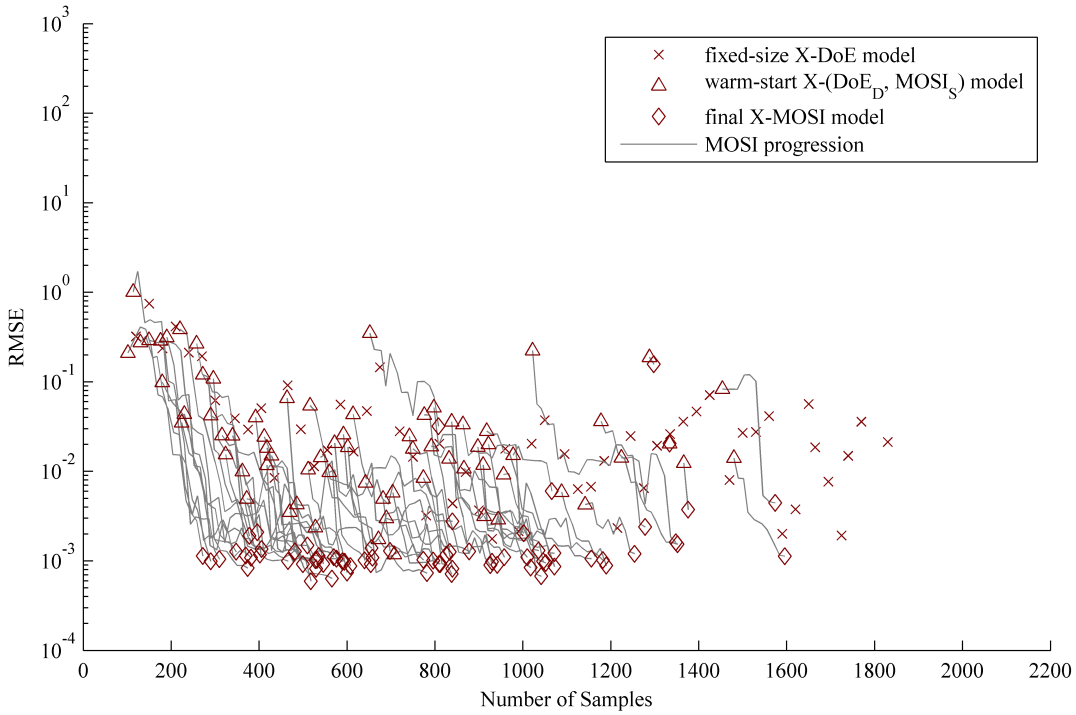


Figure 63: A set of X-MOSI paths. The triangles represent warm-start populations, and the gray paths are X-MOSI progressions, ending in diamonds. The triangles represent design-space X-DoEs, but in noise space they use X-I-SOP. The 'x's represent pure X-DoEs. X-MOSI sampling in design space dominates design space DoEs.

Plots of the paths taken by individual runs of the method are shown in Figure 63. The triangles represent the initial warm-start X-Doe (hybrid method, adaptive I-SOP sampling on noise space), and the gray lines represent X-MOSI sampling paths from those warm-start DoEs to the point of maximum accuracy. As with C-MOSI, the X-MOSI paths and endpoints dominate the X-DoE runs in terms of efficiency. The '+' symbols represent a totally separate set of X-DoE-only runs, which do not seem to differ much from the warm-start X-DoE's with I-SOP.

A plot of initial and final sample size is shown in Figure 64. As with C-MOSI, the optimal warm-start size appears to be at the lower limit of the range. This confirms that for this low-dimensional design problem at least, X-MOSI is more efficient than X-DoE.

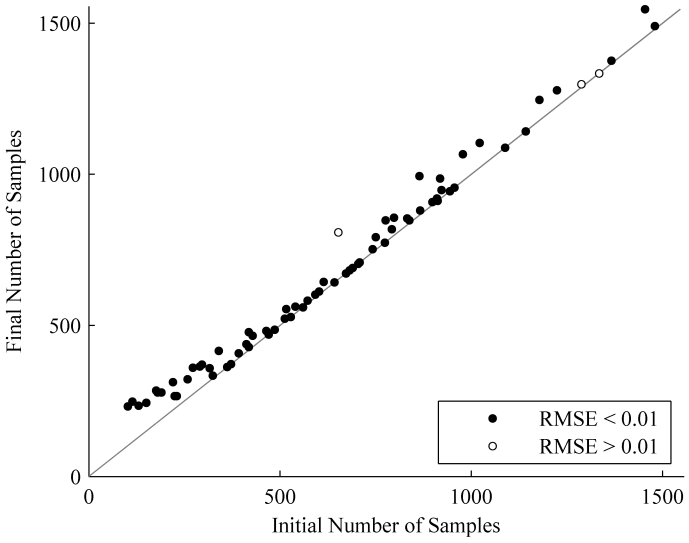


Figure 64: X-MOSI initial and final sample sizes. Solid circles achieved an RMSE below 0.01, white circles did not. The optimum warm-start size, as with C-MOSI, appears to be at the lower limit, confirming that X-MOSI is more efficient than X-DoE.

9.2.3 Comparing All Four Methods

In Figure 65, all four methods are shown. This graph is similar to the previous "path" charts, but the paths have been omitted and each method has been presented simply as its own population. Rather than exhibiting clear dominance, a frontier emerges in the trade between error and samples. Keep in mind that even though paths are not shown, the left-most paths of the MOSI methods do dominate the DoE methods, as seen in previous plots. Though the DoE methods are dominated by the MOSI methods, between the two MOSI methods there is a trade: C-MOSI dominates for low numbers of samples, but for lower error the C-MOSI method cannot compete with X-MOSI.

One of the objectives of this experiment was to answer the research question:

> **Research Question 3:** Is there a design scenario where a combined array Multi-Objective Statistical Improvement method (C-MOSI) out-performs both crossed-array and design of experiments methods in terms of efficiency?

This cannot be definitively answered yet. At low numbers of samples, and higher allowed

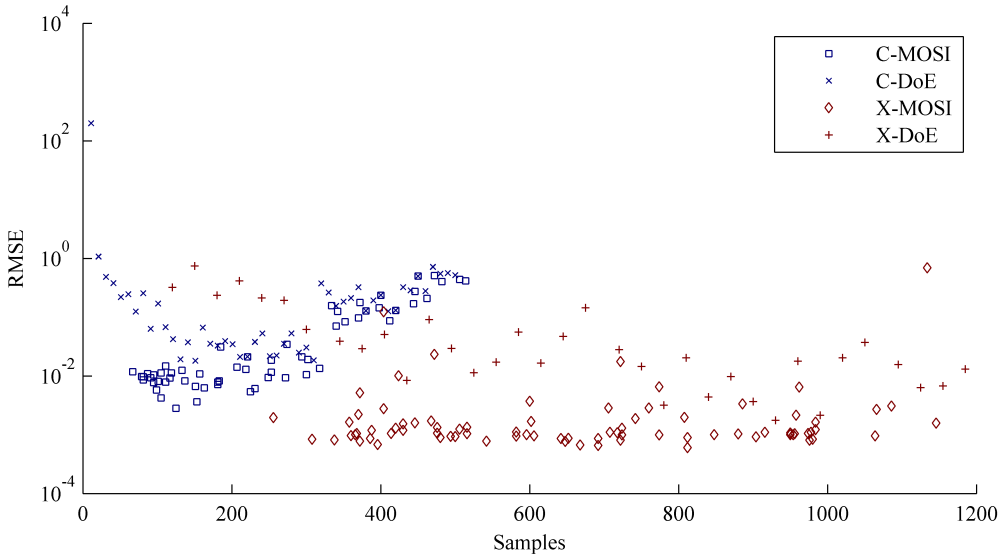


Figure 65: All four methods on the same axes. The C-MOSI and X-MOSI methods both dominate, C-MOSI for lower numbers of samples but higher error, and X-MOSI for lower error but more samples.

error, C-MOSI dominated; but for stricter accuracy requirements, X-MOSI was equally "efficient" in the Pareto sense. More comparisons will be made in the next experiment.

9.3 Sensitivity to Problem Dimensionality

Two research questions involved the sensitivity of different method types to the dimensionality of the problem. They are reprinted here, along with the corresponding hypothesis, as first stated in Chapter 5.

Research Question 1: For finding mean/risk Pareto frontiers, how does the relative efficiency of combined and crossed arrays depend on the number of noise variables?
Hypothesis 1: As the number of noise variables increases, the efficiency of combined array methods will suffer relative to the efficiency of crossed array methods.

Research Question 2: For finding mean/risk Pareto frontiers, how does the relative efficiency of design of experiments and multi-objective statistical improvement change with the number of design variables?

Hypothesis 2: As the number of design variables increases, multi-objective statistical improvement methods will become more efficient relative to a design of experiments.

These questions and hypothesis make reference to "efficiency", which has so far been imprecisely defined as the Pareto efficiency with regard to number of samples and model accuracy. In Figure 65, the error along the Pareto frontier was plotted for the four methods as a function of number of samples, with poorly conditioned models removed. In Figure 66, the MOSI methods have been re-run ten times at their minimum warm-start size, and their error progressions have been plotted alongside the previously-plotted DoE data. Each population now represents the progression of error as samples are added for each method. The data has been plotted on log-log axes, and each population appears linear, suggesting a power relationship. Linear fits and their R^2 values are shown. The points of intersection between DoE and MOSI runs can be clearly seen; the data once again show the MOSI methods to be more efficient, and these intersection points are a graphical representation of the "crossover point", where the MOSI method becomes more efficient. Note that this is not the same as optimal warm-start size; all MOSI runs are run at the minimum warm start size, but the C-MOSI runs get worse before they get better. For the crossed-array methods, the crossover point appears to be immediate.

The linear fits shown in Figure 66 can be thought of as models for the expected RMSE for each method as a function of sample size. In the next subsection, these models will be refined and developed further. Then, in the following subsection, they will be used to test the sensitivity of the different methods to the dimensionality of the problem.

9.3.1 Power Function Error Models

Since the test problem is known, there is an unusual opportunity not usually found in statistical sampling problems. It is possible to quantify how inaccurate any sampling method is expected to be, given a certain number of samples, by repeatedly running the method, as has been done in the previous section.

To determine how problem dimensionality affects method performance, it will be useful

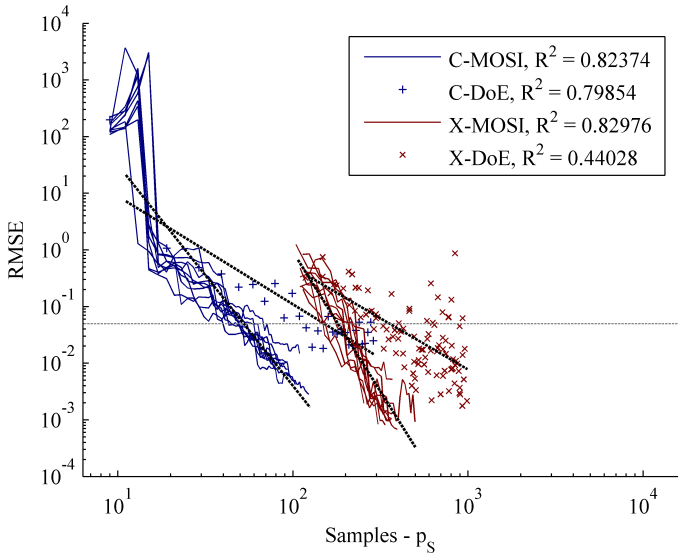


Figure 66: Log-log plot of RMSE vs. samples for the four methods, with 2 design and 2 noise variables. Both MOSI methods have been run 10 times at the smallest warm-start size. Data for DoEs is as in Figure 65. Dotted black lines are linear regressions, with R^2 values shown in the legend. Ill-conditioned models have been removed.

to have a model for how accurate the method is expected to be for a given number of samples. The previously-discussed Figure 66 suggested a power relationship for the predicted root mean square error, which will be called $\widehat{\text{RMSE}}$:

$$\log_{10}(\widehat{\text{RMSE}}(N)) = a\log_{10}(N) + b \tag{135}$$

$$\log_{10}(\widehat{\text{RMSE}}(N)) = -\alpha\log_{10}(N) + \log_{10}(\widehat{\text{RMSE}}_{N=1}) \tag{136}$$

$$\widehat{\text{RMSE}}(N) = \widehat{\text{RMSE}}_{N=1}N^{-\alpha} \tag{137}$$

where N is the number of samples. The coefficient α is a measure of how quickly the method is able to reduce error, and since the methods will be expected to reduce error, a negative sign is added to the exponent so that the coefficient will be positive, with higher values corresponding to more efficient methods. The term $\widehat{\text{RMSE}}_{N=1}$ is the predicted RMSE when the method is run for only a single sample.

However, it is not necessarily possible to run a method for a single sample. Each method has some minimum number of samples below which the method cannot be run. This limit may be due to the requirements for calculating the aleatory statistics, or it may be structural. What's more, this value of $\widehat{\text{RMSE}}_{N=1}$ may be very large, over 10^{100}. This is

fine as long as it is kept in logarithm form as the intercept of the linear regression model, $b = log_{10}(\widehat{\text{RMSE}}_{N=1})$, but it causes numerical problems if it is ever calculated directly. To remedy this, revised expressions will be used, of the form:

$$\widehat{\text{RMSE}}(N) = \left(\frac{0.1}{\hat{N}_{0.1}^{-\alpha}}\right) N^{-\alpha} \tag{138}$$

where now $\hat{N}_{0.1}$ is the number of samples where the method is predicted to have an RMSE of 0.1. From Figure 66, it can be seen that all four methods at some point pass through an RMSE of 0.1, so it is always within the regression range, and what's more it has a physical meaning that is potentially useful. It will always have a reasonable value, and can be found from the coefficients of the regression model:

$$\hat{N}_{0.1} = 10^{\left(\frac{1+b}{\alpha}\right)} \tag{139}$$

In the remainder of this subsection, still for a problem with $p_D = p_S = 2$, full error models $\widehat{\text{RMSE}}(N)$ will be found for each method. These models will provide a more complete quantification of method "efficiency". Finally, they will be combined into a unified error model, which can be used to measure the sensitivity of the methods to problem dimensionality and to answer Research Questions 1 and 2.

9.3.1.1 Combined-Array Error Models

As seen in Figure 66, the C-DoE and C-MOSI methods intersect at the low end of their ranges, as would be expected, since the C-MOSI method relies on a C-DoE warm-start. What's more, the two methods seem to overlap in the nonlinear section where samples are low. Both methods begin at the same error level for the minimum warm-start size, and both increase in error briefly before quickly dropping in error. Whether this is a fundamental property of combined-array methods or simply a property of the test problem is unclear.

The surrogates used in these experiments are Gaussian Process models with priors that include a bias term and a linear term for each input dimension. Thus, $p_D + p_S + 1$ degrees of freedom are used in estimating the parameters in the prior. Another 5 degrees of freedom are required for calculating second order probabilities, meaning that the minimum sample size is $N_{min} = p_D + p_S + 1 + 5$, or 10 for the case of $p_D = p_S = 2$.

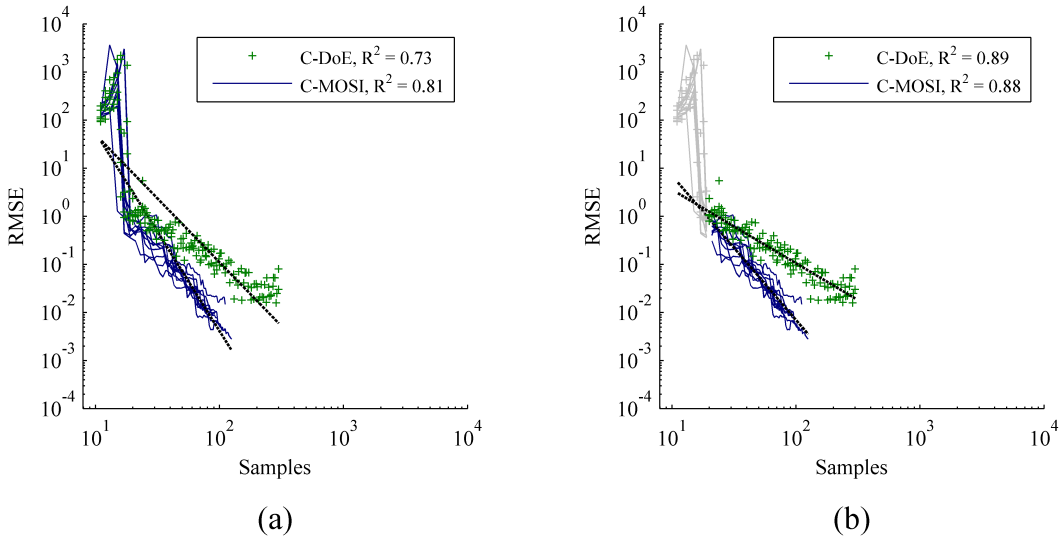


Figure 67: Combined-array error progression and error models, $p_D = p_S = 2$. In (a), only ill-conditioned runs have been removed. The power fit is poor, because of the transient behavior at low sample sizes, both for DoE and MOSI. In (b), all data with $N < 20$ have been removed (shown in gray), and the fit has improved, though the transient is now not captured at all.

However, because of the non-linear behavior for low numbers of samples, all models with fewer than 20 samples were removed from the regression. Figure 67(a) shows the fit lines for the two methods when only ill-conditioned runs have been removed, and (b) shows the fits when all data with fewer than 20 samples have been removed. The R^2 improves somewhat (0.81 to 0.88 for C-MOSI, 0.73 to 0.89 for C-DoE), and it can be seen that the behavior after the transient is better-captured. However, the method behavior at and before the transient is not captured at all. Since the primary concern is the effectiveness of the methods for realistic numbers of samples, this is acceptable. Values of α and $\hat{N}_{0.1}$ will be shown later in the section, along with those for all methods and problem sizes.

9.3.1.2 Crossed-Array Error Models

For crossed-array methods, the fit was not as challenging. Ill-conditioned runs were removed, and the remaining data appeared linear in log-log space. The fits were reasonable, $R^2 = 0.83$ for X-MOSI and $R^2 = 0.71$ for X-DoE. The two fits intersected at $N = 94$ samples, which was close to the smallest warm-start size of 106 for X-MOSI. Specific values for the regression

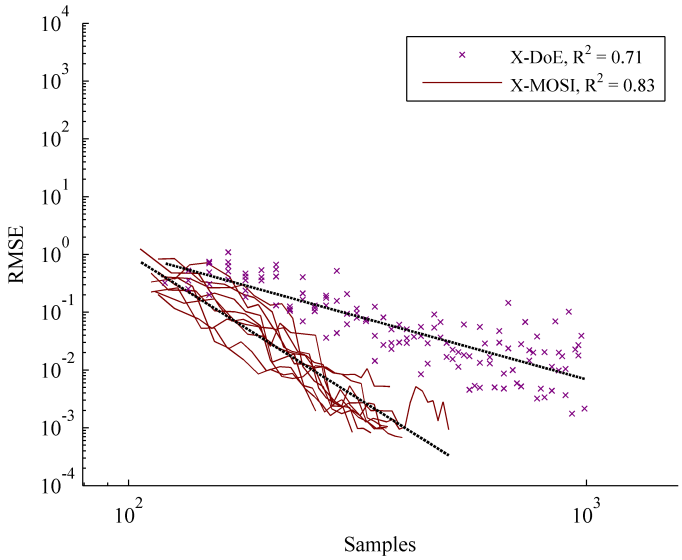


Figure 68: Crossed-array error progression and power error models, $p_D = p_S = 2$. Ill-conditioned models have been removed.

parameters will be presented in the next section for multiple problem sizes.

9.3.2 Experimental Design: Sensitivity to Problem Dimensionality

The methods were again run, three more sets of times, incrementing the number of both design and noise variables by one. The very simple design with regard to numbers of variables is shown in Figure 69. Ideally, the experiment would have extended over a much wider range of problem sizes, but the experiments were constrained by available computer budgets and the curse of dimensionality. Higher dimensional problems require more samples, and Gaussian Process models can be expensive to regress and evaluate for large sample sizes. The largest combined-array samples sizes seen here (9,000 samples in a single GP model for $p_D = p_S = 3$) were already reaching the limit of practicability, as they took about half an hour to regress and evaluate and consumed several Gigabytes of computer memory, even with the GP simplifications referred to earlier in this chapter. Ironically, though the MOSI methods required significant computational overhead to select subsequent samples, they were computationally tractable for higher problem dimensionality since they required fewer total samples, and were successfully used on problems of size $p_D = p_S = 5$. The comparison is perhaps unfair, however, since DoE methods can be used *without* Bayesian surrogates

and GP models were used here only for consistency.

Table 13 shows the number and types of runs used in the experiments, along with the fit parameters. Figure 70 shows RMSE vs. samples plots for all four settings. The two parameters of $\hat{N}_{0.1}$, and α together completely specify the error model. Together, they can be considered a model for how "efficient" the method is in terms of providing information about the Pareto frontier for a particular budget of samples. It is not complete to refer to a single metric for efficiency. If the designer has a particular fixed budget of samples, then RMSE could be used as a single metric; or, conversely, if the designer had a desired level of accuracy, then required samples could be used as a single metric. However, in the absence of such a requirement a Pareto notion of efficiency must be retained.

It may be reasonable, however, to assign more theoretical importance to the α term. If one were to consider a purely hypothetical scenario where any of the methods could be run for an infinite number of samples without ill-conditioning effects, and the goal was to drive error to zero, in the long run the intercept term would not matter. From a sufficiently "zoomed out" perspective, all four error models intersect somewhere in the region of the graphs. For a sufficiently large number of samples, the most efficient method will be the one with the largest α. In an actual design scenario, low-sample accuracy is very important, and for this test none of the methods could reduce error much further than is shown on the graphs. The discussion that follows will include both terms, but more emphasis will be placed on α.

9.3.3 Unified Linear Error Model

In order to examine the effects of methods, arrays, and space dimensionality, the 16 models above were replaced with a single regression model. The two non-numeric factors were assigned dummy variables. XC represents choice of array type:

$$XC = -1 \qquad \Rightarrow \text{crossed array (X)} \tag{140}$$

$$XC = 1 \qquad \Rightarrow \text{combined array (C)} \tag{141}$$

Table 13: Experimental Design and Regression Coefficients

X-DoE

p_D	p_S	# data	N_{min}	N_{max}	α	$\hat{N}_{0.1}$	R^2
2	2	128	120	1,000	2.18	292.82	0.71
2	3	95	264	2,000	2.52	595.61	0.86
3	2	174	135	10,600	0.50	7333.79	0.61
3	3	173	297	23,800	0.56	12930.52	0.63

X-MOSI

p_D	p_S	# data	N_{min}	N_{max}	α	$\hat{N}_{0.1}$	R^2
2	2	202	106	500	4.97	158.81	0.83
2	3	158	215	737	5.07	320.37	0.80
3	2	445	110	950	3.65	252.69	0.90
3	3	343	215	1475	3.47	490.44	0.91

C-DoE

p_D	p_S	# data	N^*_{min}	N_{max}	α	$\hat{N}_{0.1}$	R^2
2	2	133	20	300	1.53	103.66	0.89
2	3	126	25	500	1.39	164.55	0.89
3	2	137	25	2000	0.74	619.02	0.83
3	3	112	34	9000	0.69	818.92	0.86

C-MOSI

p_D	p_S	# data	N^*_{min}	N_{max}	α	$\hat{N}_{0.1}$	R^2
2	2	389	20	125	3.00	41.22	0.88
2	3	717	25	206	2.41	56.46	0.87
3	2	486	25	144	2.44	56.84	0.87
3	3	819	34	235	2.56	71.82	0.88

For combined arrays, N^*_{min} is larger than the actual minimum sample size, as small samples have been excluded to improve fits and focus on the non-transient region.

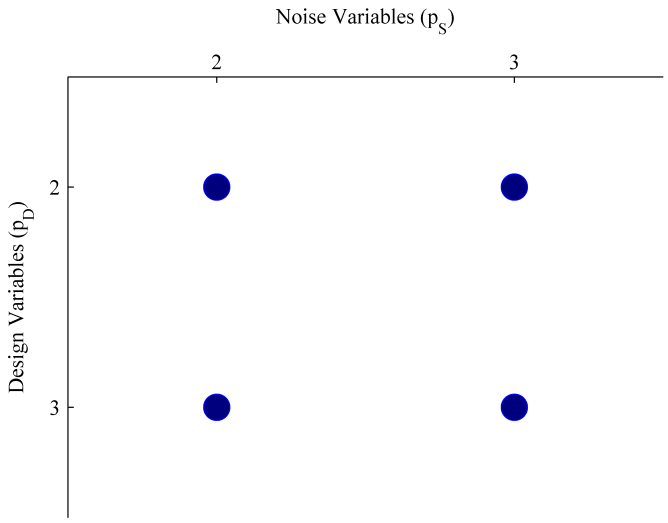


Figure 69: Experimental Design, settings for number of design variables (p_D) and noise variables (p_S).

and DM represents type of method:

$$DM = -1 \qquad \Rightarrow \text{design of experiments (DoE)} \tag{142}$$

$$DM = 1 \qquad \Rightarrow \text{multi-objective statistical improvement (MOSI)} \tag{143}$$

The power model format was maintained, but each of the two linear coefficients for the transformed equation was assumed to be a sum of linear contributions from each of the variables and interactions:

$$\begin{aligned}\log_{10}(RMSE) =&(b_0 + b_{XC} \cdot XC + b_{DM} \cdot DM + b_{p_D} \cdot p_D + b_{p_S} \cdot p_S \\ &+ b_{(XC \times DM)}(XC \times DM) + \cdots + b_{(p_D \times p_S)}(p_D \times p_S)) \\ -&(a_0 + a_{XC} \cdot XC + a_{DM} \cdot DM + a_{p_D} \cdot p_D + a_{p_S} \cdot p_S \\ &+ a_{(XC \times DM)}(XC \times DM) + \cdots + a_{(p_D \times p_S)}(p_D \times p_S)) \cdot log_{10}N\end{aligned} \tag{144}$$

All two-factor interactions were included, but no square terms, since only a two-level design was used. For convenience, the statistical package JMP was used for regression and effects testing. A list of all terms, their coefficients, their significance, and their confidence intervals are shown in Table 14. The R^2 for the whole model was 0.88, comparable to the goodness of fits of the 16 individual error models.

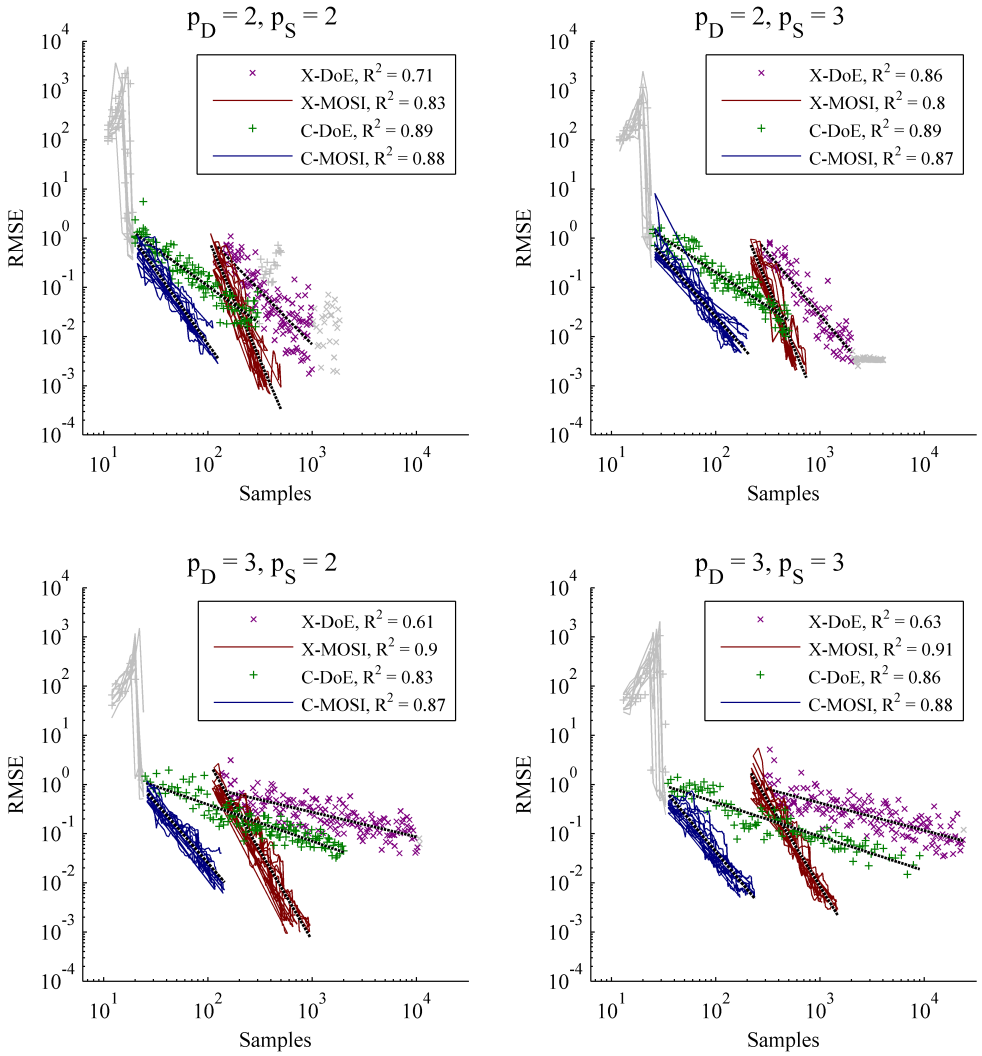


Figure 70: Regression data for RMSE vs. samples for four different settings of design (p_D) and noise (p_S) variables. All four methods are shown in each plot, with regression lines shown in black. The early transient data has been removed from the combined-array methods, and is shown in gray. Purely from visual inspection, it appears that the DoE methods are more sensitive than MOSI methods to number of design variables (top vs. bottom), and the crossed-array methods appear more sensitive than the combined-array methods. From inspection alone, the effects of increasing the number of noise variables (left vs. right) seem to be primarily to shift the graphs to the right.

Table 14: Linear Regression Terms and Regression Results

Coefficient	Term	Estim.	Std. Error	Lower 95%	Upper 95%	F Ratio	$P(>F)$
b_0	Intercept	7.440	0.636	6.193	8.687		0.E+00
b_{XC}	XC	-3.085	0.245	-3.565	-2.605	158.593	9.E-36
b_{DM}	DM	2.011	0.172	1.673	2.349	136.254	5.E-31
b_{p_D}	p_D	-1.496	0.220	-1.927	-1.065	46.271	1.E-11
b_{p_S}	p_S	0.424	0.232	-0.030	0.878	3.351	0.0672
$b_{(XC\times DM)}$	$XC \cdot DM$	-0.889	0.033	-0.955	-0.824	704.713	1.E-144
$b_{(p_D\times XC)}$	$p_D \cdot XC$	1.078	0.076	0.930	1.226	203.021	4.E-45
$b_{(p_S\times XC)}$	$p_S \cdot XC$	-0.563	0.063	-0.686	-0.440	80.342	4.E-19
$b_{(p_D\times DM)}$	$p_D \cdot DM$	0.225	0.052	0.123	0.327	18.611	2.E-05
$b_{(p_S\times DM)}$	$p_S \cdot DM$	-0.200	0.045	-0.288	-0.112	19.995	8.E-06
$b_{(p_D\times p_S)}$	$p_D \cdot p_S$	-0.078	0.079	-0.232	0.076	0.983	0.3215
a_0	$-\log_{10}(N)$	5.858	0.296	-6.438	-5.278	392.075	8.E-84
a_{XC}	$XC \cdot (-log_{10}(N))$	-1.339	0.099	1.144	1.533	181.365	1.E-40
a_{DM}	$DM \cdot (-log_{10}(N))$	1.409	0.076	-1.557	-1.260	345.140	2.E-74
a_{p_D}	$p_D \cdot (-log_{10}(N))$	-1.259	0.102	1.060	1.459	153.275	1.E-34
a_{p_S}	$p_S \cdot (-log_{10}(N))$	-0.469	0.106	0.260	0.678	19.416	1.E-05
$a_{(XC\times DM)}$	$XC \cdot DM \cdot (-log_{10}(N))$	-0.248	0.013	0.222	0.274	353.437	5.E-76
$a_{(p_D\times XC)}$	$p_D \cdot XC \cdot (-log_{10}(N))$	0.472	0.030	-0.531	-0.412	241.892	3.E-53
$a_{(p_S\times XC)}$	$p_S \cdot XC \cdot (-log_{10}(N))$	-0.127	0.024	0.079	0.174	27.632	2.E-07
$a_{(p_D\times DM)}$	$p_D \cdot DM \cdot (-log_{10}(N))$	0.079	0.022	-0.122	-0.035	12.264	0.0005
$a_{(p_S\times DM)}$	$p_S \cdot DM \cdot (-log_{10}(N))$	-0.198	0.019	0.161	0.234	112.839	5.E-26
$a_{(p_D\times p_S)}$	$p_D \cdot p_S \cdot (-log_{10}(N))$	0.122	0.036	-0.193	-0.051	11.354	0.0008

The terms α and $\hat{N}_{0.1}$ can now be re-defined in terms of the unified linear model:

$$\alpha = \sum_{i=1}^{11} a_i \cdot Z_i \tag{145}$$

$$\hat{N}_{0.1} = 10^{\left(\frac{1+\sum_{i=1}^{11} b_i \cdot Z_i}{\alpha}\right)} \tag{146}$$

where the Z_is are the control variables (XC, p_D, etc.).

9.3.4 Sensitivity to Problem Dimensionality, Results Analysis

The results presented in the previous subsection can be used to infer about the relative effects of problem dimensionality on method efficiency, and to answer Research Questions 1 and 2, re-printed here:

Research Question 1: For finding mean/risk Pareto frontiers, how does the relative efficiency of combined and crossed arrays depend on the number of noise variables?

Hypothesis 1: As the number of noise variables increases, the efficiency of combined array methods will suffer relative to the efficiency of crossed array methods.

Research Question 2: For finding mean/risk Pareto frontiers, how does the relative efficiency of design of experiments and multi-objective statistical improvement change with the number of design variables?

Hypothesis 2: As the number of design variables increases, multi-objective statistical improvement methods will become more efficient relative to a design of experiments.

9.3.4.1 Re-defining Research Question 1 in Terms of Interactions

In a previous subsection, efficiency was defined in terms of the error model, $\widehat{\text{RMSE}}(N)$, and is described by a sample sensitivity parameter α and an intercept term $\hat{N}_{0.1}$. Question 1 pertains to an interaction between number of noise variables p_S and the array choice, crossed (X) or combined (C), and the effect on the error model. When a unified linear model is adopted, with p_S and array choice as factors, then this question can be re-phrased in terms of the linear model:

Research Question 1 (re-phrased):

(a) What is the interaction effect $p_S \times$ XC on α (term $a_{(p_S \times XC)}$)?

(b) Is there a cross-over interaction effect between p_S and XC on $\hat{N}_{0.1}$?

Hypothesis 1 (re-phrased):

(a) The effect $p_S \times$ XC on α is negative

(b) If there is a cross-over interaction between p_S and XC, it does not cause C methods to improve over X methods.

where XC is the name given to a dummy variable that represents array choice, negative for crossed arrays and positive for combined arrays.

Hypothesis 1(a) states that as the array type goes from crossed (X) to combined (C), the value of α will decrease more at high values of p_S than it will at low values of p_S.

Figure 71 shows two interaction plots. The first shows the effects on α of interaction between p_S and XC. The X and C lines are not perfectly parallel, indicating an interaction, but the effect does not appear severe, and the significance of the result is not obvious. One would expect that the crossed-array line (X) would be essentially level. In a crossed-array method, the outer-array samples in each iteration can be thought of as a sort of "overhead" cost that should depend on the noise subspace and nothing else. In a power model, multiplying the number of samples by a constant factor does not change the exponent α. However, in the tests, a fixed noise array was used for all X-DoE runs, and it is possible that the array for $p_S = 2$ was better or worse than for $p_S = 3$. That α *improves* at $p_S = 3$ is an indication that the effect may simply be due to a better quality noise-space DoE at that setting. For combined-arrays, on the other hand, α appears to decline slightly. This term in the unified model is negative, as predicted by Hypothesis 1(a): $a_{(p_S \times XC)} = -0.127$, and is significant at a p-value of less than 0.0001 (see Table 14 for linear regression results).

Hypothesis 1(b) does not make as strong a prediction as does Hypothesis 1(a). Since the term $\hat{N}_{0.1}$ is not a term in the linear model, there is no interaction term that captures the effect ($p_S \times$ XC) on $\hat{N}_{0.1}$. It is not meaningful to say that $\hat{N}_{0.1}$ changes "more" or "less" for X or C models, because the changes will not be linear. The only phenomenon which can be meaningfully called in "interaction" is a crossover, for example if X is lower than C at $p_S = 2$ but higher at $p_S = 3$. So the only meaningful hypothesis with regard to the effects on $\hat{N}_{0.1}$ is that if such a cross-over *does* occur, it result in C methods getting worse than X methods at higher values p_S, and not the reverse.

Part (c) shows the effects of p_S on $\hat{N}_{0.1}$. Since $\hat{N}_{0.1}$ is not a linear effect, it is not even really proper to average out the other effects. Instead, every setting of the control variables should be looked at individually, and examined for cross-over effects. Instead, (c) shows an approximation of the effects of p_S on $\hat{N}_{0.1}$. For the plot, the logarithm of $\hat{N}_{0.1}$ was taken

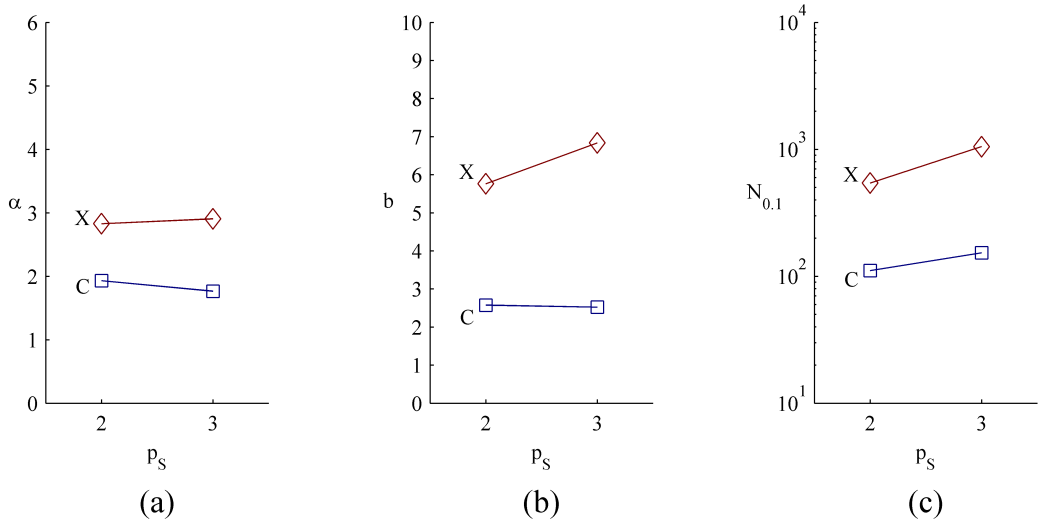


Figure 71: Interactions between noise dimensionality (p_S) and array type (crossed X or combined C). In (a), combined arrays (C) suffer greater degradation (decrease) in α from increased problem dimensionality than do crossed arrays (X). In (b), the intercept term b also shows interaction effects. Plot (c) shows that $\hat{N}_{0.1}$ degrades for both X and C methods. Note that (c) cannot properly be called an interaction plot because the y-axis is not a term in the linear model, and the values have been "illegally" averaged in log-space even though the effects may not be log linear; it is provided only to show gross effects, and no meaning should be ascribed to whether the lines are parallel or not.

for every combination of control variable settings, and these were averaged *as if* they were linear effects. For both X and C, $\hat{N}_{0.1}$ gets worse with increasing noise dimensionality, and is worse for X at both high and low values of p_S. As long as this holds true for all variable settings (and Figure 70 indicates that it does), there is no risk of a cross-over. If there is no cross-over, Hypothesis 1(b) is supported trivially.

Part (b) of the figure shows an interaction plot for the intercept, $b = \log_{10}(\widehat{\text{RMSE}}_{N=1})$. This is a true interaction plot, since b is a linear effect, but it is not physically meaningful since the method cannot be run for a single sample and thus $N = 1$ is outside the regression range. There appears to be a strong interaction, but any meaning of the effect must be derived from its influence on physically meaningful parameters.

Of the two requirements for Hypothesis 1, then, both (a) and (b) are supported. It appears that there *are* interaction effects between p_S and XC on α, that cause combined-array methods to suffer more than crossed-array methods from increases in noise space dimensionality. Any interaction effects on $\hat{N}_{0.1}$ cannot be readily interpreted with the current linear effects model.

9.3.4.2 Re-defining Research Question 2 in Terms of Interactions

Research Question and Hypothesis 2 can also be re-phrased as interaction, this time between number of design variables and method choice (DoE or MOSI):

Research Question 2 (re-phrased):

(a) What is the interaction effect $p_D \times$ DM on α (term $a_{(p_D \times DM)}$)?

(b) Is there a cross-over interaction effect between p_D and DM on $\hat{N}_{0.1}$?

Hypothesis 2 (re-phrased):

(a) The effect $p_D \times$ DM on α is positive

(b) If there is a cross-over interaction between p_D and DM, it does not cause DoE methods to improve over MOSI methods.

where DM is a dummy variable representing sampling method, negative for DoE.

Figure 72(a) shows the effects on α of interaction between p_D and DM. Both methods suffer with increasing design space dimensionality. Further, the lines are not parallel, indicating an interaction, and the DoE line suffers more than the MOSI line, supporting Hypothesis 2. In the unified regression model, this effect has an estimate $a_{(p_D \times DM)} = 0.07$, and the term was significant at a p-value of 0.0005. This was one of the less significant effects.

Hypothesis 2(b), like Hypothesis 1(b), is weaker than (a). Again, $\hat{N}_{0.1}$ is not a linear effect in the regression model, so it is not possible to draw conclusions directly from a particular coefficient. Instead, one would need to examine slices at every other combination of control variable settings to check for cross-over. There is no danger of cross-over, however, as long as MOSI methods always have lower $\hat{N}_{0.1}$ than their neighboring DoE method (which can be verified by looking at Figure 70). Hypothesis 2(b), then, is supported.

Plot (b) shows the interaction effect on the intercept term, $b = \log_{10}(\widehat{\text{RMSE}}_{N=1})$. There appears to be an interaction, but since none of the methods can be run for a single sample, this effect will only be seen through indirect influence on other physical parameters.

Both Hypothesis 2 (a) and (b) are supported by the data. It appears that MOSI methods suffer less from increasing design space dimensionality than DoE methods. The possible reason which led to this hypothesis in the first place was that the Pareto frontier is a smaller-dimensional subspace of the total design space, and the dimensionality of this subspace is purely a function of the number of objectives. As the design dimensionality is increased, the hypervolume ("tube") of designs around the frontier becomes a smaller fraction of the total space, so adaptive sampling methods that concentrate their efforts around this frontier were expected to suffer less than DoE methods which must model the response globally. Adaptively sampling with a method like EI helps alleviate the "curse of dimensionality", as does any optimization method.

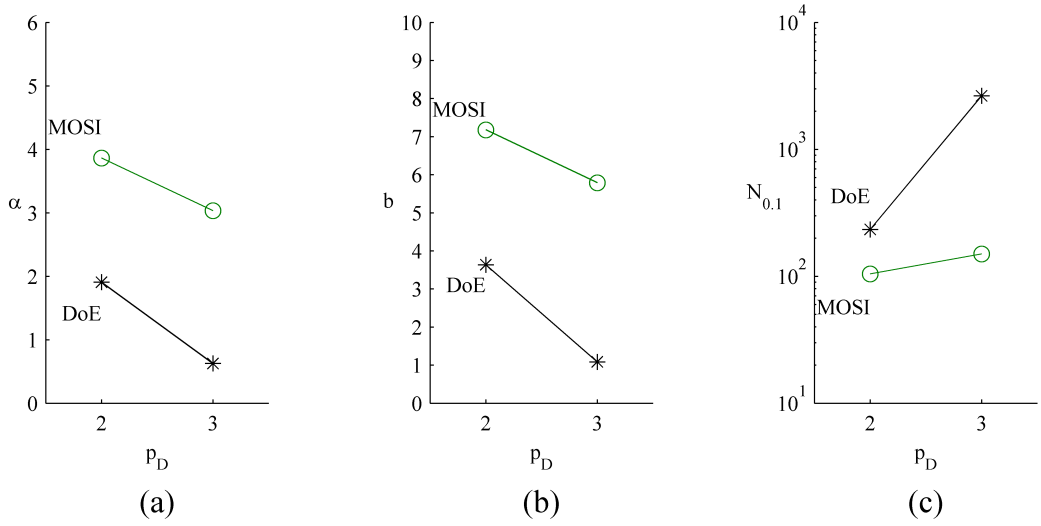


Figure 72: Interactions between design space dimensionality (p_D) and method type (DoE or MOSI). In (a), DoE methods suffer greater degradation to α than do MOSI methods as the number of design variables is increased. In (b), the interaction effect on the intercept term ($b = \log_{10}(\widehat{\text{RMSE}}_{N=1})$) is also positive, though the coefficient b is not physically very meaningful and appears to "improve". In (c), the more meaningful value of $\hat{N}_{0.1}$ is plotted, and both methods are seen to actually degrade. Plot (c) is not a true interaction plot, since $\log_{10}(\hat{N}_{0.1})$ is not a linear effect of the model; its values have been "illegally" averaged in log space to show gross effects, even though it may not be log-linear, and no meaning should be ascribed to whether or not the lines are parallel.

9.3.4.3 Other Notable Interactions

Of all the interactions tested, the previously discussed ($p_S \times XC$) and ($p_D \times DM$) were the fourth and fifth most significant, respectively. The three most significant interactions were ($XC \times DM$), ($p_D \times XC$), and ($p_S \times DM$), ranking by their effects on α. It is worth discussing these three, as they give further insight into the method behavior.

Figure 73 shows the interaction between array type (crossed X or combined C) and method type (DoE or MOSI). In (a), the α term is higher for MOSI methods than for DoE methods. Going from crossed to combined arrays, α declines, and this effect is stronger for MOSI methods. The intercept term b also shows an interaction, where the effects of going from crossed to combined arrays are stronger for MOSI methods than for DoEs.

The benefit of combined arrays is seen in (c), where $\hat{N}_{0.1}$ is lower for combined arrays. This brings back an earlier discussion about the non-domination of combined vs. crossed arrays. Though C-MOSI can provide a reasonable RMSE at a low number of samples, X-MOSI reduces error faster as more samples are added (and can reduce error further without hitting ill-conditioning). If no ill-conditioning effects were encountered, and the effects here could be extrapolated, eventually X-MOSI would become more efficient than C-MOSI for high numbers of samples. However, this is extrapolation into a region where experience has shown the relationships do *not* hold, so no real conclusions can be drawn from it.

Figure 74 shows the interaction between design space dimensionality and array type (crossed X or combined C). In (a), both array types experience degraded α's as the design dimension is increased, but crossed arrays are more sensitive than combined arrays. Plot (b) shows the sensitivity of the intercept term $b = \log_{10}(\widehat{\text{RMSE}}_{N=1})$, where crossed arrays are also more sensitive to design dimensionality. In (c), the degradation effect is also seen in $\hat{N}_{0.1}$. The take-away is that using combined arrays helps alleviate the curse of dimensionality somewhat, relative to crossed arrays. This is the basis for the entire field of Design of Experiments, where a careful design is used rather than multidimensional grids.

Figure 75 shows the interaction between number of noise variables p_S and sampling approach (DoE or MOSI). In (a), neither approach appears strongly influenced by number of noise variables; however, while the MOSI methods experience some expected degradation,

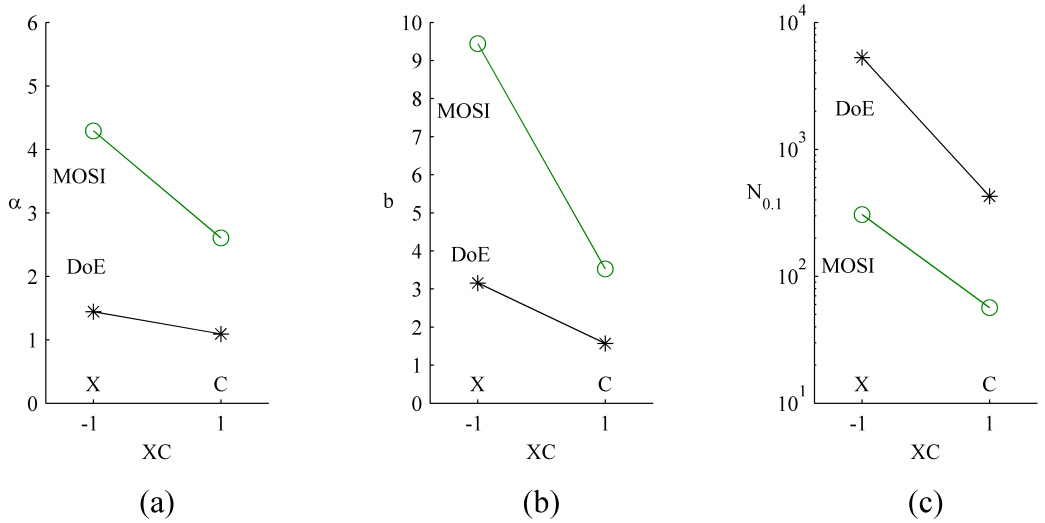


Figure 73: Interaction between array type and method type ($XS \times DM$). Plots (a) and (b) are interaction plots for the terms α and b, while (c) shows the effects on $\hat{N}_{0.1}$ and is instructive but cannot properly be considered an interaction plot.

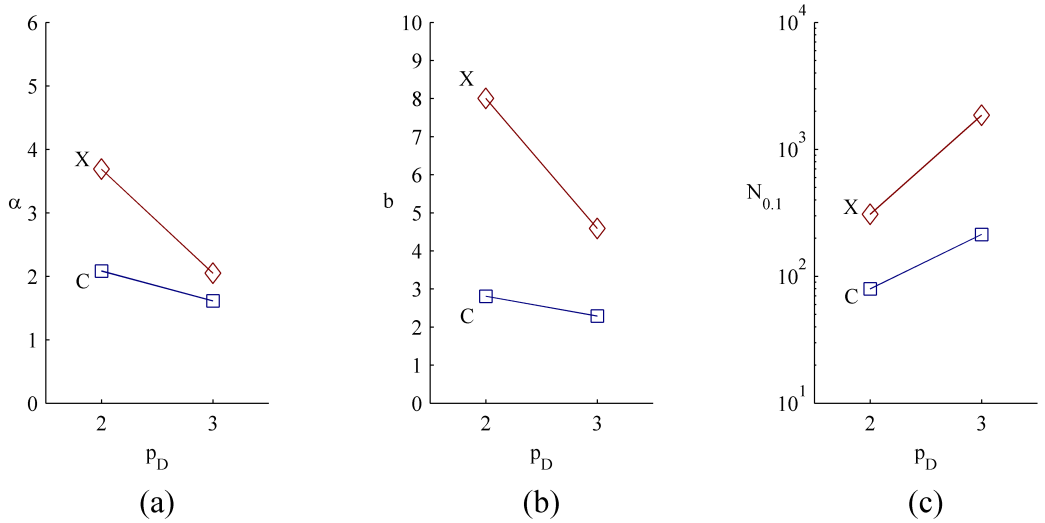


Figure 74: Interaction between number of noise variables p_D and array type X or C. There is an interaction effect on α, where crossed arrays are more sensitive to design space dimensionality than combined arrays. Plot (b) shows an interaction in the intercept term b, and plot (c) shows that $\hat{N}_{0.1}$ degrades for both combined and crossed arrays. Plot (c) is not a true interaction plot because $\log_{10}(\hat{N}_{0.1})$ is not a linear effect of the model, so no meaning should be ascribed to whether the lines are parallel.

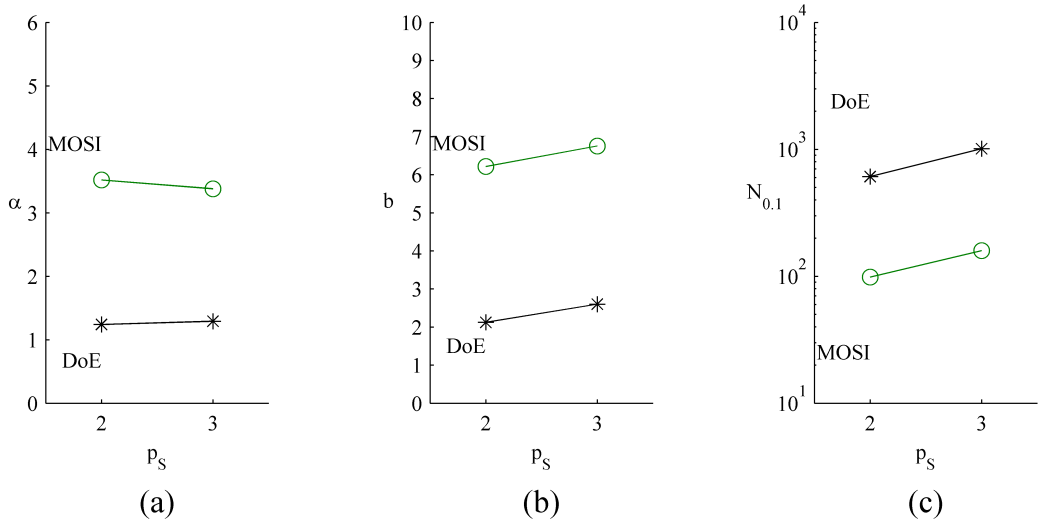


Figure 75: Interaction between number of noise variables p_S and sampling approach (DoE or MOSI). In (a), α degrades with increasing p_S for MOSI methods, as might be expected. However, it appears to improve for DoE methods. This is puzzling and left unexplained. The intercept term does not appear to show significant interaction effects in (b).

the DoE methods are seen to actually improve as noise dimensionality increases. This is a puzzling result, and no explanation will be offered, though one wonders if it is simply bias or a shortcoming of the unified model.

9.3.4.4 Broad Interpretation

After staring at so many interaction plots, it is now possible to make several generalizing statements.

- Crossed (X) arrays are more sensitive to design dimensionality than Combined (C) arrays

  but they are less sensitive to noise dimensionality

- DoE sampling is more sensitive to design dimensionality than MOSI sampling

  but it is less sensitive to noise dimensionality

These statements should be taken with the caveats that this is but one small experiment, that the results may not be generalizable, and that in the above statements the noise space-based affects may be biased by noise array methodology in the crossed-array methods.

9.3.4.5 A Partial Answer to Research Question 3

Recall Research Question 3:

> **Research Question 3:** Is there a design scenario where a combined array Multi-Objective Statistical Improvement method out-performs both crossed-array and design of experiments methods in terms of efficiency?

It is not possible to give a definitive answer to this, partly because it is not possible to give a single metric for "efficiency". It is possible to give a qualified answer, however:

- If the sample budget is below a certain threshold, under certain design scenarios, C-MOSI will be more efficient than C-DoE, X-MOSI, and X-DoE.
- If the required accuracy level is above a certain threshold, under certain design scenarios, C-MOSI will be unable to reduce error sufficiently, and X-MOSI will be more efficient than the other three methods.

The first statement pertains to a fixed sample size, and efficiency measured as RMSE along the true Pareto frontier. The second statement pertains to a target RMSE, and efficiency measured as the number of samples. Together, the statements reflect that in the tests performed, C-MOSI and X-MOSI together formed an efficient frontier, C-MOSI at the low-sample end, and X-MOSI at the low-error end, as can be see in every quadrant of Figure 70.

9.3.4.6 Caveats and Qualifications

Though care has been taken to qualify results and interpretations as they have been presented, it is perhaps worthwhile to collect the important caveats and qualifications related to this experiment in one place.

First and foremost, all results in this chapter are for a single scalable test function that has low roughness, a nearly linear noise space, and little interaction between the

design and noise spaces. The tests were run only over a small range of design and noise variable sizes, between 2 and 3, which are very small numbers relative to any actual likely design problems. There are many other problem characteristics that will change, including roughness, linearity, interactions, and a host of other characteristics that perhaps don't even have names but would be lumped under the non-specific term "shape". The observations here might hold under other conditions, but there is no guarantee of that nor any way to use language more precise than to say it "seems likely".

Some of the specific implementation choices from this experiment likely affected the outcome. The use of fixed arrays in noise space for the X-DoE (and X-MOSI warm-starts) introduced an element that did not "scale" precisely with noise dimensionality. It is possible that the choice introduced a bias. Using random noise arrays might have resulted in less bias, but might also have been less fair to the crossed array methods since less effort would have been spent optimizing each noise array.

The choice of Gaussian Process models also significantly affected the results, since ill-conditioning of the covariance array is a problem unique to them. Other options could have included treed Gaussian Processes [47] or Relevance Vector Machines [110].

The hypotheses were re-phrased in terms of interaction effects, which seems like a good way to quantify them, but is not the only possible choice. The choice of a linear effects model was somewhat arbitrary, as was the choice of transformed sample and RMSE variables. The only justification is that the final model had good fit, at least over the regression data (which had notable exclusions).

9.4 Summary of Experimental Results

In this chapter, several experiments explored the behavior of four methods: two different sampling approaches, Design of Experiments (DoE) and Multi-Objective Statistical Improvement or (MOSI); and two different array types, Crossed (X) and Combined or (C).

All four methods (X-DoE, X-MOSI, C-DoE, and C-MOSI) were evaluated in terms of their final Gaussian Process surrogate models, and how accurately they could predict the mean and Value-at-Risk (percentile) along a known "true" Pareto frontier. The error in the

mean and VaR terms was combined into a single Root Mean Square Error metric (RMSE).

First, the combined-array methods (C-DoE and C-MOSI) were treated as a continuum. Each execution starts at a warm-start C-DoE of a particular size, and ends after a certain number of C-MOSI iterations. By doing a sweep on C-DoE warm-start size, and running C-MOSI on each until the model accuracy stopped improving, the optimal warm-start size was found. For this test case, a threshold RMSE could be reached with the fewest samples by starting with the smallest possible warm-start. This also showed that for the test problem with dimensionality $p_D = p_S = 2$, C-MOSI was more efficient than C-DoE.

In most tests, C-MOSI was run until the true error began to climb due to ill-conditioning effects. Since this point would not be known in practice, a stopping criteria based on epistemic Bayesian uncertainty was implemented, and shown to be workable.

Next, the crossed-array methods (X-DoE and X-MOSI) were also treated as a continuum, with similar results. X-MOSI was found to be more efficient than X-DoE in terms of accuracy and samples. Between C-MOSI and X-MOSI, however, it was not possible to pick a dominant method, because while C-MOSI dominated at low numbers of samples, it was incapable of reaching the high level of accuracy produced by X-MOSI for much larger sample budgets.

In the second set of experiments, the error progressions of the four methods were modeled as a function of number of samples. With the exception of ill-conditioned models and some transient effects for combined arrays with small sample sizes, a power law was found to represent the data well. Number of design dimensions and number of noise dimensions were treated as sensitivity variables, and were both varied from 2 to 3. A total of 16 error models were developed:

2 array types

x 2 sampling approaches

x 2 noise space dimensionalities

x 2 design space dimensionalities

A unified linear effects model was then developed to simultaneously capture the effects of

samples, methodological choices, and space dimensionalities. The model had fit comparable to the individual error models, with an $R^2 = 0.88$. Research Questions 1 and 2 pertained to the sensitivities of the different methods to problem dimensionality, and they were rephrased as questions about the interactions in the unified error model. Both hypotheses were supported by the data.

CHAPTER X

DEMONSTRATING C-MOSI ON AN ELECTRIC POWER PORTFOLIO TEST PROBLEM

Though to this point C-MOSI has been demonstrated and tested alongside three other methods, the question remains whether it can be used on a problem more complex than the analytic scalable test function. To that end, the method was tested on the electric power portfolio simulation model described in Chapter 7.

10.1 Electric Power Portfolio Test Problem

The test problem involved the simulation model described in Chapter 7. In order to represent an elecric power utility test case, the test problem had to present a mean/risk Pareto frontier similar to the ones found in utility planning documents. Two such frontiers from the PacifiCorp IRP are shown in Figure 76, previously shown in Chapter 2.

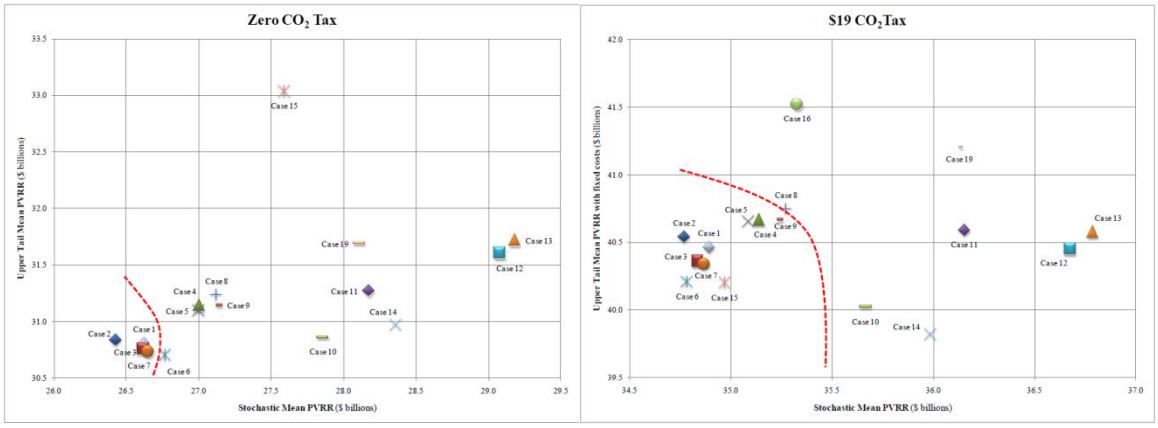


Figure 76: PacifiCorp's frontier plots for two carbon price scenarios. The IRP document contains additional plots for other carbon scenarios [90]

The most important characteristic is simply that there *is* a frontier. Depending on the assumptions made about costs and performance, it is possible that a single portfolio have both the lowest mean cost and lowest cost risk, or that the frontier be so small as

Table 15: Capital Cost Assumptions For Demonstration Case

Equipment	Annualized Capital Cost	Units
Wind Turbines	412,500	$/MW
Photovoltaic Arrays	300,000	$/MW
Energy Storage	65,000	$/MWh
Natural Gas Plants	25,000	$/MW
Demand Side Management	80,000	$/unit

Table 16: Design Variable Ranges

Variable	Units	Min	Max
Wind turbines	MW rated capacity	0	200
PV installations	MW rated capacity	0	200
Energy storage	MWh capacity	0	200
Natural gas plants	MW rated capacity	0	200
Demand Side Management	"units"	0	10

to be insignificant. Some of this researcher's early attempts to apply C-MOSI encountered precisely this problem, where the problem did not actually require a frontier-finding method at all, and was "too easy" compared to a real portfolio problem. To ensure that the test problem reflected the challenge presented by a utility portfolio planning problem, the capital cost assumptions and noise variable uncertainty distributions were adjusted so that a frontier was present. The capital assumptions used for the test case are shown in Table 15, and the noise variable distribution assumptions are shown in Table 17. These assumptions do not necessarily reflect the best available information for any particular time period, so the results of this test should not be construed as reflecting portfolios that are truly optimal in the real world.

The design variables (portfolio options) for this test case were wind farms, photovoltaic

Table 17: Noise Variables

Variable	Base Value	Units	Modifier	Distribution
Mean wind speed	8.0	m/s	Added	$\mathcal{N}(\mu = 0, \sigma = 1/30)$
Natural Gas Price	45	$/MW	Multiplier	$\Gamma(k = 1.8928, \theta = 0.8928) + 0.5$
Demand	(series)*	MW	Multiplier	$\Gamma(k = 1.4434, \theta = 0.4434) + 0.5$
Market Trans. Price	80	$/MW	Multiplier	$\Gamma(k = 1.8000, \theta = 0.8000) + 0.5$

*Demand parameters: $P_{\text{mean load}} = 100\text{MW}$, $P_{\text{annual}} = 20\text{MW}$, $P_{\text{daily}} = 50\text{MW}$, $\sigma_{\text{demand}} = 10\text{MW}$.

installations, energy storage, natural gas plants, and demand-side-management, as described in Chapter 7.

The demand and the design variable ranges were kept small so that the number of wind turbines and photovoltaic arrays would remain small, for simulation run-time considerations.

The noise variables used were mean wind speed (to represent wind farm siting uncertainty), natural gas price, demand (a linear re-scaling factor), and market electricity price. Mean wind speed was given a Gaussian distribution with a fairly small variance. The other three variables were given Gamma distributions.

The number of design and noise variables (5 and 4, respectively) were larger than in the previous experiments conducted on the scalable test problem. As a first step, that same scalable test problem was run at a higher dimensionality, with 5 design and 5 noise variables. All four methods were run, in the same manner as in the sensitivity experiment from Section 9.3. The results are shown in Figure 77, and show the same trends as in the lower-dimensional problems. The algorithm details were not changed.

10.1.1 Transformed Noise Variables

Natural gas price, electricity demand, and market electricity price were given Gamma distributions, shown in 78. The Gamma pdf is denoted $\Gamma(k, \theta)$ where k is a shape parameter and θ is a scale parameter. A Gamma distribution is bounded below, but has a long upper tail. The three noise variables, also, are bounded below (they cannot go negative) and may potentially increase substantially. The choice of a Gamma distribution does not reflect any source of data, but was used to demonstrate that the method can be used with non-Gaussian noise distributions. The parameters for the three variables can be found in Table 17. Note that a constant of 0.5 was added in all three cases, and acted as a strict lower bound.

The analytical second order probability analysis used for experiments in the previous chapter requires that the noise variable distributions be Gaussian. However, even if the true noise distributions are not Gaussian, it is still possible to use the method by transforming the noise inputs. The Gaussian Process is then fit to a transformed function, rather than directly fit to the simulation data. For the purposes of second order probability analysis,

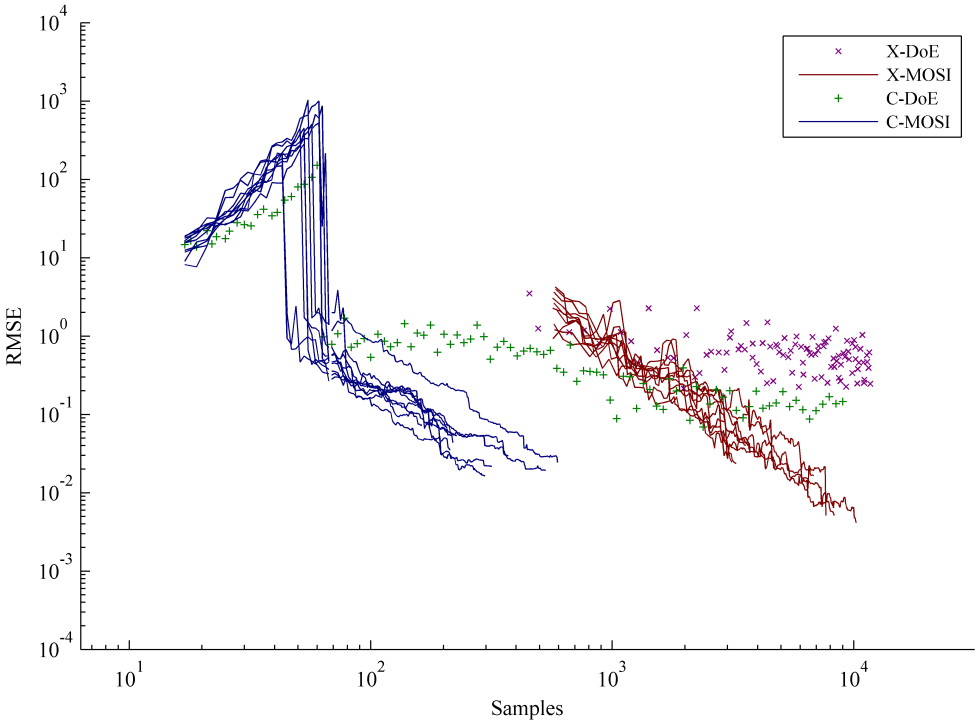


Figure 77: Results from scalable test problem with $p_D = p_S = 5$. The scalable test problem showed the same trends at this dimensionality as at lower dimensionality, without changes to the algorithm.

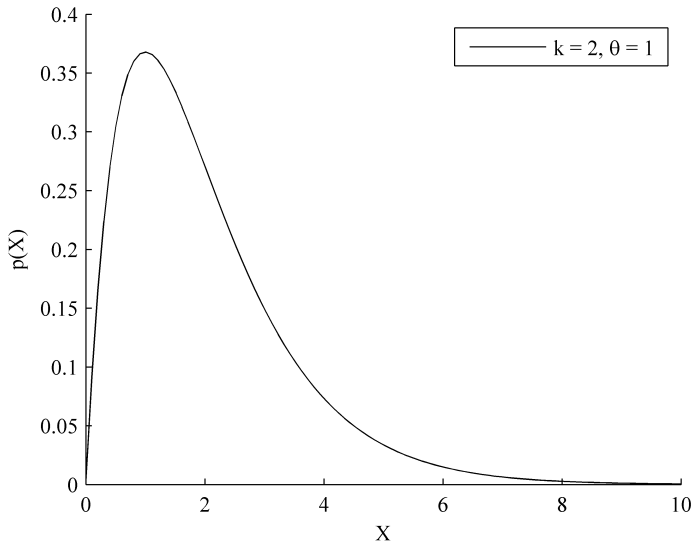


Figure 78: A gamma distribution, with $k = 2$ and $\theta = 1$, similar to the distributions used in the simulations.

the noise variables were assumed to follow the standard normal distribution:

$$S \sim \mathcal{N}(\mu = 0, \sigma^2 = 1) \tag{147}$$

However, when fed into the simulation, they were transformed by:

$$S^* = F_\Gamma^{-1}(\Phi(S); k, \theta) \tag{148}$$

where $\Phi(\cdot)$ is just the standard normal CDF, and $F_\Gamma^{-1}(\cdot; k, \theta)$ is the inverse CDF of the Gamma distribution with shape parameter k and scale parameter θ. The Gaussian Process, however, was fit to S. Any response that is linear with respect to simulation inputs S^*, therefore, will appear more complex as a function of S.

In this particular test case, for any given portfolio, the electricity cost is linear with respect to the fuel price and market electricity price. If this were a real engineering problem, rather than a method demonstration, it would not even be necessary to explore those two noise dimensions. In fact, since the Gaussian Process models used here have a linear prior, the linear dimensions become "too easy" if fit to un-transformed variables: the correlation parameter terms (see Equation 20) become very small, and this can lead to numerical instabilities. Transforming the noise variables therefore has a second function in this experiment, of making the noise dimension "harder" so that it both better represents a real problem and does not experience numerical instabilities. An example of a linear response that is modified by transformed input variables is shown in Figure 79. It might be the case in practice that such noise variables truly would be linear; in such a case, the simulation should be used to calculate intermediate variables such as fuel use and electricity imports, and total cost could be calculated externally. This would require some modification of the method, and this is left to future work.

10.1.2 Wind vs. Natural Gas Trade and Adjusted Assumptions

In order to properly test the method, it was important that the test problem exhibit a mean/risk frontier. To assure that this would be the case, a small number of cases were run to ensure that a trade existed between natural gas plants and wind farms. Wind farms exhibit low price risk: though wind production is not guaranteed or perfectly predictable, it

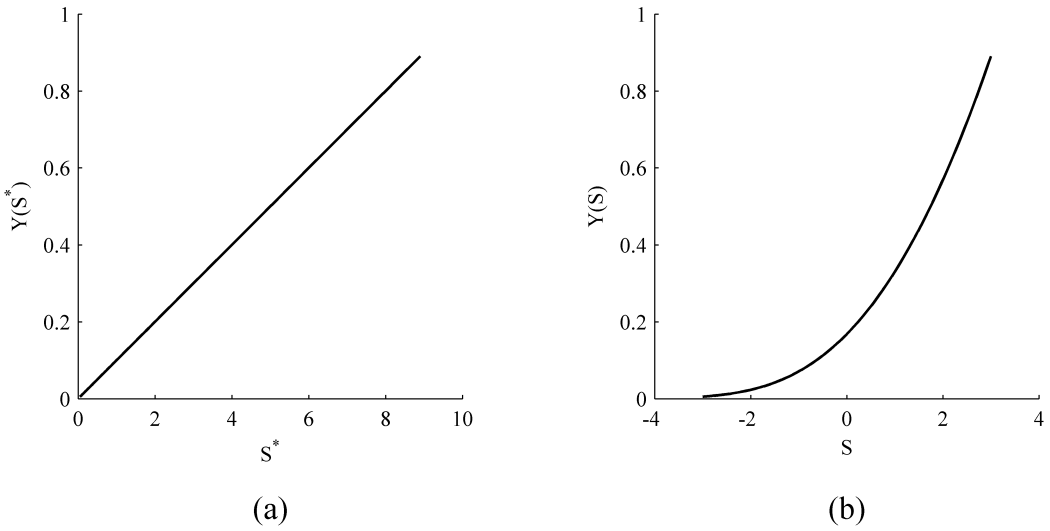


Figure 79: A transformed input. S^* has the Gamma distribution found in Figure 78, which is assumed to be the true distribution of the noise variable. In (a), Y is a linear function of S^*. In (b), Y is shown as a function of S, which has a standard normal distribution. The functional form is more complex, but analytic SOP analysis can be used. All S inputs must be transformed into S^* before they are input to the simulation.

is completely insensitive to fuel price fluctuations. Natural gas plants, however, are sensitive to fluctuations in the price of natural gas. To ensure that at least one trade existed, the capital costs of the two resources and the uncertainty of fuel prices were set so that wind power had higher expected cost than natural gas, but natural has had higher cost risk. This was verified by running a five-by-five grid of 25 designs where only wind farms and natural gas plants were present, each varied from low to high levels. For each design, 200 Monte Carlo cases were run to assess uncertainty. These 25 cases are shown on a mean/risk plot in Figure 80. The model assumptions were adjusted and the process was repeated several times to ensure that this frontier was significant.

Photovoltaics are also low-risk once they have been installed, so the cost of Photovoltaic systems was set high enough (on a per MWh basis) that it could not fully dominate this entire frontier. Neither storage nor demand-side-management can completely satisfy demand, so the appearance of a wind/gas trade was a good indicator that a similar trade would appear on the final frontier.

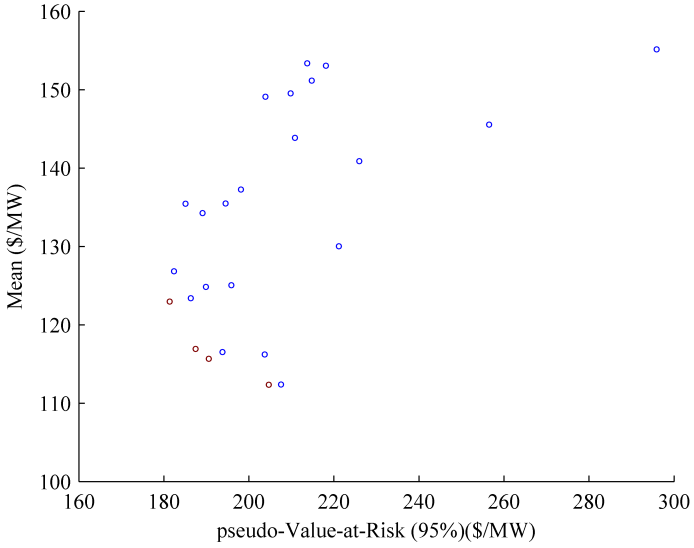


Figure 80: Small population of test cases to assure that a trade exists between mean and pseudo-VaR. These represent a full factorial combination of wind farm and natural gas plant sizes, from 0 to 200MW in increments of 50MW.

10.2 Independent Search for the Frontier

A multi-objective evolutionary algorithm with a very large number of function calls was used to find an approximation of the true mean/VaR Pareto frontier. The MOEA was combined with Monte-Carlo simulations to find mean and pseudo-VaR at every design. The MOEA used was NSGA-ii, implemented using MATLAB's built-in *gamultiobj* function. A fixed Monte Carlo population of 200 was used for all designs. From the 25 wind/gas cases, the Pareto set were fed into the initial population of the GA, to ensure that it quickly progressed beyond them.

The MOEA was allowed to run for 20 generations, with a population of 40, and 200 Monte Carlo runs at each design, for a total of 160,000 function calls. The resulting frontier is shown in Figure 81. Despite the large number of function calls, the search was not truly very exhaustive due to run-time limitations. Some of the 25 wind/gas only runs remained on the frontier, and with only 200 Monte Carlo runs per design, the exact values are likely somewhat inaccurate.

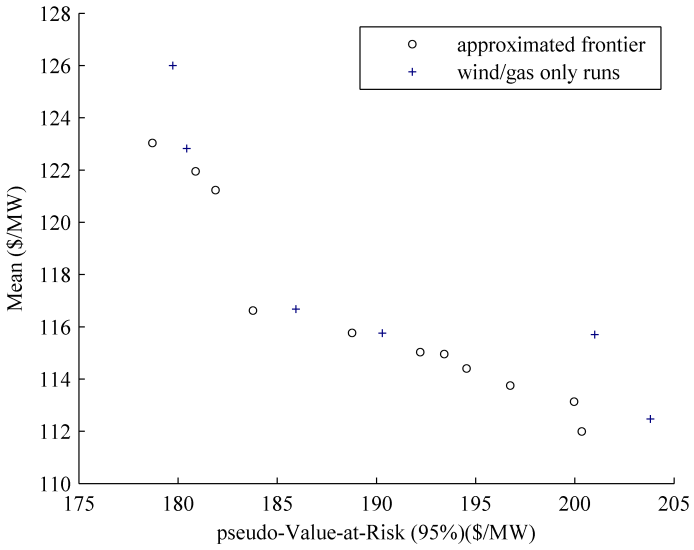


Figure 81: The Pareto frontier found through an NSGA-ii run with Monte Carlo runs in noise space and X function calls. The six '+' symbols show the Pareto set from the 25 wind/gas only runs.

10.3 Implementing C-MOSI and C-DoE

For the most part, the implementation of C-MOSI and C-DoE was the same as in previous experiments, with a couple of changes. The noise variable transformation was implemented as a wrapper around the simulation code. Additionally, since the design variable ranges were much larger than in the previous experiments, they were re-scaled to the interval [0,1] before the Gaussian Process was fit, so that the θ correlation parameters would not become poorly scaled. Additionally, the responses were divided by a factor of 70 to bring them closer to unity, again to keep all scaling reasonable. This was all handled internally within a Gaussian Process object.

Also unlike in the experiments conducted on the scalable test function, there was no pre-knowledge of the underlying functional shape, so the relative scaling of the correlation parameters could not be fixed. Instead, the GP fitting process treated each input as being independently scaled, which increased the complexity and computational overhead of the GP fitting stage. The θ correlation parameters were fit by maximizing log-likelihood with a Genetic Algorithm that used a population of 100 and allowed a maximum of 100 generations. The algorithm was run 8 independent times, and stopped sometime after its epistemic

uncertainty bottomed out.

Unlike in the experiments on the scalable test problem, the "true" Pareto frontier was not known during the C-MOSI run. This will be true in a real situation, as well. However, a surrogate for RMSE was tracked over the course of the runs, namely the epistemic uncertainty along the current predicted frontier, σ_{total} as presented in section 9.2.1.1 and Equation 131.

$$\sigma_{\text{total}} = \sqrt{E[\sigma_e^2]} = \sqrt{\frac{1}{2 \cdot N_P} \sum_{i \in P} \left(\frac{\text{Var}[\mu]^{(i)}}{R_\mu^2} + \frac{\text{Var}[\text{pVaR}]^{(i)}}{R_{\text{pVaR}}^2} \right)}$$

where R_μ and R_{pVaR} terms are the ranges of the Pareto set. The summation is over the current estimated Pareto set based on predicted mean values, as discussed in section 6.3. This metric can be thought of as the epistemic equivalent of the RMSE. It was previously shown to correlate with RMSE, and was proposed as a stopping criteria in section 9.2.1.1.

For C-DoE, the same settings were used, but the combined design/noise samples were selected with the built-in MATLAB function *lhsdesign*, as in the previous experiments. The same type of GP was fit.

10.4 Implementing X-MOSI

The implementation of X-MOSI was again largely the same as in the previous set of experiments, with the same set of modifications. The noise variables were transformed using a wrapper around the simulation code, and I-SOP was used to sample them at every design. The design variables were re-scaled on the interval [0,1] before the two GP models were fit, and all response values were divided by 70. The correlation parameters were fit with the same Genetic Algorithm settings, a population of 20 and a maximum of 100 generations. The algorithm was allowed to continue for about 300 iterations, over which it amassed about 30,000 samples.

X-DoE was not run, due to run-time limitations.

10.5 Comparative Performance of the Methods

The methods were assessed using basically the same methods as seen in the previous experiments. The RMSE of the model was assessed with regard to the "true" Pareto set. However,

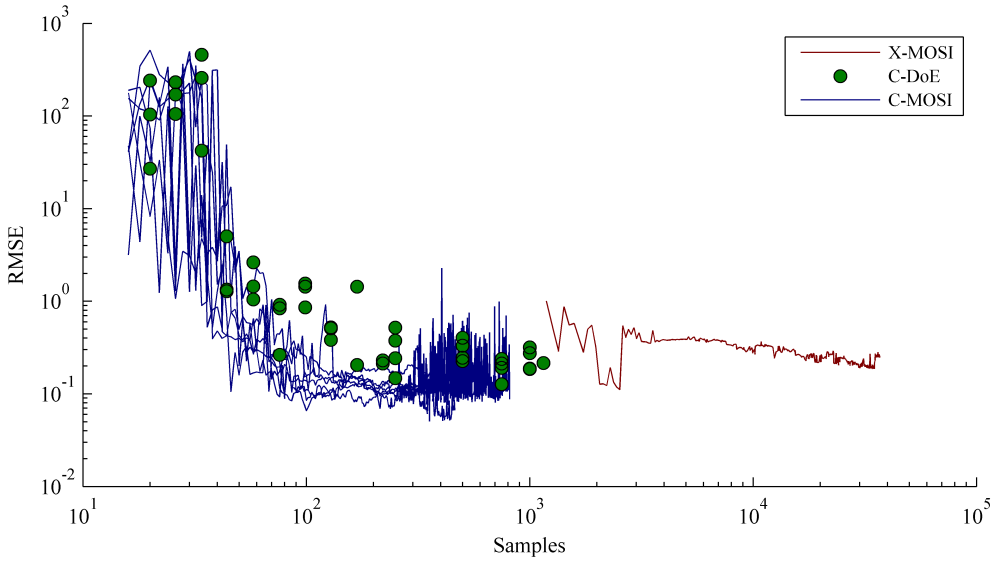


Figure 82: Error progression for C-MOSI, C-DoE, and X-MOSI.

unlike with the scalable test problem, where the frontier was known analytically, in this case it was found by NSGA-ii, with a limited sample budget. Therefore the "true" Pareto set used here was actually an approximation, and importantly was found independently of the three methods tested.

The results are similar to those seen in the previous experiments, and the error progression as a function of samples is shown in Figure 82. This figure is the equivalent of Figure 77, though only a single X-MOSI run was performed, and no X-DoE runs. It does show the relative performance of C-MOSI and C-DoE.

The X-MOSI trace is shown on the same plot, in red. Based on a single run, it appears that its relative performance is similar to that in the analytic tests, though it was not run with enough samples to determine if it would eventually reduce error further.

10.5.1 Discussion of Method Performance

As with the analytic test function, both C-MOSI and C-DoE experience an initial period of very high error, followed by a sharp drop. The C-MOSI method can be seen to reduce error faster than C-DoE, though the benefit is not as clear as in the analytic tests, and there is some overlap between the performance of the two methods.

Unlike in the analytic cases, the RMSE of the C-MOSI runs quickly levels off, showing

very slow improvements after about 100 samples. At around 300 samples, the RMSE of the C-MOSI runs begins to fluctuate from one sample to the next.

Since the true Pareto frontier has only been approximated, it is possible that the leveling off of error is due to this. Error in the approximated frontier will result in a lower bound on RMSE, since the apparent error could never go below the error in the approximated frontier except by an unlikely random chance. It is also possible that the leveling off is due to poor fits or ill-conditioning in the GP, or some other set of factors.

The most likely explanation for the oscillations appears to be ill-conditioning of the Gaussian Process models. The spikes in RMSE correspond to iterations where the epistemic uncertainty in the model is also high, which could simply be a symptom of poor fits but was previously seen as a symptom of ill-conditioning (see section 9.2.1.1). As can be seen in Figure 85, high epistemic uncertainty at the current estimated Pareto frontier correlates highly with high RMSE along the true Pareto frontier. Although it is not a perfect predictor, it was proposed as a stopping criteria in section 9.2.1.1. Unlike in the analytic test cases, however, the RMSE does not climb continuously after it bottoms out; instead, it fluctuates wildly, and the low points still show low error. However, the plots in Figure 85 show that once the σ_{total} reaches its minimum value, RMSE does not improve much further. It is proposed, therefore, that the runs be stopped after the σ_{total} does not reduce for some number of iterations. At that point the model corresponding to the lowest σ_{total} should be used.

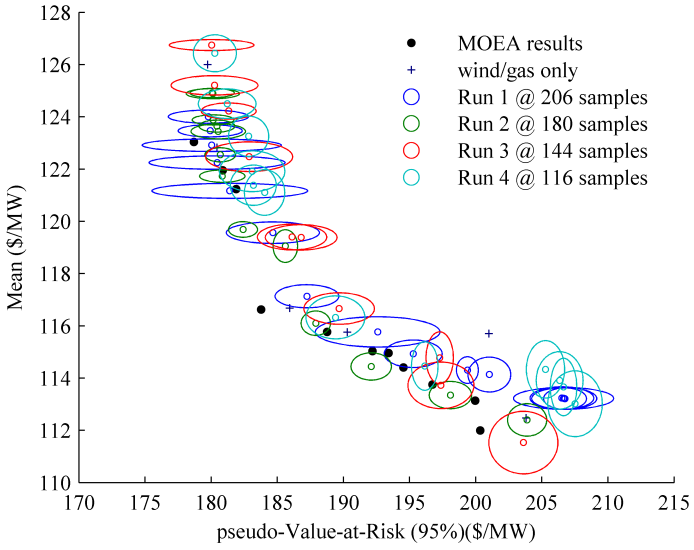
Figure 83(a) shows the estimated Pareto frontier, as reached by four runs of the method, at their points of minimum σ_{total}. The plot also shows the epistemic uncertainty ellipses around those designs. Note that some designs appear to dominate the designs found by the MOEA; it is entirely possible that some do, but it is not possible to tell without running extensive Monte Carlo on the designs, since the plot only shows the *estimated* frontier, according to the best knowledge of the GP. For comparison, Figure 84 shows four of the C-DoE runs on a similar plot, at varying sample sizes. The estimated frontier in these cases were found by exhaustively optimizing the GP, which was inexpensive. Note that even at large sample sizes, the C-DoE has higher epistemic uncertainty, and does not appear to do

Run	Total samples	Min RMSE	@ sample	Min σ_{total}	@ sample	RMSE at min σ_{total}	RMSE % increase
1	837	0.081	701	0.025	206	0.178	120%
2	658	0.077	346	0.031	180	0.127	66%
3	548	0.065	100	0.034	144	0.111	70%
4	358	0.073	182	0.040	116	0.098	35%
5	360	0.106	68	0.050	116	0.142	34%
6	474	0.051	356	0.045	190	0.136	169%
7	496	0.075	426	0.034	186	0.113	51%
8	410	0.072	86	0.041	250	0.133	86%

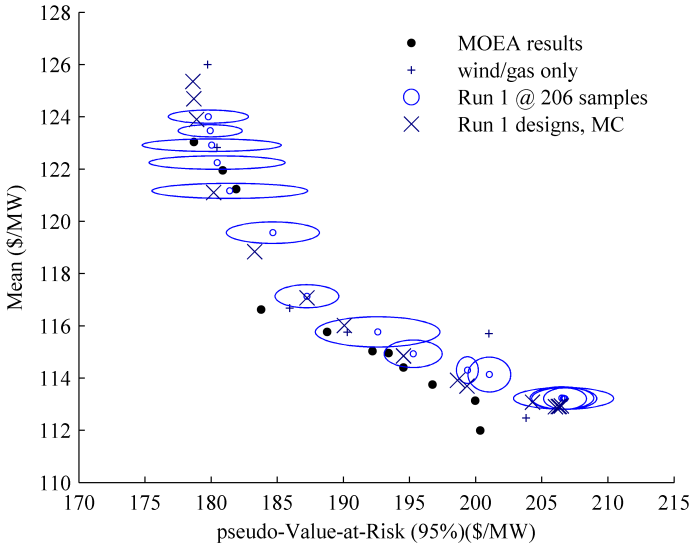
Table 18: RMSE and σ_{total}

as good a job of approximating the frontier.

In an absolute sense, the RMSE stays quite high in all of the cases. Table 18 shows the minimum RMSE reached by the eight test cases, as well as the RMSE reached at their point of minimum σ_{total}. The RMSE tends to hover around 0.1, with the lowest (omnisciently found) values at around 0.05. The sample budgets are on the order of a few hundred. For such low sample budgets, a combined-space DoE could not produce such low RMSE; however, ill-conditioning means that further samples will not help the model. From a theoretical standpoint, C-MOSI is very appealing for its sample efficiency, but ill-conditioning presents a practical limitation, as does the high overhead cost.



(a) Epistemic ellipses



(b) True location

Figure 83: (a) Best C-MOSI estimates of the Pareto frontier, first 4 runs. These are snapshots taken at minimum σ_{total}. The 95% Bayesian epistemic confidence ellipses are shown. Whether any of the designs genuinely dominate the MOEA points cannot be discerned from this graph. (b) The first run has been assessed with 1000-run Monte Carlo (x's), and is shown with its predicted values (ellipses)

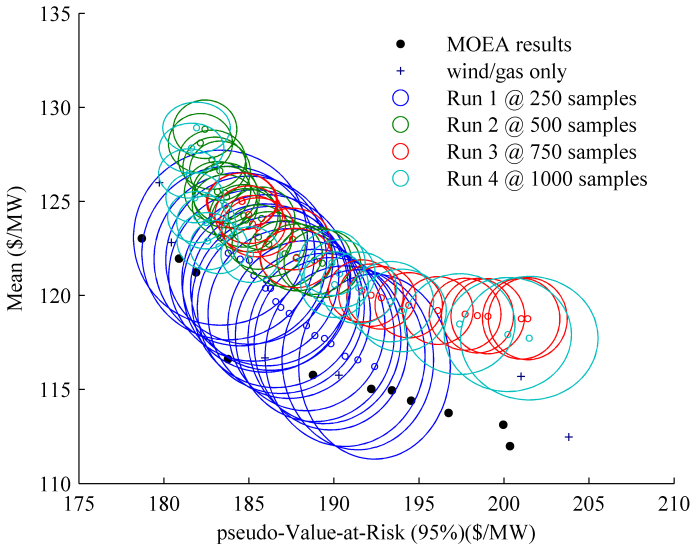


Figure 84: Best C-DoE estimates of the Pareto frontier, four differently sized DoEs. The GP models were exhaustively optimized with an MOEA to find these Pareto sets. The 95% Bayesian epistemic confidence ellipses are shown.

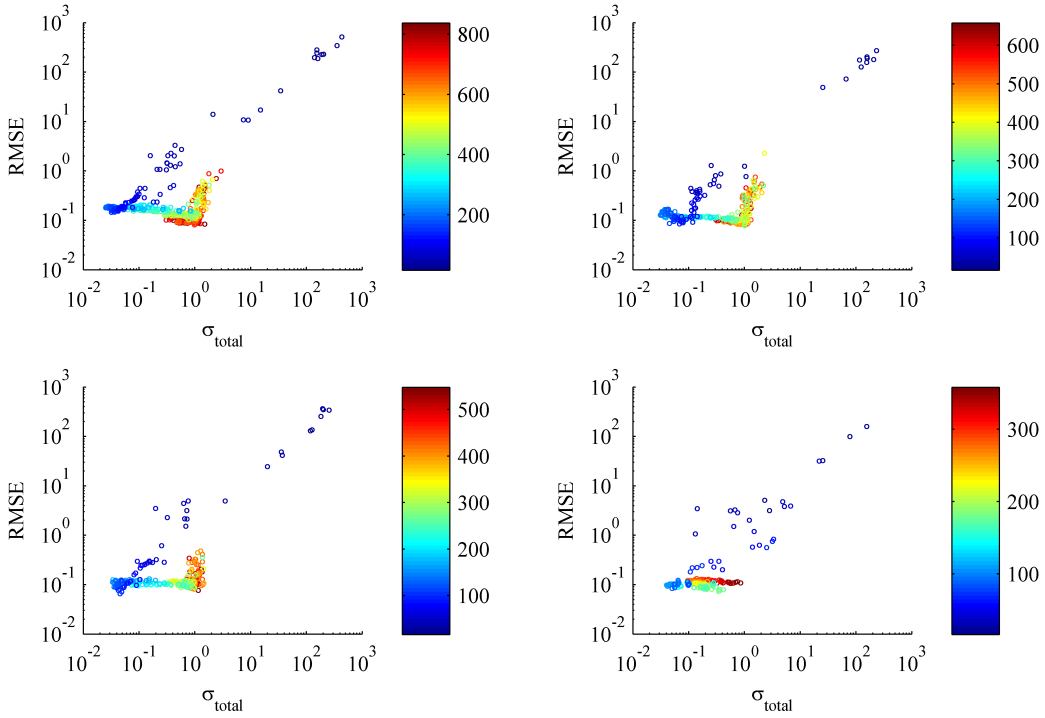


Figure 85: For four C-MOSI runs, RMSE as a function of the root mean epistemic variance along the predicted frontier (σ_{total}, Eq. 131). Color corresponds to number of samples. Epistemic uncertainty roughly correlates with actual error, but the point of minimum uncertainty is not necessarily the point of minimum error.

CHAPTER XI

SUMMARY AND CONCLUSIONS

Electric power portfolio selection was re-cast as a robust design problem. A scalable test problem was developed that mimicked the general behavior of an electric power simulation problem, and this was used to test four robust design methods, shown in Table 19.

The lower-right method, C-MOSI, is not previously found in the literature. This lead to an overall research objective:

Research Objective: Implement multi-objective statistical improvement methods using surrogate models that are functions of both design and noise variables (combined arrays).

Implementing it presented several challenges, which were solved with a combination of existing methods and some new work. New contributions included an extension to O'Hagan's [88] and Apley's [4] works in finding epistemic uncertainty in aleatory uncertainty metrics from GP models: the methods were extended to combined-space models. Additionally, Emmerich *et al.*'s multi-objective expected improvement algorithm [30] was modified to handle uncertain Pareto sets, which encouraged additional sampling near uncertain designs.

Three research questions were raised:

Research Question 1: For finding mean/risk Pareto frontiers, how does the relative efficiency of combined and crossed arrays depend on the number of noise variables?

Table 19: Taxonomy of Methods

	Crossed Array	Combined Array
Design of Experiments	X-DoE	C-DoE
Multi-Objective Statistical Improvement	X-MOSI	C-MOSI

Hypothesis 1: As the number of noise variables increases, the efficiency of combined array methods will suffer relative to the efficiency of crossed array methods.

Research Question 2: For finding mean/risk Pareto frontiers, how does the relative efficiency of design of experiments and multi-objective statistical improvement change with the number of design variables?

Hypothesis 2: As the number of design variables increases, multi-objective statistical improvement methods will become more efficient relative to a design of experiments.

Research Question 3: Is there a design scenario where a combined array Multi-Objective Statistical Improvement method out-performs both crossed-array and design of experiments methods in terms of efficiency?

No hypothesis was formulated for Research Question 3. These research questions were answered with a series of experiments.

11.1 Experiment: Sweep of Warm-Start Size

Two main experiments were conducted. In the first, combined-array (C) methods and crossed-array (X) methods were each treated as continuums. By running a sweep of warm-start DoE size, and then running a MOSI method until it stopped improving, it was shown that the MOSI methods were more efficient than the DoE methods. Efficiency was defined as achieving some level of accuracy along the true Pareto frontier for a particular number of samples. The sweep of warm-start size is shown in Figure 86. It shows that the MOSI methods largely dominate the DoE methods, but for higher numbers of samples all of the methods begin to break down. This was the result of ill-conditioning in the Gaussian Process models.

It was found that for this test problem, the smallest possible warm-start size was always optimal.

11.2 Experiment: Sensitivity of the Four Methods

In a second experiment, all four methods were run at varying numbers of design and noise variables. In each case, the DoE methods were swept from small to large sizes (as in the previous experiment), and the MOSI methods were run 10 times from the smallest possible

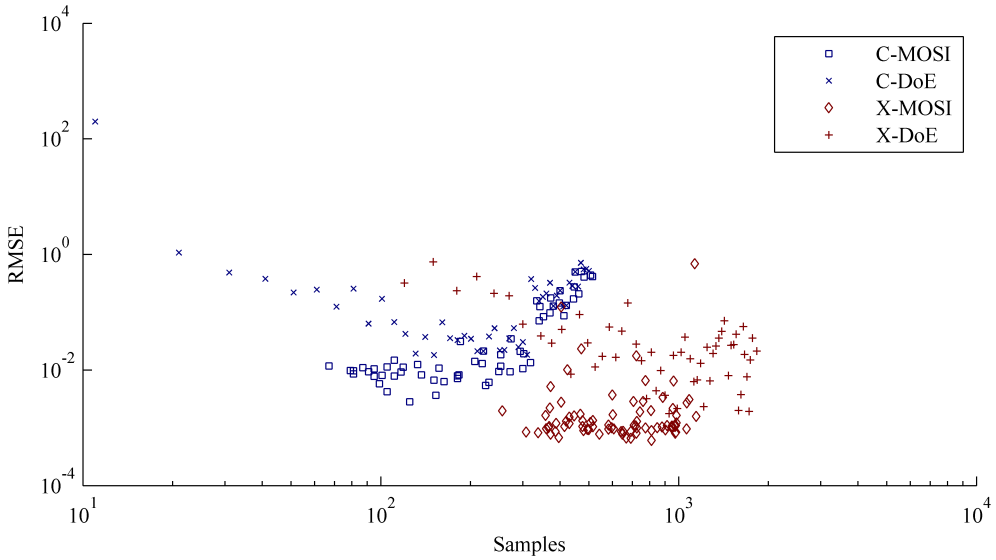


Figure 86: All four methods, after a sweep of warm-start DoE sizes. Every MOSI end-point (box and diamond) is the result of starting from a DoE warm-start (+ and x) and running the method until the RMSE stopped improving. Note that all four methods suffer from ill-conditioning effects at higher numbers of samples; this was both the stopping criteria for the MOSI methods, and the reason for the performance degradation seen above.

warm-starts. Every DoE and every iteration of the MOSI methods was taken as a data point, achieving some level of accuracy (measured by root mean square error along the true Pareto frontier) for some number of samples. All of this data was used to construct a model for how the methods reduced error with samples. A power model was found to fit the data well.

The numbers of design and noise variables were both varied from 2 to 3. This is a small range, much smaller than is likely to be found in a real design problem, but it enabled extensive data collection on the four methods and good fits for the error models. A total of 16 error models were fit, for every combination of array type, sampling approach, number of design variables, and number of noise variables. All 16 models and their underlying data can be seen in Figure 87.

This same data was then used to regress a single unified error model, that described root mean squared error (RMSE) as a function of not only samples, but also array type (X or C), sampling approach (DoE or MOSI), number of design variables (p_D) and number of noise variables (p_S). The interaction terms in this model were used to answer the first two

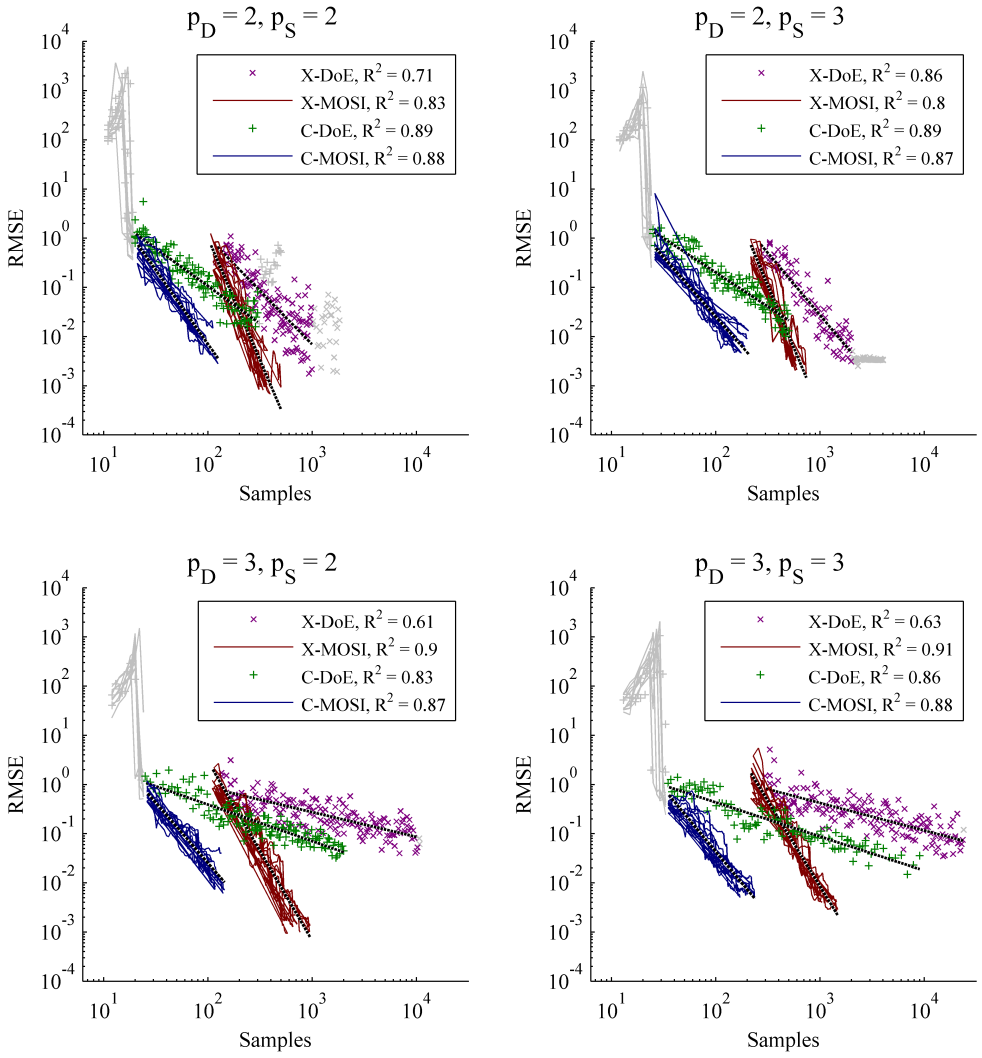


Figure 87: All 16 error models, for every possible combination of array types (X or C), sampling approaches (DoE or MOSI), number of design variables (p_D), and number of noise variables (p_S). Each x-axis shows number of samples, and each y-axis shows root mean squared error (RMSE) along the true Pareto frontier.

research questions, as well as providing additional insight, all of which can be summarized as follows:

- Crossed (X) arrays are more sensitive to design dimensionality than Combined (C) arrays

  but they are less sensitive to noise dimensionality (confirming **Hypothesis 1**)

- DoE sampling is more sensitive to design dimensionality than MOSI sampling (confirming **Hypothesis 2**)

  but it is less sensitive to noise dimensionality

Lastly, it was not possible to give a definitive answer to Research Question 3, partly because was is not possible to give a single metric for "efficiency". It was possible to give a qualified answer, however:

- If the sample budget is below a certain threshold, under certain design scenarios, C-MOSI will be more efficient than C-DoE, X-MOSI, and X-DoE.

- If the required accuracy level is above a certain threshold, under certain design scenarios, C-MOSI will be unable to reduce error sufficiently, and X-MOSI will be more efficient than the other three methods.

Though it should be noted that all of these conclusions may only apply to the scalable test problem, and might not apply generally.

11.3 Demonstration of C-MOSI on an Electric Portfolio Test Problem

Lastly, C-MOSI was demonstrated on a low-fidelity electric power portfolio simulation, and compared with C-DoE runs of varying sizes. Like in the previous experiments, the adaptive sampling approach showed higher efficiency for low sample budgets. As the number of samples increased, however, ill-conditioning in the Gaussian Process surrogates meant that further samples did not improve the model, and the error along the true frontier could not be reduced further. In the simulation test case, however, this effect was more pronounced than in the analytic case, and the benefits of using C-MOSI were relatively less than in the analytic tests.

11.4 When Should C-MOSI be Used?

A final result from the experiments showed that at least for this test problem, C-MOSI was more efficient than the other methods at reducing RMSE for low numbers of samples. For high numbers of samples, ill-conditioning prevented further error reduction, and for high sample budgets X-MOSI was most efficient.

This might imply that C-MOSI should always be used for robust design problems with low sample budgets, but this is not necessarily the case.

First, there are limitations to when the method can be used. In the scalable test problem, the noise variables were made to have Gaussian distributions. This, combined with the use of Gaussian Process models, allowed the use of analytic expressions for second-order uncertainty metrics. These second-order terms are necessary for the use of C-MOSI. There are ways to calculate them using Monte Carlo, when the noise variable distributions are not Gaussian, but this requires *nested* Monte Carlo, which is *extremely* expensive computationally. Though not presented in this document, early experiments with such an approach had poor results because inaccuracies in the Monte Carlo results eliminated any advantage of using MOSI in the first place. In the power portfolio simulation demonstration case, the analytic expressions were used when the noise variables were not Gaussian, by transforming the inputs to the simulation code. This worked, but added complexity to the space being modeled by the GP, and may have contributed to the poorer performance of C-MOSI in that case. This distortion and added complexity will be low if the noise variable distribution is *close* to Gaussian.

There is also an overhead cost associated with C-MOSI. Statistical improvement methods in general have computational overhead that makes them ill-advised in cases where simulations are cheap, partly because the Bayesian models they rely on must be re-fit after every set of samples. C-MOSI has even more computational overhead. Even the analytically computed SOPs incurred a significant cost ($O(n^3)$), and this cost was nested inside of an optimizer, used to select the most promising design. Once a design had been selected, there were still two equally expensive steps that were required to select samples in noise space. Even this least expensive version of C-MOSI required on the order of minutes to

hours to complete each sampling iteration.

Finally, this implementation was limited in its choice of risk metric. The chosen metric was Value-at-Risk, which is just a percentile (here 95th percentile was used). However, because of the analytic SOP calculations, this metric could not be used directly; instead, a metric which was referred to as "pseudo-Value-at-Risk" was used. It was the same as a metric used by Apley *et al.*[4], simply $\mu + c \cdot \sigma$, and with Gaussian output distributions it is equivalent to a percentile. Outputs can never be relied on to be Gaussian, however, and if the response is far from Gaussian this would be a poor choice. To truly adaptively sample for VaR, it would be necessary to resort again to a Monte Carlo approach, and all of the costs associated with it.

Based on these results and experience, then, C-MOSI can be recommended as method for robust design under the following conditions:

1. The noise variables are Gaussian or close to Gaussian, and their distributions are well-known
2. The sample budget is small
3. Every simulation takes on the order of minutes or longer
4. Pseudo-VaR ($\mu + c \cdot \sigma$) is an acceptable risk metric

The first item above has been augmented with the condition that the noise variable distributions are well-known. This is a condition inherent in any adaptive sampling approach to robust design. If the model is only accurate around the Pareto frontier, and the frontier changes (because the noise variable distributions change, for example), then it might not be accurate around the new frontier. That is not to say that such a model could not be updated, and if the changes to the noise variable distributions were small, it might not require very many additional samples.

11.5 Future Work

As might be expected, the investigations in this dissertation raised many more questions than could be answered, and a number of them might prove interesting for future work.

11.5.1 Other Surrogate Models

All of the experiments in this document make use of Gaussian Process models. GPs incur significant regression overhead which increases as the cube of the sample size, and suffer ill-conditioning effects as their samples get close together. Other options exist, such as treed GPs [47] and sparse linear models [110], and their use should be investigated.

11.5.2 Parallelization

Much modern simulation is done with massively parallel computing, but the C-MOSI method as presented here is not trivially parallelizable. Expected improvement methods can be parallelized [43], and for the method to be maximally useful, this should be investigated. Additionally, there is the potential to combine a general C-MOSI framework with other optimization concepts. An approach which was tested and found promising was the use of a multi-points EI criteria combined with a non-nested Monte Carlo approach; such an approach can work with arbitrary noise variable distributions, but further development is required.

11.5.3 Other Risk Measures

The choice of "pseudo-Value-at-Risk" for the test cases in this research allowed for clean analytical results, but it is not a risk metric that is used in often in practice, and it is an imperfect approximation of Value-at-Risk. Metrics such as Value-at-Risk (percentile) or Conditional Value-at-Risk are more commonly used and trusted. More efforts are needed to find analytic, semi-analytic, or efficient numerical estimates of these risk metrics.

11.5.4 Decision Theory Approach

Much work remains in the intersection between statistical improvement methods and robust design. Rather than a multi-objective approach, a decision theory approach can be used with a-priori risk preference elicitation and the use of utility functions. For a particular design, with epistemic uncertainty present in the combined-space surrogate model, there will be *epistemic uncertainty* in the expected utility; this leads naturally to a single-objective expected improvement method. Such an approach was tested, but requires further work.

Such an approach can take one of two possible routes. In the first (which was tested to a limited degree, but not documented) it can use Monte Carlo (including MCMC) methods to estimate the epistemic uncertainty in the expected utility of designs. This would most likely require very large numbers of Monte Carlo samples in order to work effectively. In an alternate approach, analytic expressions could be sought to find the epistemic uncertainty for common utility functions (such as an exponential utility function).

11.5.5 Multiple Stochastic Objectives

Alternately, *more* objectives could be considered, rather than the two included in this work. Generally, there are many possible criteria for use in adaptive sampling methods, and many ways of dealing with uncertainty.

11.5.6 Stochastic Time Series

In the field of power portfolio selection, and many other areas, uncertainty comes in the form both of uncertain variables (as was treated here) and as stochastic time series (such as natural gas price or wind speed series). In the latter case, there may be interest in risk *within* the simulation. Even for aggregated measures the results might be different for simulations run with the same inputs. This latter effect is the domain of *heteroscedastic* problems, and there is much work to be done to integrate this type of problem into a robust design framework.

11.5.7 Avoiding Sampling over Linear Inputs

In the electric power portfolio simulation used in this work, the cost of electricity was linear with respect to several of the inputs, namely natural gas price and market electricity price. In practice, if such a situation existed, it would be inefficient to fit a surrogate model to those inputs. Instead, it would be more sensible to fit surrogates of reduced dimensionality to intermediate variables such as natural gas consumed and market purchases. This would require a re-formulation of the adaptive sampling method.

11.5.8 Efficient Numerical Approaches

This work used a direct modification of existing Statistical Improvement methods. For reasons of computational efficiency, analytical Second Order Probability analysis was used, but this introduced limitations on the conditions under which the method could be used. Those limitations can by bypassed if the designer is willing to incur the computational expense of performing nested Monte Carlo runs. However, there may be other numerical approaches that bypass nested Monte Carlo while still preserving the idea of reducing epistemic uncertainty with regard to aleatory uncertainty metrics. Several variations were attempted (though un-documented) that used single (non-nested) Monte Carlo populations and selected sample points directly from a relatively small population of points; these attempts showed some promise, but require further theoretical development.

11.5.9 Further Real-World Testing

Perhaps most importantly, these methods must be tested on real-world engineering or electric power portfolio problems. The initial results from a simple electric power simulation showed some advantage of C-MOSI over C-DoE, but not to the extent that was demonstrated on a simpler test case. It appears that the relative merits of the two methods are therefore sensitive to algorithm parameters or problem characteristics, and these sensitivities should be investigated.

APPENDIX A

SECOND ORDER PROBABILITY ANALYSIS FOR GAUSSIAN PROCESS MODELS

In order to find expected improvement with a combined design/noise array Gaussian Process, it is first necessary to compute second-order moments, to characterize the epistemic uncertainty in the aleatory moments. The test cases in this document use two objectives: mean μ, and the value $\mu + c \cdot \sigma$, which is referred to in this text as pseudo-Value-at-Risk (pVaR), since in the case of a Gaussian distribution it is the same as a percentile. To find expected improvement in those two metrics, the epistemic mean and variance of the two must be found.

This appendix describes the necessary computational steps, without any derivation. Nearly all of the information in this appendix, along with the relevant derivations, can be found in two sources:

- An un-published paper by O'Hagan [88] presents the computation of $\mathrm{E}[\mu]$, $\mathrm{Var}[\mu]$, $\mathrm{E}[\sigma^2]$, and $\mathrm{Var}[\sigma^2]$.
- An earlier paper by Apley *et al.* [4] provides a less-detailed approach to computing the same terms, and from those terms also provides computation of mean and variance for $f(d) = \mu(d) + c \cdot \sigma(d)$.

This appendix also provides a single correction to O'Hagan's paper, and modifies the expressions to apply in the case of combined arrays.

A.1 O'Hagan's Approach to Calculating SOPs

The following mostly follows O'Hagan's un-published paper [88], omitting all derivations and with one correction. In the original paper, the GP is *only* a function of noise variables, with no design variables at all, which will lead to some differences in the expressions,

usually through the appearance of extra terms to let the deterministic design variables "pass through".

Note also that the form of the emulator used is technically a t-Process (tP) rather than a Gaussian Process (GP) because it has been specified with a global scalar variance multiplier σ^2 that has an inverse-gamma prior and estimated value $\hat{\sigma}^2$.

An asterisk (*) denotes epistemic expectations, variances, and covariances that are with respect to the Gaussian Process emulator, rather than with respect to some externally specified probability distribution.

The aleatory mean and variance with respect to an externally defined uncertainty distribution are denoted here as M and V to match O'Hagan's notation (rather than as μ and σ^2 as previously). They are due to uncertainty on the noise variables, which are assumed to have a probability distribution $g(S)$ that is multivariate Gaussian:

$$g(S) \sim MVN(m, B^{-1}) \tag{149}$$

where m is a column vector of aleatory mean values and B is an aleatory *precision* matrix. Note that all expressions that follow will use the precision matrix rather its inverse, the covariance matrix Σ.

A.1.1 Mean and Covariance Function

It is assumed that there is some true function $f(x)$, and a GP emulator has been fit to it. For any point x, the GP is assumed to provide a mean function $m^*(x) = \mathrm{E}^*[f(x)]$ and covariance function $v^*(x, x') = \mathrm{Cov}^*[f(x), f(x')]$.

The mean function for the tP was presented in Equation 28:

$$\mu_{\hat{T}}(x) = \phi(x)^T \hat{\beta} + \psi^T \Psi (T - \phi(\mathbf{X})\hat{\beta})$$

or, using the notation of O'Hagan,

$$m^*(x) = h(x)^T \hat{\beta} + t(x)^T e$$

$$e = A^{-1}(y - H\hat{\beta})$$

where W is the same as in Chapter , y is the response data vector (called $\mathbf{T}$ previously), H is the design matrix (called ϕ previously), $t(x)$ is the correlation between x and all existing data points (called $\phi(\mathbf{X})$ previously) and A is the correlation matrix (called Ψ previously).

The covariance function for the tP was presented in Equation 30:

$$\begin{aligned}\mathrm{Cov}[\hat{T}(x^{(i)}),\hat{T}(x^{(j)})] = \hat{\sigma}^2[&k(x^{(i)},x^{(j)}) - \psi(x^{(i)})^T\Psi^{-1}\psi(x^{(j)}) \\ &+ \{\phi(x^{(i)}) - G^T\psi(x^{(i)})\}^T W\{\phi(x^{(j)}) - G^T\psi(x^{(j)})\}]\end{aligned}$$

or, using O'Hagan's notation:

$$\begin{aligned}v^*(x,x') =&\hat{\sigma}^2[c(x,x') - t(x)^T A^{-1}t(x') \\ &+ \{h(x) - G^T t(x)\}^T W\{h(x') - G^T t(x')\}]\end{aligned}$$

where $c(x,x')$ is the Gaussian kernel correlation function between two points. In O'Hagan's paper, this takes a Gaussian form with a nugget,

$$c(x,x') = \nu I(x = x') + (1-\nu)\exp\{-(x-x')C(x-x')\} \tag{150}$$

Where $I(x = x')$ is an indicator function that is 1 if $x = x'$, and $\nu \in [0,1]$ is a nugget. This work did not use a nugget ($\nu = 0$), but it will be left in for completeness since it is largely unobtrusive in the expressions.

The term C is a positive definite matrix of correlation parameters, referred to as θ's previously in this document, that are estimated with optimization of a likelihood function. In this work, the C matrix is assumed to be be diagonal. Indeed, the modifications made here to accommodate the presence of both design and noise variables *assume* that C is diagonal, or at least that it can be broken into two separate matrices C_D and C_S for the design and noise variables, respectively. These two are each referred to as matrices primarily to make the notation cleaner.

A.1.2 Required Integrals

The required integrals are:

$$\begin{aligned}
\mathrm{E}^*[M] &= \int m^*(x)dg(x), \\
\mathrm{Var}^*[M] &= \int\int v^*(x,x')dg(x)dg(x'), \\
I_1 &= \int v^*(x,x)dg(x), \\
I_2 &= \int m^*(x)^2dg(x), \\
I_3 &= \int\int v^*(x,x')^2dg(x)dg(x'), \\
I_4 &= \int\int m^*(x)m^*(x')v^*(x,x')dg(x)dg(x'), \\
I_5 &= \int\int\int v^*(x,x')v^*(x,x'')dg(x)dg(x')dg(x''), \\
I_6 &= \int\int m^*(x)v^*(x,x')dg(x)dg(x'). \qquad (151)
\end{aligned}$$

and in terms of these integrals, the statistics for σ^2 are:

$$\begin{aligned}
\mathrm{E}^*[\sigma^2] =& (I_1 - \mathrm{Var}^*[M]) + (I_2 - \mathrm{E}^*[M]^2) \\
\mathrm{Var}^*[V] =& 2(I_3 - 2I_5 + \mathrm{Var}^*[M]^2) \\
&+ 4(I_4 - 2\mathrm{E}^*[M]I_6 + \mathrm{E}^*[M]^2\mathrm{Var}^*[M]) \\
&+ \frac{2}{N-L-4}\{2(I_3 - 2I_5 + \mathrm{Var}^*[M]^2) + (I_1 - \mathrm{Var}^*[M])^2\} \qquad (152)
\end{aligned}$$

The last line is due to the emulator being a t-Process (tP) rather than a Gaussian Process (GP). N is the number of data points, and L is the number of basis vectors in the linear model prior.

The eight integrals in 151 are evaluated in terms of 14 additional expressions, using

O'Hagan's notation:

$$
\begin{aligned}
R_h &= \int h(x)dg(x) \\
R_t &= \int t(x)dg(x) \\
R_{hh} &= \int h(x)h(x)^T dg(x) \\
R_{ht} &= \int h(x)t(x)^T dg(x) \\
R_{tt} &= \int t(x)t(x)^T dg(x) \\
U &= \int\int c(x,x')dg(x)dg(x') \\
U_h &= \int\int h(x)c(x,x')dg(x)dg(x') \\
U_t &= \int\int t(x)c(x,x')dg(x)dg(x') \\
U_{hh} &= \int\int h(x)c(x,x')h(x')^T dg(x)dg(x') \\
U_{ht} &= \int\int h(x)c(x,x')t(x')^T dg(x)dg(x') \\
U_{tt} &= \int\int t(x)c(x,x')t(x')^T dg(x)dg(x') \\
\tilde{U} &= \int c(x,x)dg(x) \\
S &= \int\int\int c(x,x')c(x,x'')dg(x)dg(x')dg(x'') \\
\tilde{S} &= \int\int c(x,x')^2 dg(x)dg(x') \qquad (153)
\end{aligned}
$$

Expanding 151 in terms of 153 leads to:

$$
\begin{aligned}
\mathrm{E}^*[M] &= R_h^T\hat{\beta} + R_t^T e \\
\mathrm{Var}^*[M] &= \hat{\sigma}^2[U - R_t^T A^{-1} R_t + (R_h - G^T R_t)^T W (R_h - G^T R_t)] \\
I_1 &= \hat{\sigma}^2[\tilde{U} - \mathrm{tr}(A^{-1}R_{tt}) + \mathrm{tr}(W(R_{hh} - 2R_{ht}G + G^T R_{tt}G))] \\
I_2 &= \hat{\beta}^T R_{hh}\hat{\beta} + 2\hat{\beta}^T R_{ht} e + e^T R_{tt} e
\end{aligned}
$$

$$
\begin{aligned}
I_3 = \hat{\sigma}^4[\tilde{S} &- 2\mathrm{tr}(A^{-1}U_{tt}) + \mathrm{tr}(A^{-1}R_{tt}A^{-1}R_{tt}) + 2\mathrm{tr}(W(U_{hh} - 2U_{ht}G + G^T U_{tt}G)) \\
&- 2\mathrm{tr}(A^{-1}(R_{ht} - G^T R_{tt})^T W (R_{ht} - G^T R_{tt})) \\
&+ \mathrm{tr}(W(R_{hh} - 2R_{ht}G + G^T R_{tt}G) W (R_{hh} - 2R_{ht}G + G^T R_{tt}G))]
\end{aligned}
$$

$$
\begin{aligned}
I_4 = \hat{\sigma}^2[&\hat{\beta}^T U_{hh}\hat{\beta} + 2\hat{\beta}^T U_{ht}e + e^T U_{tt}e \\
&-\hat{\beta}^T R_{ht}A^{-1}R_{ht}^T\hat{\beta} - 2\hat{\beta}^T R_{ht}A^{-1}R_{tt}e - e^T R_{tt}A^{-1}R_{tt}e \\
&+(R_{hh}\hat{\beta} - G^T R_{ht}^T\hat{\beta} + R_{ht}e - G^T R_{tt}e)^T \\
&\quad W(R_{hh}\hat{\beta} - G^T R_{ht}^T\hat{\beta} + R_{ht}e - G^T R_{tt}e)]
\end{aligned}
$$

$$
\begin{aligned}
I_5 = \hat{\sigma}^4[&S - 2R_t^T A^{-1}U_t + R_t^T A^{-1}R_{tt}A^{-1}R_t \\
&+ 2(U_h - G^T U_t)^T W(R_h - G^T R_t) \\
&- 2R_t^T A^{-1}(R_{ht}^T - R_{tt}G)W(R_h - G^T R_t) \\
&+ (R_h - G^T R_t)^T W(R_{hh} - 2_{ht}G + G^T R_{tt}G)W(R_h - G^T R_t)]
\end{aligned}
$$

$$
\begin{aligned}
I_6 = \hat{\sigma}^2[&\hat{\beta}^T U_h - \hat{\beta}^T R_{ht}A^{-1}R_t + \hat{\beta}^T(R_{hh} - R_{ht}G)W(R_h - G^T R_t) \\
&+ e^T U_t - e^T R_{tt}A^{-1}R_t + e^T(R_{ht} - G^T R_{tt})^T W(R_h - G^T R_t)]
\end{aligned} \tag{154}
$$

These are the expressions necessary to compute the four second-order statistics, with expressions for the terms to be presented shortly. Note that wherever a trace of a product is taken (especially in term I_3), rather than multiplying out the full expressions, it is faster to take advantage of the fact that not all information from the product is needed:

$$
\mathrm{tr}(X^T Y) = \sum_{i,j} X_{i,j} Y_{i,j} \tag{155}
$$

A.1.3 The R Integrals

R_h and R_{hh} are expectations with respect to the noise variable distributions, denoted in O'Hagan's paper by $\mathrm{E}_X[\cdot|m, B]$ since X is a random variable in O'Hagan's paper. In this case, X is composed of a random component S and a deterministic design variable component D, so that $X = [D, S]$ for the whole dataset or $x = [d, s]$ for a single data point. The notation and expressions do not change very much, except that the design variables d will essentially "pass through" the expressions.

$$
\begin{aligned}
R_h &= \mathrm{E}_S[h(x)|m, B] \\
R_{hh} &= \mathrm{E}_S[h(x)|m, B]
\end{aligned} \tag{156}
$$

If the prior linear model has only a constant term and linear terms, as was used in this work, and the design vector $h(x)$ is

$$h(x) = \begin{bmatrix} 1 \\ d \\ s \end{bmatrix} \tag{157}$$

where d and s are column vectors of the design and noise variable values, then

$$R_h = \begin{bmatrix} 1 \\ d \\ m \end{bmatrix} \tag{158}$$

$$R_{hh} = \begin{bmatrix} 1, & d^T, & m^T \\ d, & d\,d^T, & d\,m^T \\ m, & m\,d^T, & mm^T + B^{-1} \end{bmatrix} \tag{159}$$

This differs from O'Hagan's expressions in the addition of d and associated terms.

Many of the remaining terms will follow a consistent format. There will be $Q_S(s)$ expressions, which will be used to evaluate a specific modified mean vector m'. There will also be a $Q_D(d)$ expression, not found in O'Hagan's paper, to deal with the deterministic design variables that are just "passing through". Both $Q(x)$ expressions must be computed for every data point $x_k, k \in (1 \cdots N)$. For R_t and R_{ht}, the terms are:

$$Q_{S,k}(s) = 2(s - s_k)^T C_S (s - s_k) + (s - m)^T B (s - m) \tag{160}$$

$$m'_k = (2C + B)^{-1}(2C_S\ s_k + Bm) \tag{161}$$

$$Q_{D,k}(d) = (d - d_k)^T C_D (d - d_k) \tag{162}$$

The k-th element of $N \times 1$ vector R_t, and the k-th column of $N \times p_S$ matrix R_{ht} are

$$R_t(k) = (1 - \nu)|B|^{1/2}|2C_S + B|^{-1/2} \exp\{-Q_{S,k}(m'_k)/2 - Q_{D,k}(d)\} \tag{163}$$

$$R_{ht}(k) = R_t(k)\ \mathrm{E}_S[h(x)|m'_k, 2C + B] = R_t(k) \begin{bmatrix} 1 \\ d \\ m'_k \end{bmatrix} \tag{164}$$

These expressions differ from O'Hagan's due to the addition of the $Q_{D,k}(d)$ term and the d term. In practice, all $Q_S(s)$ terms can be pre-computed once every time the dataset is updated, and only the $Q_D(d)$ terms need to be computed individually for every un-sampled design.

The last R term, R_{tt}, is computed with

$$Q_{S,kl}(s) = 2(s - s_k)^T C_S (s - s_k) + 2(s - s_l)^T C_S (s - s_l) + (s - m)^T B (s - m) \tag{165}$$

$$m'_{kl} = (4C_S + B)^{-1}(2C_S s_k + 2C_S s_l + Bm) \tag{166}$$

$$Q_{D,kl}(d) = (d - d_k)^T C_D (d - d_k) + (d - d_l)^T C_D (d - d_l) \tag{167}$$

$$R_{tt}(k,l) = (1-\nu)^2 |B|^{1/2} |4C_S + B|^{-1/2} \exp\{-Q_{S,kl}(m'_{kl})/2 - Q_{D,kl}(d)\} \tag{168}$$

In practice, if the C terms are diagonal, it is possible to write faster-executing versions of the above expressions, but the above form is retained for clarity. Additionally, in MATLAB there are significant benefits from using vectorized expressions, and these expressions were completely re-written, with all of the terms distributed, to eliminate looping and speed up the computations. Those forms are not presented here, as they are cumbersome.

A.1.4 The U Integrals

Skipping all derivation (at the risk of adding confusion due to lack of context), the U-terms are:

$$U = (1-\nu)|B||\mathbf{B}|^{-1/2} \tag{169}$$

$$U_h = U \begin{bmatrix} 1 \\ d \\ m \end{bmatrix} \tag{170}$$

$$U_{hh} = U \begin{bmatrix} 1, & d^T, & m \\ d, & d\,d^T, & d\,m \\ m, & m\,d^T, & m\,m^T + (\mathbf{B}^{-1})' \end{bmatrix} \tag{171}$$

where

$$\mathbf{B} = \begin{bmatrix} 2C_S + B & -2C_S \\ -2C_S & 2C_S + B \end{bmatrix} \tag{172}$$

and $(\mathbf{B}^{-1})'$ is the lower left **submatrix** of the inverse of $\mathbf{B}$. The term m in the above expressions is actually a sub-vector from a vector O'Hagan calls $\mathbf{m}$, but it is equal to m.

Next,

$$Q^u_{S,k}(\mathbf{s}) = 2(s' - s_k)^T C_S (s' - s_k) + 2(s - s')^T C_S (s - s') \tag{173}$$

$$+ (s - m)^T B (s - m) + (s' - m)^T B (s' - m) \tag{174}$$

$$U_t(k) = (1 - \nu)^2 |B| |\mathbf{B}_k|^{-1/2} \exp\{-Q^u_{S,k}(\mathbf{m}'_k)/2 - Q_{D,k}(d)\} \tag{175}$$

where $Q_{D,k}(d)$ is as in R_t. Boldface $\mathbf{s}$ is just a $p_S \times 1$ concatenated vector,

$$\mathbf{s} = \begin{bmatrix} s \\ s' \end{bmatrix} \tag{176}$$

and

$$\mathbf{m}'_k = \mathbf{B}_k^{-1} \begin{bmatrix} Bm \\ 2C_S s_k + Bm \end{bmatrix}, \quad \mathbf{B}_k = \begin{bmatrix} 2C_S + B & -2C_S \\ -2C_S & 4C_S + B \end{bmatrix} \tag{177}$$

Next,

$$U_{ht}(k) = U_t \, \mathrm{E}_S[h(X)|\mathbf{m}'_k, \mathbf{B}_k] = \begin{bmatrix} 1 \\ d \\ (\mathbf{m}'_k)' \end{bmatrix} \tag{178}$$

where $(\mathbf{m}'_k)'$ is just a cumbersome way of signifying the upper $p_S \times 1$ sub-vector from $\mathbf{m}'_k$.

Last of the U integrals, the expression for U_{tt} in O'Hagan has an error, and should be

$$Q^u_{S,kl}(\mathbf{s}) = 2(s - s_k)^T C_S (s - s_k) + 2(s' - s_l)^T C_S (s' - s_l) \tag{179}$$

$$+ 2(s - s')^T C_S (s - s') \tag{180}$$

$$+ (s - m)^T B (s - m) + (s' - m)^T B (s' - m) \tag{181}$$

$$= (\mathbf{s} - \mathbf{m}'_{kl})^T \mathbf{B}_{kl} (\mathbf{s} - \mathbf{m}'_{kl}) + Q^u_{S,kl}(\mathbf{m}'_{kl}) \tag{182}$$

$$U_{tt}(k, l) = (1 - \nu)^3 |B| |\mathbf{B}_{kl}|^{-1/2} \exp\{-Q^u_{S,kl}(\mathbf{m}'_{kl})/2 - Q_{D,kl}(d)\} \tag{183}$$

where

$$\mathbf{m}'_{kl} = \mathbf{B}_{kl}^{-1} \begin{bmatrix} 2C_S s_k + Bm \\ 2C_S s_l + Bm \end{bmatrix}, \quad \mathbf{B}_{kl} = \begin{bmatrix} 4C_S + B & -2C_S \\ -2C_S & 4C_S + B \end{bmatrix} \tag{184}$$

(To use the previous expression in the context of O'Hagan's paper, with no design variables, simply replace the s's with x's, remove all S subscripts, and remove the $Q_{D,kl}(d)$ term.)

A.1.5 The S Integrals

Lastly, the two S integrals are

$$S = (1-\nu)^2|B|^{3/2}\begin{vmatrix} 4C_S+B & -2C_S & -2C_S \\ -2C_S & 2C_S+B & 0 \\ -2C_S & 0 & 2C_S+B \end{vmatrix}^{-1/2} \tag{185}$$

$$\tilde{S} = (1-\nu)|B|\begin{vmatrix} 4C_S+B & -4C_S \\ -4C_S & 4C_S+B \end{vmatrix}^{-1/2} \tag{186}$$

All together, the expressions in the previous section provide all information necessary to compute the second-order statistics E[M], Var[M], E[V], and Var[V]. In MATLAB, where vectorization results in significant speedup over looping, considerable time savings can be found by replacing all looping over k and l with non-looped expressions. Since MATLAB cannot perform matrix operations on multi-dimensional arrays, this requires expansion of all matrix polynomials, and results in considerably more complex expressions, which are not provided here, but can be provided upon request.

A.2 Apley's Objective Function

The objectives used in the test cases were mean and an objective used by Apley [4],

$$f(d) \equiv \mu(d) + c \cdot \sigma(d) \tag{187}$$

where $\mu(d)$ and $\sigma^2(d)$ are the aleatory mean and variance due to the noise variable distributions (the same as M and V in O'Hagan's notation). $f(d)$ was referred to previously as "pseudo-Value-at-Risk" since it's the same as a percentile when the aleatory response distribution is Gaussian. When there is epistemic emulator uncertainty, the second-order

statistics for this function are

$$\mu_f(d) = \mu_\mu(d) + c \cdot \mu_\sigma(d) \tag{188}$$

$$\sigma_f^2(d) = \sigma_\mu^2(d) + c^2\sigma_\sigma^2(d) + 2c\text{Cov}[\mu(d), \sigma(d)] \tag{189}$$

where in O'Hagan's notation, σ_μ^2 is $\text{Var}[M]$, etc.

Apley points out that assuming the epistemic distribution $\sigma(d)$ is Gaussian is a better assumption than assuming $\sigma^2(d)$ is Gaussian, and if this assumption is made, it is possible to find the statistics

$$\mu_\sigma(d) = [\mu_V^2 - \text{Var}[V]/2]^{1/4} \tag{190}$$

$$\sigma_\sigma^2(d) = \mu_V - [\mu_V^2 - \text{Var}[V]/2]^{1/2} \tag{191}$$

where the notation is getting messy (and Apley uses S instead of V), but hopefully the intent is clear.

Importantly, Apley performs the same set of derivations as O'Hagan (presumably, though much is left un-said in the paper) and additionally arrives at an expression for $\text{Cov}[\mu(d), V]$. It is provided here using O'Hagan's notation and terms:

$$\text{Cov}[M, V] = \text{E}[M]\left(I_1 + I_2 - \text{E}[M]^2 - 3\text{Var}[M]\right) + 2I_6 - \text{E}[M]\text{E}[V] \tag{192}$$

Armed with $\text{Cov}[M, V]$, it is possible to go back to Equation 189 and find the epistemic variance of the pseudo-Value-at-Risk. This was the last missing piece of the SOP puzzle, and now it is possible to find the multi-objective expected improvement in both aleatory mean and aleatory pseudo-Value-at-Risk for a combined-array Gaussian Process.

REFERENCES

[1] Ahmed, S. and Husseiny, A. A., "A multivariate-utility approach for selection of energy sources," *Energy*, vol. 3, pp. 669–700, 1978.

[2] Ameren Missouri, "Integrated resource plan." Report, 2011.

[3] Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I., "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, pp. 5–23, 2003.

[4] Apley, D. W., Liu, J., and Chen, W., "Understanding the effects of model uncertainty in robust design with computer experiments," *Journal of Mechanical Design*, vol. 128, p. 945, 2006.

[5] Arrow, K. J., "A difficulty in the concept of social welfare," *The Journal of Political Economy*, vol. 58, no. 4, pp. 328–346, 1950.

[6] Artzner, P., Delbaen, F., Eber, J., and Heath, D., "Coherent measures of risk," *Mathematical Finance*, vol. 9(3), p. 203, 1999.

[7] Atmospheric Science Data Center.

[8] Avista, "Electric integrated resource plan." Report, August 2009.

[9] Baleriaux, H., Jamoulle, E., and de Guertechin, F. L., "Simulation de l'exploitation d'un parc de machines thermiques de production d'electricite couple a des stations de pomage," *Revue E (edition S.R.B.E.)*, vol. V, 7, pp. 3–24, 1967.

[10] Bandte, O., Mavris, D. N., and DeLaurentis, D. A., "Viable designs through a joint probabilistic estimation technique," *SAE Transactions*, vol. 108(1), pp. 1365–1377, 1999.

[11] Bar-Lev, D. and Katz, S., "A portfolio approach to fossil fuel procurement in the electric utility industry," *The Journal of Finance*, vol. 31(3), pp. 933–947, 1976.

[12] Barbose, G., "Managing carbon regulatory risk in utility resource planning: Current practices in the western united states," *Energy policy*, vol. 36(9), p. 3300, 2009.

[13] Barbose, G., Wiser, R., Phadke, A., and Goldman, C., "Reading the tea leaves: How utilities in the west are managing carbon regulatory risk in their resource plans," tech. rep., U.S. Department of Energy, 2008.

[14] Bates, R. A., Kenett, R. S., Steinberg, D. M., and Wynn, H. P., "Achieving robust design from computer simulations," *Quality Technology and Quantitative Management*, vol. 3(2), pp. 161–177, 2006.

[15] Bautista, D. C., *A Sequential Design for Approximating the Pareto Front Using the Expected Pareto Improvement Function.* PhD thesis, The Ohio State University, 2009.

[16] Bernoulli, D., "Exposition of a new theory on the measurement of risk," *Papers of the Imperial Academy of Sciences in Petersburg*, vol. V, pp. 175–192, 1738. Translated from Latin into English by Dr. Louise Sommer.

[17] Beyer, H.-G. and Sendhoff, B., "Robust optimization a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 3190–3218, 2007.

[18] Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] Booth, R. R., "Power system simulation model based on probability analysis," in *IEEE transactions on power apparatus and systems*, 1972.

[20] Chafekar, D., Shi, L., Rasheed, K., and Xuan, J., "Multiobjective ga optimization using reduced models," *IEEE Transactions on Sysems, Man, and Cybernetics*, vol. 35(2), p. 261, 2005.

[21] Chen, W., Allen, J., Tsui, K., and Mistree, F., "A procedure for robust design: Minimizing variations caused by noise factors and control factors," *ASME Journal of Mechanical Design.*, vol. 118(4), p. 478, 1996.

[22] Chen, W., Tsui, K., Allen, J., and Mistree, F., "Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure," in *Design Automation Conference, Boston, MA*, ASME, 1995.

[23] Cox, D. D. and John, S., "Sdo: A statistical method for global optimization," in *in Multidisciplinary Design Optimization: State-of-the-Art*, pp. 315–329, 1997.

[24] Currin, C., Mtichell, T., Morris, M., and Ylvisaker, D., "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, vol. 86(416), pp. 953–963, 1991.

[25] Daskilewicz, M. J. and German, B. J., "Observations on the topology of pareto frontiers with implications for design decision making," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition.*, 2012.

[26] Daskilewicz, M. J., German, B. J., Takahashi, T., and Donovan, S., "Considering uncertainty quantification in the multi-objective problem of aircraft conceptual design," *Structural and Multidisciplinary Optimization*, vol. ?, p. ?, 2011. Pre-print.

[27] De Simone, L. E., "A strategic planning framework for energy utilities in an era of uncertainty and capital constraints," in *Energy Modeling III*, 1980.

[28] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6(2), p. 182, 2002.

[29] Eldred, M. S. and Swiler, L. P., "Efficient algorithms for mixed aleatory-epistemic uncertainty quantification with application to radiation-hardened electronics," tech. rep., Sandia National Laboratories, 2009.

[30] Emmerich, M., Deutz, A., and Klinkenberg, J.-W., "The computation of the expected improvement in dominated hypervolume of pareto front approximations," tech. rep., Leiden Institute for Advanced Computer Science, 2008.

[31] Emmerich, M. a., "Source code and software." Web.

[32] Emmerich, M. T. M., Giannakoglou, K. C., and Naujoks, B., "Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10(4), p. 421, 2006.

[33] Energy Information Administration, "The national energy modeling system: An overview," tech. rep., Energy Information Administration, 2009.

[34] EnerNex Corporation, "Eastern wind integration and transmission study," tech. rep., National Renewable Energy Laboratory, 2010.

[35] Entergy, "Integrated resource plan." Report, 2010.

[36] Farina, M., "A neural network based generalized response surface multiobjective evolutionary algorithm," in *IEEE*, 2002.

[37] Florida Power and Light, "Ten year power plant site plan." Report, April 2011.

[38] Forrester, A. I. J., Sobester, A., and Keane, A. J., *Engineering Design via Surrogate Modelling: A Practical Guide*. Progress in Astronautics and Aeronautics, Wiley, 2008.

[39] Forrester, A. I. J., Sobester, A., and Keane, A. J., "Website code," November 2010. Accompanies "Engineering Design via Surrogate Modelling".

[40] Frey, D. D. and Li, X., "Validating robust-parameter-design methods," in *Proceedings of DETC'04*, (Salt Lake City, Utah, USA), October 2004.

[41] Gaspar-Cunha, A. and Vieira, A., "A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations," *International Journal of Computers, Systems and Signals*, vol. 6(1), p. 18, 2005.

[42] GE Energy, "Western wind and solar integration study," tech. rep., National Renewable Energy Laboratory, 2010.

[43] Ginsbourger, D., Le Riche, R., and Carraro, L., "A Multi-points Criterion for Deterministic Parallel Global Optimization based on Kriging," in *Intl. Conf. on Nonconvex Programming, NCP07*, (Rouen, France), p. ..., Dec. 2008.

[44] Gotham, D., Muthuraman, K., Preckel, P., Rardin, R., and Ruangpattana, S., "A load factor based meanvariance analysis for fuel diversification," *Energy Economics*, vol. 31, pp. 249–256, 2009.

[45] Graham, V. A. and Hollands, K. G. T., "A method to generate synthetic hourly solar radiation globally," *Solar Energy*, vol. 44, pp. 333–341, 1990.

[46] Graham, V. A., Hollands, K. G. T., and Unny, T. E., "A time series model for k_t with application to global synthetic weather generation," *Solar Energy*, vol. 40, pp. 83–92, 1988.

[47] Gramacy, R. B. and Taddy, M., "Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an r package for treed gaussian process models," *Journal of Statistical Software*, vol. 33, 2010.

[48] Hawe, G. I. and Sykulski, J. K., "An enhanced probability of improvement utility function for locating pareto optimal solutions," in *16th Conference on the Computation of Electromagnetic Fields COMPUMAG*, 2007.

[49] Hazelrigg, G. A., *Fundamentals of Decision Making for Engineering Design and Systems Engineering*. 2012.

[50] Hollands, K. G. T. and Huget, R. G., "A probability density function for the clearness index, with applications," *Solar Energy*, vol. 30, pp. 195–209, 1983.

[51] Howard, R. A., "The foundations of decision analysis," *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-4, pp. 211–219, 1968.

[52] Humphreys, H. B. and McClain, K. T., "Reducing the impacts of energy price volatility through dynamic portfolio selection," *The Energy Journal*, vol. 19(3), p. 107, 1998.

[53] Idaho Power, "Integrated resource plan." Report, June 2011.

[54] Jin, R., Chen, W., and Simpson, T. W., "Comparative studies of metamodelling techniques under multiple modelling criteria," *Structural and Multidisciplinary Optimization*, vol. 23, pp. 1–13, 2001.

[55] Jin, Y., "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, pp. 3–12, 2005.

[56] Jin, Y. and Sendhoff, B., "Trade-off between performance and robustness: An evolutionary multiobjective approach," in *Evolutionary Multi-Criterion Optimization*, 2003.

[57] Jones, D. R., Schonlau, M., and Welch, W., "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.

[58] Jorion, P., *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, 2001.

[59] Kaplan, S. and Garrick, B. J., "On the quantitative definition of risk," *Risk Analysis*, vol. 1(1), p. 11, 1981.

[60] Keane, A. J., "Statistical improvement criteria for use in multiobjective design optimization," *AIAA Journal*, vol. 44(4), p. 879, 2006.

[61] Keane, A. J. and Nair, P., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. Wiley, 2005.

[62] Keeney, R. L., Beley, J. R., Fleischauer, P., Kirkwood, C. W., and Sicherman, A., "Decision framework for techology choice - volume 1. a case study of one utility's coal-nuclear choice," tech. rep., Electric Power Research Institute, 1981.

[63] Keeney, R. L., Lathrop, J. F., and Sicherman, A., "An analysis of baltimore gas and electric company's technology choice," *Operations Research*, vol. 34, no. 1, pp. 18–39, 1986.

[64] Keeney, R. L. and Raiffa, H., *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.

[65] Knowles, "Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10(1), p. 50, 2005.

[66] Korpas, M., Warland, L., Tande, J. O. G., Uhlen, K., Purchala, K., and Wagemans, S., "Tradewind: Further developing europe's power market for large scale integration of wind power," tech. rep., European Wind Energy Association, 2007.

[67] Kumar, A., *Robust Design Methodologies: Application to Compressor Blades*. PhD thesis, University of Southampton, 2006.

[68] Kumar, A., Nair, P. B., Keane, A. J., and Shahpar, S., "Robust design using bayesian monte carlo," *International Journal for Numerical Methods in Engineering*, vol. 73, pp. 1497–1517, 2007.

[69] Lehman, J. S., Santner, T. J., and Notz, W. I., "Designing computer experiments to determine robust control variables," *Statistica Sinica*, vol. 14, pp. 571–590, 2004.

[70] Lilienthal, P., Gilman, P., and Lambert, T., *HOMER 2.19*. National Renewable Energy Laboratory, 2005. Computer program documentation.

[71] Lin, M., Breipohl, A., and Lee, F., "Comparison of probabilistic production cost simulation methods," *IEEE Transactions on Power Systems*, vol. 4(4), p. 1326, 1989.

[72] Lucas, J. M., "How to achieve a robust process using response surface methodology," *Journal of Quality Technology*, vol. 26(4), p. 248, 1994.

[73] Luce, R. D. and Raiffa, H., *Games and Decisions*. New York: John Wiley & Sons, 1957.

[74] MacKay, D. J. C., *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[75] Markowitz, H., "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[76] Mavris, D., Bandte, O., and DeLaurentis, D. A., "Robust design simulation: A probabilistic approach to multidisciplinary design," *Journal of Aircraft*, vol. 36(1), p. 298, 1999.

[77] McKay, M. D., Beckman, R. J., and Conover, W. J., "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21(2), p. 239, 1979.

[78] Milligan, M. R., "Modelling utility-scalewind power plants. part 2: Capacity credit," *Wind Energy*, vol. 3, p. 167, 2000.

[79] Mockus, J., Tiesis, V., and Zilinskas, A., "The application of bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, pp. 117–129, 1978.

[80] Morris, M. and Mitchell, T. J., "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, vol. 43, pp. 381–402, 1995.

[81] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. Wiley, 2 ed., 2002.

[82] Nain, P. K. S. and Deb, K., "Computationally effective search and optimization procedure using coarse to fine approximations," in *IEEE*, 2003.

[83] National Renewable Energy Laboratory, "United states photovoltaic solar resource: Flat plate tilted at latitude."

[84] NorthWestern Energy, "Electric supply resource procurement plan." Report, June 2009.

[85] Oakley, J. E., "Estimating percentiles of uncertain computer code outputs," *Applied Statistics*, vol. 53(1), pp. 83–93, 2004.

[86] Oakley, J. E., "Decision-theoretic sensitivity analysis for complex computer models," *Technometrics*, vol. 51(2), p. 121, 2009.

[87] Oakley, J. E. and O'Hagan, A., "Bayesian inference for the uncertainty distribution of computer model outputs," *Biometrika*, vol. 89(4), p. 769, 2002.

[88] O'Hagan, A., "Uncertainty analysis: the variance of the variance." Available online, Managing Uncertainty in Complex Models (MUCM) toolkit., 2011.

[89] Pacific Gas & Electric, "Integrated resource planning at pg&e." Presentation, June 2007.

[90] PacifiCorp, "Integrated resource plan." Report, March 2011.

[91] Patel, C. B., *A Multi-Objective Stochastic Approach to Combinatorial Technology Space Exploration*. PhD thesis, Georgia Institute of Technology, 2009.

[92] PCI, "Pci gentrader." Website, 2011. Energy portfolio optimization software.

[93] Progress Energy, "Progress energy carolinas integrated resource plan." Report, September 2009.

[94] Pruzan, P. M. and Jackson, J. T. R., "On the development of utility spaces for multi-goal systems," *Ledelse og Erhvervskonomi/Handelsvidenskabeligt Tidsskrift/Erhvervskonomisk Tidsskrift*, vol. 27, 1963.

[95] Puget Sound Energy, "Integrated resource plan." Report, 2011.

[96] Rasmussen, C. E. and Ghahramani, Z., "Bayesian monte carlo," *Advances in neural information processing systems*, vol. 15, p. 489, 2003.

[97] Reyes-Sierra, M. and Coello Coello, C. A., "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research.*, vol. 2(3), pp. 287–308, 2006.

[98] Sacks, J., Welch, W., Mitchell, T. J., and Wynn, H. P., "Design and analysis of computer experiments," *Statistical Science*, vol. 4(4), pp. 409–423, 1989.

[99] Schonlau, M., *Computer Experiments and Global Optimizat ion.* PhD thesis, University of Waterloo, 1997.

[100] Sharma, S., Ukkusori, S. V., and Mathew, T. V., "Pareto optimal multiobjective optimization for robust transportation network design problem," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2090, p. 95, 2009.

[101] Sharpe, W. F., "Capital asset prices: A theory of market equilibrium under conditions of risk," *The Journal of Finance*, vol. 10, pp. 425–442, 1964.

[102] Shewry, "Maximum entropy sampling," *Journal of Applied Statistics*, vol. 14(2), p. 165, 1987.

[103] Shoemaker, A. C., Tsui, K., and Wu, C. F. J., "Economical experimentation methods for robust design," *Technometrics,*, vol. 33(4), pp. 415–427, 1991.

[104] Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K., "Metamodels for computer-based engineering design: Survey and recommendations," *Engineering with Computers*, vol. 17, pp. 129–150, 2001.

[105] Sobester, A., Leary, S. J., and Keane, A. J., "On the design of optimization strategies based on global response surface approximation models," *Journal of Global Optimization*, vol. 33, no. a, pp. 31–59, 2005.

[106] Steuer, R. E., Qi, Y., and Hirschberger, M., "Multiple objectives in portfolio selection," *Journal of FInancial Decision Making*, vol. 1(1), 2005.

[107] Taguchi, G., *Introduction to Quality Engineering: Designing Quality into Products and Processes.* Asian Productivity Organization, 1986.

[108] Tan, K. C. and Goh, C. K., "Handling uncertainties in evolutionary multiobjective optimization," in *Computational Intelligence: Research Frontiers, IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008, Plenary/Invited Lectures* (Zurada, J. M., Yen, G. G., and Wang, J., eds.), vol. 5050 of *Lecture Notes in Computer Science*, Springer, 2008.

[109] Ting, H. M., "Aggregation of attributes for multiattributed utility assessment," tech. rep., MIT Operations Research Center, 1971.

[110] Tipping, M. E., "Bayesian inference: An introduction to principles and practice in machine learning," in *Advanced Lectures on Machine Learning* (Bousquet, O., von Luxburg, U., and Ratsch, G., eds.), pp. 41–62, Springer, 2004.

[111] Ventyx, "Integrated resource planning." Website, 2011. System Optimizer Software.

[112] von Neumann, J. and Morgenstern, O., *Theory of Games and Economic Behavior*. Princeton University Press, 3rd ed., 1944.

[113] Welch, W., Yu, T., Kang, S. M., and Sacks, J., "Computer experiments for quality control by parameter design," tech. rep., University of Illinois at Urbana-Champaign, 1989.

[114] Wiser, R. and Bolinger, M., "Balancing cost and risk: The treatment of renewable energy in western utility resource plans," *The Electricity Journal*, vol. 19(1), p. 48, 2006.

[115] Wood, E., "Applying multiattribute utility theory to evaluation of tisza river basin development plans," in *IIASA Conference*, vol. 2, pp. 9–18, The International Institute for Applied Systems Analysis, 1976.

[116] Xcel Energy, "Application for resource plan approval." Report, 2010.

[117] Zitzler, E., Laumanns, M., and Thiele, L., "Spea2: Improving the strength pareto evolutionary algorithm." 2001.