PROPAGATION OF IMPRECISE PROBABILITIES THROUGH BLACK-BOX MODELS

A Thesis
Presented to
The Academic Faculty

by

Morgan Bruns

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2006

PROPAGATION OF IMPRECISE PROBABILITIES THROUGH BLACK-BOX MODELS

Approved by:

Dr. Christiaan J.J. Paredis, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Bert Bras
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Leon F. McGinnis
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: April 7, 2006

ACKNOWLEDGEMENTS

This thesis would not have been possible without the assistance of many of my friends and colleagues. I would like to thank my advisor, Dr. Chris Paredis, for his support, guidance, and insight. I would also like to thank the students of the Systems Realization Laboratory for the intellectually nurturing environment they have provided. In particular, I would like to thank Jason Aughenbaugh, Scott Duncan, Jay Ling, Rich Malak, and Steve Rekuc for their criticism and advice regarding my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DBC	Dependency Bounds Convolution
DLS	Double Loop Sampling
DM	Decision Maker
OPS	Optimized Parameter Sampling
PBA	Probability Bounds Analysis
PCS	P-box Convolution Sampling

LIST OF SYMBOLS

$\delta \underline{E}_M[Z]$	Absolute error in lower bound expected value of Z using method M
$\boldsymbol{\theta}$	Vector of distribution parameters
$\boldsymbol{\Theta}$	Super-vector containing all parameters from all uncertain input quantities
μ	Distribution mean
π	Decision policy
σ	Distribution standard deviation
$\Phi_{\mu,\sigma}(x)$	Cumulative normal distribution with mean μ and standard deviation σ
$\mathcal{B}$	Belief function
$C(u,v)$	Copula function
d_i	Design variable indexed by i
$\mathbf{D}_h$	Set of design alternatives at step h
$\mathbf{D}^*$	Final design alternative
$\mathbf{D}^\Gamma$	Γ-maximin design alternative
E_a	Expected value of black box output given input parameter vector $\boldsymbol{\Theta}^a$
$E[U]$	Expected utility
FE_M	Number of function evaluations used by method M
F_X	Cumulative distribution function on X
$\underline{F}_X^{(-1)}(p)$	Inverse cumulative distribution function
$F_X^{\square}(x)$	P-box function on X
$g(\boldsymbol{\Theta})$	Function mapping parameters to expected value of output

$\mathbb{IR}^n$	n-dimensional space of real interval vectors
p	Probability
P	Precise probability function
$\mathcal{P}$	Preference function
q_j	Number of parameters associated with uncertain quantity j
s	Number of inner loop samples used in DLS and OPS
t	Number of outer loop samples used in DLS
u	Utility
U	Precise utility function
x_j	Realized value of an uncertain quantity
$\mathbf{x}$	Single realizable state of affairs
X_j	Uncertain quantity
$\boxed{X}$	General p-box
$\boxed{X}^P$	Parameterized p-box
$\mathbf{X}$	Set of all relevant states of affairs
Z_j	Set of uncertain variables specific to design alternative $\mathbf{D}^j$
Z_s	Set of shared uncertain variables

SUMMARY

From the decision-based design perspective, decision making is the critical element of the design process. All practical decision making occurs under some degree of uncertainty. Subjective expected utility theory is a well-established method for decision making under uncertainty; however, it assumes that the DM can express his or her beliefs as precise probability distributions. For many reasons, both practical and theoretical, it can be beneficial to relax this assumption of precision. One possible means for avoiding this assumption is the use of imprecise probabilities. Imprecise probabilities are more expressive of uncertainty than precise probabilities, but they are also more computationally cumbersome. Probability Bounds Analysis (PBA) is a compromise between the expressivity of imprecise probabilities and the computational ease of modeling beliefs with precise probabilities. In order for PBA to be implemented in engineering design, it is necessary to develop appropriate computational methods for propagating probability boxes (p-boxes) through black box engineering models. This thesis examines the range of applicability of current methods for p-box propagation and proposes three alternative methods. These methods are applied towards the solution of three successively complex numerical examples.

CHAPTER 1: INTRODUCTION

Engineering design is a challenging problem due largely to uncertainty in making design decisions. When making any design decision, the engineer must predict the uncertain consequences of each alternative under consideration so that the most preferred alternative can be identified and chosen. From a mathematical perspective, this requires the use of a formalism in which uncertainty can be expressed, in which one can compute with uncertain quantities to infer information about decision consequences, and based upon which the choice of a particular decision alternative can be justified rationally.

Many alternative formalisms for representing uncertainty have been proposed. Nearly all of these formalisms have mathematical rules for propagating their associated uncertain quantities. Most uncertainty formalisms also have rationally justified decision policies. In this thesis, we take one uncertainty formalism—namely, imprecise probabilities—and examine ways to make the propagation of these uncertain quantities computationally feasible for engineering design.

In this chapter, the problem of decision making under uncertainty in design is introduced. The specific computational aspect of this problem is identified, and formal research questions and hypotheses are then presented.

1.1 Design decision making

Design is the process of converting information about system requirements into a specification of a system that satisfies those requirements. This set of system specifications constitutes a design solution. The space of possible design solutions is unstructured and effectively infinite both in dimension and size. In order to navigate successfully through the structurally complex design space, it is necessary to proceed systematically.

Decision-based design is a useful paradigm for thinking systematically about the design process [1-3]. Designers proceed through the design process with the help of basically two mechanisms: the generation of design alternatives and decision making. From the decision-based design perspective, the critical elements of the design process are the decisions to select from among these alternatives. Note that decision-based design is not an *approach* to design—it is a *perspective*. That is, from the decision-based design perspective, decisions should be the focus of the designer. Within this perspective, there still exist many different approaches to the design process.

Every decision in the design process must be made under some degree of uncertainty. Uncertainty exists when the decision maker (DM) does not know the outcome of at least one decision alternative definitely. The dilemma that uncertainty poses for decision making is clear: different decision alternatives might be preferable in different *possible* (but uncertain) states of the world.

1.2 Imprecision in design

Since uncertainty strongly influences decision making, and therefore design, it is necessary to study the nature of uncertainty. Uncertainty is often divided into two components that we call *variability* and *imprecision*. Variability corresponds to naturally random behavior of a physical system or process. The standard representation of variability is the probability distribution function.

Many of the uncertainties in engineering design are imprecise. Imprecision is uncertainty due to a lack of knowledge or information [4]. Imprecision is alternatively referred to as incertitude, but to maintain consistency with past research in the engineering design community we use the term "imprecision" in this paper. The standard representation of pure imprecision is the interval [5, 6]. Imprecision arises in design from sequential decision-making, statistical data from finite samples, bounded rationality, and many other sources. For a detailed discussion of the sources of imprecision in engineering design, see [7].

Traditionally, the formalism for expressing uncertainty has been probability theory. Dating back to the first half of the twentieth century, probability theory has been shown to support the expression of a DM's beliefs, operationalized as the DM's willingness to bet [8, 9]. Combined with utility theory to express the DM's preferences, a normative decision theory has been established in which the most preferred alternative is determined by maximizing the expected utility [10]. However, in practice, this normative decision theory poses some problems; in order to apply it, one must assume that the DM can express his or her beliefs and preferences accurately and coherently in precise mathematical functions. Even if this were possible, it would require significant resources. Therefore, much of the recent research in decision theory has focused on relaxing the assumptions of precise expressions of beliefs and preferences [11-14].

Although some authors question the philosophical validity of the distinction between variability and imprecision, it has been compellingly argued that such a distinction is useful in practice [15-18]. Even in traditional decision analysis imprecision is accounted for through sensitivity analysis. Although it is assumed in decision analysis that all uncertainty can be represented as precise probability distributions, a sensitivity analysis is considered essential to determine whether the selected decision alternative is sensitive to variations in the problem parameters [19, 20], i.e., to imprecision in the parameters. In this thesis, we focus on Probability Bounds Analysis (PBA) [21] as a way to treat such imprecision in a more systematic fashion. In PBA, all uncertainty is represented as probability boxes, or p-boxes [21-23]. PBA will be discussed in greater detail in Chapters 2 and 3.

PBA is a restricted form of the more general *imprecise probability theory* introduced by Walley [12]. PBA strikes a balance between expressiveness and ease of computing. It is more expressive than traditional probabilities because it allows for ranges or intervals of probability, but it is still relatively easy to process computationally. Aughenbaugh and Paredis have recently shown that under certain circumstances, PBA will lead to better design decisions than obtained through traditional decision analysis [18]. However, to assess more accurately under which circumstances the benefit of additional expressivity outweighs the additional cost of computation, a more careful study of the computational complexity of algorithms for propagating imprecise uncertainty is needed.

1.3 Challenges for decision making formalisms

If more expressive representations of uncertainty are to be introduced into engineering design practice, several aspects of decision making under uncertainty must be addressed. In particular, any method for making decisions under uncertainty must provide three essential tools: (1) a formal representation for uncertain quantities; (2) a method for computing with uncertain quantities; and (3) a decision policy that determines an action under uncertainty. Because of the widespread presence of imprecise uncertainty in engineering design, we seek to develop these three tools for the special case of PBA.

This thesis addresses item (2). For insight into the development of representations of imprecise probabilities see [15, 21, 24]. Decision making with imprecise probabilities has been addressed in [12, 25] and with specific emphasis on engineering design in [18, 26].

1.4 Research questions, hypotheses, and thesis outline

This thesis addresses two research questions relevant to the problem of propagating the uncertain quantities of PBA. The first research question concerns the applicability of current computational methods for PBA to problems in engineering design. If PBA is to be used in realistic design problems, we need to determine if the available computational methods can cost effectively propagate uncertain quantities through the types of mathematical and computer models commonly used by engineers. Specifically, the first research question addressed in this thesis is:

> **Research Question 1:** *To what classes of engineering design problems can current computational methods in PBA be applied?*

A specific statement of the problem posed by typical engineering design problems is presented in Chapter 2. The state-or-the-art in computational methods for PBA will be discussed in detail in Chapter 3 at the end of which, the limitations of these methods will be discussed in detail. For the reasons discussed at the end of Chapter 3, the hypothesis to Research Question 1 is:

> **Hypothesis 1:** *Current computational methods for PBA are only applicable to engineering design problems in which*

i) the entire mathematical model used for decision making is analytically defined as a sequence of basic arithmetical operations and elementary functions and

ii) no uncertain variables are repeated—i.e., all variables appear only once in the model expressions.

This hypothesis specifies a very limited class of engineering design problems. If this hypothesis proves to be correct, then alternative computational methods will be needed if PBA is to be applicable towards many of the engineering design problems in which uncertainty is a critical factor.

The second research question addressed in this thesis concerns the possibility of alternative computational methods for PBA. If the current computational methods are unsatisfactory for engineering design, it needs to be determined if other methods exist that make the PBA formalism compatible with the demands of designers. Additionally, the range of applicability of these alternative methods needs to be examined. The specific research question is:

Research Question 2: *Do there exist alternative computational methods for PBA that are compatible with the demands of engineering design?*

Research question two essentially asks what can be done to propagate the uncertain quantities of PBA through realistic engineering design models. In this thesis, three alternative methods for propagating uncertain quantities are introduced. These three methods constitute the hypothesis to research question two:

Hypothesis 2: *The PBA formalism can be made compatible with a broader class of engineering design problems by* *i) a double loop sampling algorithm using parameterized uncertain quantities,*

ii) a modified double loop sampling algorithm called optimized parameter sampling, or

iii) a generalized version of probabilistic sampling called p-box convolution sampling.

These three alternative computational methods are presented in Chapter 4.

The two hypotheses to the two research questions will be evaluated throughout the remainder of this thesis. Before either hypothesis can be studied in detail, though, it is first necessary to clarify the problem posed by engineering design decision making. The problem of design decision making has been introduced in vague terms at the beginning of this chapter, but a more formal description of a design decision problem is presented in Chapter 2. This is necessary in order to set the context and clarify the challenges faced by decision making under uncertainty. The context for the problems posed by the two research questions involves both a representation of uncertain quantities and methods for making decisions in the presence of these types of uncertain quantities. After the context has been set, it is then possible to introduce the current, or state-of-the-art, computational methods for PBA in Chapter 3. Specifically, the dependency bounds convolution method of Williamson and Downs [27] is given as the paradigmatic method for uncertainty propagation in PBA. The weaknesses of the method are then discussed with specific reference to dilemmas associated with engineering design. In Chapter 4, the three alternative methods mentioned in Hypothesis 2 are presented. Although several *a priori* benefits and limitations of these methods can be identified, a more thorough evaluation requires numerical experiments. Therefore, the three methods are applied towards the solution of three increasingly complex example problems in Chapter 5. Finally, the thesis is concluded with a summary of what has been done in light of the research questions. Several avenues for future research are also pointed out.

CHAPTER 2: DECISION MAKING UNDER IMPRECISE UNCERTAINTY

In order to understand the computational challenges of using imprecise uncertainties, it is necessary to understand the computations present in the design process. The design process progresses by a sequence of decisions in which the space of possible design alternatives under consideration is sequentially reduced. We denote a set of design alternatives at step h by $\mathbf{D}_h$. In the early stages of design, $\mathbf{D}_h$ is complex and poorly defined. Much of design research focuses on developing heuristics for refining $\mathbf{D}_h$ to a mathematically manageable size and structure. In this paper, we are not concerned with such methods. Instead we assume that $\mathbf{D}_h$ is an interval vector (or hypercube) of dimension m, $\mathbf{D}_h \in \mathbb{IR}^m$, where $\mathbb{IR}^m$ is the m-dimensional space of real interval vectors: $\mathbb{IR} \equiv \left\{ \left[\underline{x}, \overline{x} \right] : \underline{x}, \overline{x} \in \mathbb{R}, \underline{x} \le \overline{x} \right\}$. More specifically, we assume that we can write

$$\mathbf{D}_h = \left[\left[\underline{d}_1, \overline{d}_1 \right], \left[\underline{d}_2, \overline{d}_2 \right], ..., \left[\underline{d}_m, \overline{d}_m \right] \right]$$

where each $\underline{d}_i$ and $\overline{d}_i$ represent the lower and upper bounds of some real, continuous design variable d_i. Similarly, for discrete design variables, $\underline{d}_i$ and $\overline{d}_i$ correspond to the smallest and largest of the finite set of alternatives. In this context, the set reduction in each design step, $\mathbf{D}_h \to \mathbf{D}_{h+1}$, corresponds to a decrease in interval width for at least one of the n design variables. This process of sequential width reduction converges to a final decision which specifies a precisely defined (singleton) design alternative, $\mathbf{D}^* = \left[d_1^*, d_2^*, ..., d_m^* \right]$. Analogously, a *set* of discrete design alternatives could be reduced sequentially through a series of decisions converging to $\mathbf{D}^*$.

Since design computations often involve only a sequence of decisions that are assumed to be decoupled, we focus on the computations involved in a single decision. In the following sections, we will examine in greater detail the mechanics of a single design decision. This involves representing the DM's beliefs and preferences and using performance models to predict how a particular design will satisfy the DM's preferences. This chapter will close with a precise statement and discussion of the computational problem we hope to solve.

2.1 Elements of a design decision problem

A rational decision should reflect the DM's beliefs and preferences. Given a set of beliefs, preferences, and a set of design alternatives, the DM uses some decision policy to determine the preferred decision alternative. It is important here to differentiate a *decision alternative* and a *design alternative*. A decision alternative is any choice that the DM has available at any step in the sequential design process. A design alternative, on the other hand, is any completely specified design. A single decision alternative might correspond to multiple design alternatives. See [7] for a more detailed discussion.

A decision policy can be represented by the expression

$$\mathbf{D}_{h+1} = \pi(\mathcal{B}, \mathcal{P}, \mathbf{D}_h)$$

which can be translated into the decision to eliminate the set $\mathbf{D}_e = \mathbf{D}_h \setminus \mathbf{D}_{h+1}$. Here $\mathcal{B}$ is a functional representation of the DM's belief state, $\mathcal{P}$ is a functional representation of the DM's preference state, and π is the decision policy. The set of decision alternatives is the set of all the proper subsets of $\mathbf{D}_h$ excluding the null set. In effect, the decision alternatives are all the ways in which the DM can impose a constraint on $\mathbf{D}_h$. The different constraints imposed are necessarily defined in terms of the design variables that compose $\mathbf{D}_h$.

The belief state, $\mathcal{B}$, is some general multi-valued function that embodies the DM's beliefs about the state of the relevant world at the time of the decision. It is a general but quantifiable measure of the DM's uncertainty about the set of relevant states of affairs. In general, $\mathcal{B}$ is a multi-valued function because of the possibility of imprecision. Realizable relevant states (assumed to be quantifiable) will be represented as scalar vectors, $\mathbf{x} \in \mathbb{R}^n$, where each element, x_j, of $\mathbf{x}$ corresponds to some relevant uncertain quantity. The set of all relevant states of affairs will be denoted by $\mathbf{X} = \left[X_1, X_2, ..., X_n\right]$. In the context of design, $\mathbf{X}$ can be thought of as the set of variables over which the DM has no control. Mirroring the notation for random variables, uppercase is used to emphasize that the actualized relevant state is an uncertain quantity ranging over the space of possible states of affairs. Note that the DM might choose to model any X_j as certain—that is, $X_j = x_j$ is a known quantity. The most common representation of a belief state is a precise probability measure over the sample space of relevant states of affairs. A precise probability measure is a single-valued function $P : \mathbb{R}^n \to [0,1]$. That is, $P(\mathbf{x}) = p$ such that $p \in \left[0,1\right]$.

The preference state, $\mathcal{P}$, is some general multi-valued function that embodies the DM's preferences about possible consequences of the decision. Like $\mathcal{B}$, $\mathcal{P}$ can be multi-valued in order to account for imprecision. The uncertain consequences of a decision are dependent on the actual relevant state of affairs $\mathbf{x}^l = \left[x_1^l, x_2^l, ..., x_n^l\right]$ (corresponding to state l) as well as the design, $\mathbf{D}^k = \left[d_1^k, d_2^k, ..., d_m^k\right]$ (corresponding to a specific design alternative k), chosen. The preference state, $\mathcal{P}$, at the time of the decision is a measure of the value of a particular consequence to the designer. The most common representation of the preference state is a single-valued utility function $U : \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}$. That is, $U\left(\mathbf{D}^k, \mathbf{x}^l\right) = u$. The utility of a of a particular design, $\mathbf{D}^k$, given some specific outcome, $\mathbf{x}^l$, is deterministic, but since $\mathbf{x}^l$ is uncertain, the utility of $\mathbf{D}^k$ is also uncertain.

Unlike the uncertain state vector, **X**, the design alternative search space, $\mathbf{D}$, is controlled by the DM. Generally, each d_i in **D** might be continuous or discrete and bounded or unbounded. For simplicity, we make the assumption that **D** is an interval vector in $\mathbb{IR}^m$ as was discussed at the beginning of this section.

Finally, the decision policy, π, is a general multi-valued functional mapping from the DM's beliefs and preferences to the set of *non-dominated* decision alternatives $\mathbf{D}_{h+1}$. A non-dominated decision alternative is an alternative that, given some body of information, the DM cannot rationally eliminate. In classical decision theory, π is "maximize expected utility." Mathematically, the preferred solution is found as

$$\mathbf{D}^* = \arg\max_{\mathbf{D}^k \in \mathbf{D}_h} \left[\sum_{x_1} ... \sum_{x_n} u(\mathbf{D}^k, \mathbf{x}) \, p(\mathbf{x}) \right]$$

or

$$\mathbf{D}^* = \arg\max_{\mathbf{D}^k \in \mathbf{D}_h} \left[\int_{x_1} ... \int_{x_n} u(\mathbf{D}^k, \mathbf{x}) \, p(\mathbf{x}) dx_n ... dx_1 \right]$$

for discrete and continuous problems, respectively. In this case, $\mathbf{D}_e = \mathbf{D}_h \setminus \mathbf{D}^*$.

Two special cases of the general decision problem should be mentioned. Both of these specific cases make assumptions about the uncertainty of the DM's beliefs and preferences. The decision is deterministic when all beliefs and preferences are certain. In this case, the DM can simply maximize the utility over $\mathbf{D}_h$. The preferred design solution will be $\mathbf{D}^* = \arg\max_{\mathbf{D}^k \in \mathbf{D}_h}(U(\mathbf{D}^k))$. The DM selects the design that necessarily results in the best system performance. This case is unrealistic since design decisions always involve uncertainty with regards to beliefs and preferences.

The second special case of a general design decision acknowledges the presence of uncertainty, but represents that uncertainty as precise probability distributions. That is the DM's beliefs are purely probabilistic, and his or her preferences are deterministic. Sampling strategies such as Monte Carlo and Latin Hypercube are well-established and frequently-used solutions for propagating precise probabilistic uncertainty [28]. The DM is able to make a decision by maximizing the expected utility of the design through stochastic programming. The resulting design solution will be $\mathbf{D}^* = \arg\max_{\mathbf{D}^k \in \mathbf{D}_h}(E_{\mathbf{X}}[U(\mathbf{D}^k)])$ where the subscript on E denotes that the expectation is taken over the vector of random variables $\mathbf{X}$. This case is more realistic than the purely deterministic solution described above but is still an approximation because the DM is not able to account for imprecise uncertainty.

Before we can study computational methods for handling imprecise uncertainty, we must first make some simplifying assumptions about the representation of uncertain quantities and the decision and performance models to be used.

2.2 The p-box representation of uncertain quantities

An *uncertain quantity* is a generalization of a random variable. It is an event or variable characterized by a *set* of degrees of belief. Whereas a random variable characterizes a quantity by some precise belief function—namely, a probability distribution function—an uncertain quantity is characterized by a set of belief functions. For instance, consider a bent quarter. I am uncertain about whether it will land heads-up or tails-up on a given toss, but until I have seen it flipped many times, I am also uncertain about how probable it is that it will land heads-up or tails-up. I believe that the probability of the bent quarter landing heads-up is less than 0.6 and greater than 0.3. My belief state then corresponds to the interval of probability values between 0.3 and 0.6. That is, $P(H) = \left[\underline{P}(H), \overline{P}(H) \right] = \left[0.3, 0.6 \right]$.

The probability box, or p-box, is a formalism for representing uncertain quantities [21, 22]. P-boxes are less general than imprecise probabilities, but the loss of generality is compensated by increased computational convenience. The defining characteristics of a p-box are the probability bounds that define upper and lower limits on the cumulative probability over the domain of the uncertain quantity. When defining a p-box formally, there are essentially two structures involved: the p-box proper, and the p-box function. The p-box proper, $\boxed{X}$, of some uncertain quantity *X* defines the p-box as a set of non-decreasing distribution functions constrained by probability bounds:

$$\boxed{X} = \left\{ F_X(x) : \forall x \in \mathbb{R},\ \underline{F}_X(x) \le F_X(x) \le \overline{F}_X(x) \right\}$$

where $\underline{F}_X, F_X, \overline{F}_X : \mathbb{R} \to [0,1]$, $\underline{F}_X = \underline{P}(X \le x)$ and $\overline{F}_X = \overline{P}(X \le x)$ are the lower and upper cumulative probability bounds, and F_X is non-decreasing with *x*. An example p-box is shown in Figure 1. These probability bound functions are determined by the p-box function. The p-box function is an interval-valued mapping from *x* to the interval $[0,1]$. We express the p-box function as

$$F_X^{\square}(x) = \left[\underline{F}_X(x), \overline{F}_X(x) \right]$$

where $\underline{F}_X(x) \le \overline{F}_X(x)$ for all x. In some discussions, it might be useful to reverse the order of the bounding distributions in the interval above such that $F_X^{\square}(x) = \left[\overline{F}_X(x), \underline{F}_X(x) \right]$. In this case, $\overline{F}_X(x)$ denotes the left bound on the p-box and $\underline{F}_X(x)$ denotes the right bound. In other words, upper and lower are defined with respect to *x* rather than with respect to cumulative probability. The essential problem is whether we want to model lower and upper in terms of probability or in terms of the value of the uncertain quantity. For our purposes it is more convenient to interpret *upper* and *lower* with respect to probability.

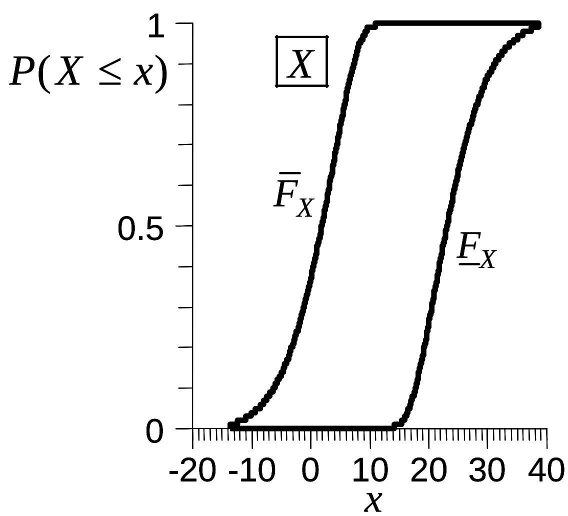

Figure 1. Example p-box $\boxed{X}$ with lower and upper bounding functions.

The p-box is general enough to represent intervals, probability distributions, scalars, as well as imprecise probability distributions. An interval $X = [a, b]$ corresponds to the p-box defined by the probability bounds

$$\underline{F}_X(x) = \begin{cases} 0, & x < b \\ 1, & x \geq b \end{cases}$$

and

$$\overline{F}_X(x) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases}.$$

As an example of a precise probability distribution, a normally distributed random variable, $X \sim N(\mu, \sigma)$, corresponds to the p-box containing only one cdf, $\boxed{X} = \left\{ \Phi_{\mu,\sigma}(x) \right\}$, and the degenerate p-box function with $\underline{F}_X(x) = \overline{F}_X(x) = \Phi_{\mu,\sigma}(x)$ where $\Phi_{\mu,\sigma}(x)$ is the cumulative distribution function of the normal distribution with mean μ and standard deviation σ. A scalar, *a*, corresponds to the degenerate p-box function with

$$\underline{F}_X(x) = \overline{F}_X(x) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases}.$$

Finally, and most importantly, the p-box can be used to represent imprecise probability distributions such as $X \sim N\left(\left[\underline{\mu}, \overline{\mu} \right], \left[\underline{\sigma}, \overline{\sigma} \right] \right)$. Here it is known that the uncertain quantity has normal variability with an imprecise mean, $\mu \in \left[\underline{\mu}, \overline{\mu} \right]$, and an imprecise standard deviation, $\sigma \in \left[\underline{\sigma}, \overline{\sigma} \right]$. This imprecise probability distribution corresponds to the *parameterized* p-box

$$\boxed{X}^P = \left\{ F_X(x; \mu, \sigma) = \Phi_{\mu,\sigma}(x) : \mu \in \left[\underline{\mu}, \overline{\mu} \right], \sigma \in \left[\underline{\sigma}, \overline{\sigma} \right] \right\}$$

where the superscript P denotes that the p-box is parameterized. It is not meaningful to speak of bounding functions for parameterized p-boxes since the parameterized p-box will not contain all non-decreasing functions between its lower and upper bounding functions.

Parameterized p-boxes are less general than the p-box as defined above. A parameterized p-box is the set of all possible distributions resulting from some known distribution function with imprecisely known parameters. Formally,

$$\boxed{X}^P = \left\{ F_X(x;\boldsymbol{\theta}) : \boldsymbol{\theta} \in \left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}} \right] \right\}$$

where $F_X(x;\boldsymbol{\theta})$ is non-decreasing with x, and $\boldsymbol{\theta} \in \mathbb{R}^q$ is a vector of distribution parameters that affect the shape or scale of F_X. Imprecision is introduced through uncertainty in the parameters. Specifically, the DM is uncertain of the true values of the distribution parameters except for the fact that they lie within known bounds. That is, for all $\theta_k \in \boldsymbol{\theta}$, $\underline{\theta}_k \le \theta_k \le \overline{\theta}_k$.

It is important to emphasize the difference between a parameterized p-box and a general p-box. Similar to a general p-box, a parameterized p-box is a set of non-decreasing probability distribution functions constrained by upper and lower bounds. But unlike a general p-box, a parameterized p-box does not contain all possible non-decreasing distributions lying between its lower and upper bounds. In set notation, if $\boxed{X}$ and $\boxed{X}^P$ share the same bounding functions, then $\boxed{X}^P \subset \boxed{X}$. To see this, consider a p-box and a parameterized p-box with the same upper and lower bounds.

$$\boxed{X} = \left\{ F_X(x) : \underline{F}_X(x) \le F_X(x) \le \overline{F}_X(x) \right\}$$

where $\underline{F}_X$ is normally distributed with mean $\mu = 4$ and standard deviation $\sigma = 1$ and $\overline{F}_X$ is normally distributed with $\mu = 1$ and $\sigma = 1$. A parameterized p-box with identical bounds is

$$\boxed{X}^P = \left\{ F_X(x;\mu,\sigma) : X \sim Normal\left(\mu = [1,4], \sigma = 1\right) \right\}.$$

Both of these sets of functions are constrained by the bounds $\underline{F}_X$ and $\overline{F}_X$, but $\boxed{X}$ contains functions not found in $\boxed{X}^P$ as shown in Figure 2.

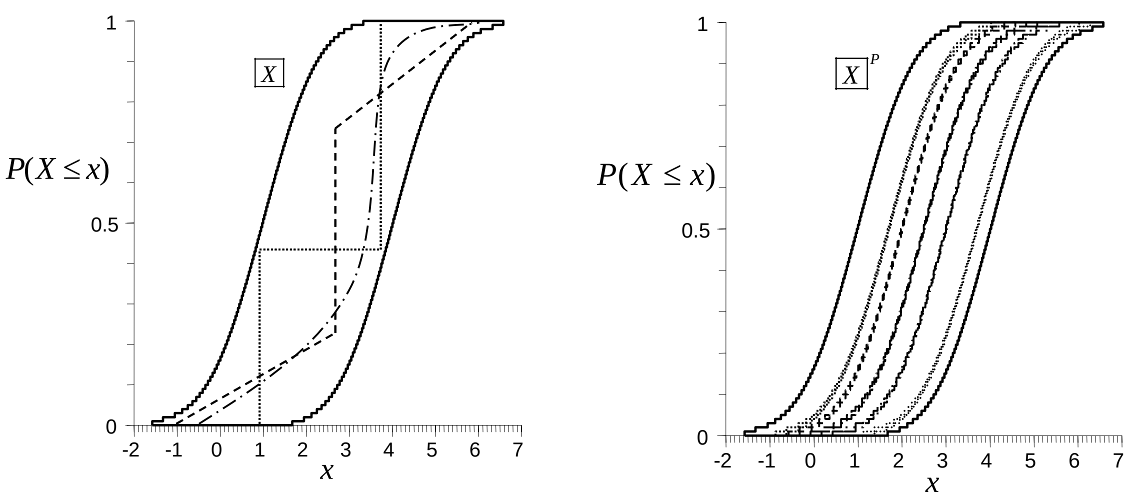

Figure 2. Comparison of a general (on left) and a parameterized p-box (on right).

Though less general, a parameterized p-box is in many cases a better representation of the DM's beliefs about an uncertain quantity. A common example of this arises in statistical parameter estimation where data gives rise to confidence intervals on the true parameter values for some random variable with a known distribution.

P-boxes are intuitive representations for uncertainty. They are direct generalizations of both intervals and probability distributions, and are a sub-set of the more general class of imprecise probabilities as introduced by Walley [12]. Unlike some other representations that have been used for imprecise uncertainties such as possibilities [29] or fuzzy sets [30], probability boxes have a clear operational definition. An operational definition is "a rule which indicates how the mathematical notions are intended to be interpreted [31]." The subjective interpretation of probability provides an operational definition in terms of subjective degree of belief expressed through a willingness to bet [8, 9]. Walley extends the subjective interpretation to include ranges of probabilities (lower and upper previsions to be more exact) by differentiating between the minimum selling prices and maximum buying prices of gambles [12]. For a criticism of uncertainty models without clear operational definitions, see [31].

Although not quite as expressive as imprecise probabilities, p-boxes have the advantage that relatively efficient algorithms have been developed for their propagation. As examples of p-box propagation algorithms in the literature, see the work of Williamson and Downs [27], Ferson [15, 21, 23, 32], and Berleant [33-36]. We will describe and criticize these algorithms in Chapter 3.

2.3 Utility models represented by black box functions

So far, we have only studied decision policy models in terms of abstract functional mappings from beliefs and preferences to a preferred action. To complete the link from generic decision theory to specific design practice, we must first present and justify a key assumption regarding the mathematical models to be used in design decision making.

For practical reasons, proposed methods for propagating p-boxes should assume that all mathematical models are black boxes. Although it is not true that engineering models are truly black boxes, in the sense that nobody knows the mathematical operations inside, it is true that much of engineering practice uses previously developed models as if they were black boxes. In Chapter 3, some methods for propagating p-boxes through open box models will be discussed. In the future, it is possible that these methods will be implemented in much of the standard engineering software. At this point in time, however, this is not the case. Additionally, much of engineering design practice requires the aid of advanced simulation software for finite element analysis or computational fluid dynamics. If the representation of beliefs as p-boxes is to take hold in the engineering design community it is necessary that methods be developed that propagate p-boxes through advanced software black box models.

2.4 Decision policies for imprecise beliefs and preferences

A rational DM must choose decision alternatives that maximize his or her utility. In the presence of uncertainty, utility is no longer certain. Therefore, in accordance with the axioms of decision theory, the DM should choose the alternative that maximizes his or her expected utility, $E[U]$. If the DM's uncertainty is all due to variability, maximizing expected utility is sufficient. However, in the previous discussion, it has been argued that the DM's beliefs and preferences are imprecise. The presence of imprecision results in intervals of expected utility, $\left[\underline{E}[U],\overline{E}[U]\right]$. While imprecise beliefs and preferences more accurately reflect the DM's knowledge state, they also complicate considerably the act of decision making. The lower and upper bounds on expected utility are the fundamental quantities necessary for decision making under imprecise uncertainty. A DM with an imprecise knowledge state needs a more sophisticated decision policy than classical decision theory's prescription of "maximize expected utility." Specifically, imprecise preferences lead to indeterminacy, and indeterminacy results in sets of *non-dominated* decision alternatives. In other words, imprecise preferences result in situations in which rational decision makers cannot choose a single alternative from the set of non-dominated alternatives. Researchers in the imprecise probability community have proposed several decision policies to overcome the indeterminacy in imprecise decision making [26, 37]. Here we limit our discussion to two of these criteria: maximality [12] and Γ-maximin [11]. Any proposed method for propagating p-boxes through black box decision models should be compatible with these decision criteria for imprecise utilities.

To better understand the indeterminacy associated with imprecise knowledge, consider a simple decision problem in which the DM must select a value for a continuous design variable, *d*. The DM in this situation can quantify his or her preferences for single values of *d* with an imprecise expected utility function, $E\left[U(d)\right]=\left[\underline{E}\left[U(d)\right],\overline{E}\left[U(d)\right]\right]$. The upper and lower bounds of the DM's utility function are shown in Figure 3.

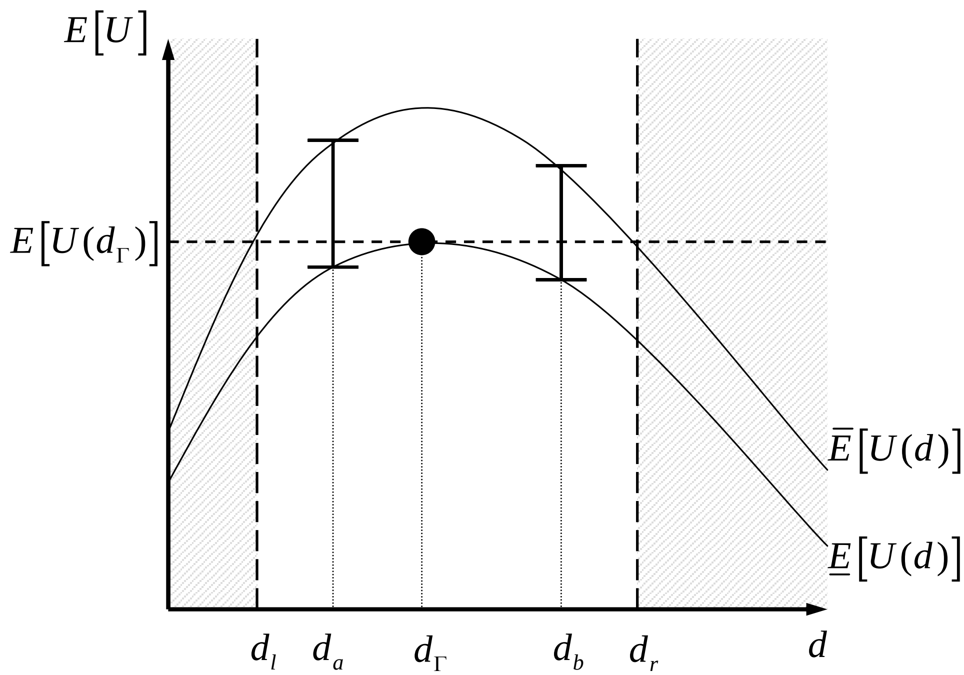

Figure 3. Decision indeterminacy with imprecise utilities.

Which value of *d* should the DM select? The higher the utility the more preferred the design, but in this example the utility bounds overlap. Consider a comparison between design alternatives d_a and d_b as shown in Figure 3. The actual utility of either of these alternatives could fall anywhere between their corresponding upper and lower utility bounds, but the DM has no information about where in those bounds. In some actual cases, d_a will be preferable, but in other cases, d_b will more fully satisfy the DM's preferences. We say that d_a and d_b are *pairwise non-dominated*, and the decision between d_a and d_b is indeterminate. In our example, there is a set of dominated design alternatives. All designs between d_l and d_r are non-dominated by every other design alternative in $[d_l, d_r]$. However, all design alternatives outside of this region are pairwise dominated by the design alternative d_Γ. Therefore, a rational DM will eliminate the set of design alternatives $d < d_l$ and $d > d_r$. The elimination criterion used to find these bounds is called *interval dominance*. Indeterminacy remains for all designs between these two bounds. In engineering design, indeterminacy is ultimately not an option since a final design for production cannot be imprecisely specified. Therefore, the DM needs a more sophisticated decision policy in order to reduce further the space of non-dominated decision alternatives.

Indeed there is no decision policy that is able to identify a *single* rationally preferred solution in the presence of imprecise uncertainty because indeterminacy is inherent in the problem. The DM could rationally choose *any* of the alternatives in the set of non-dominated alternatives, but none of the decision alternatives in that reduced set is rationally preferable to any of the others in that set given the current knowledge state of the DM. Decision policies for imprecise uncertainty can be grouped into two general strategies: (1) those that seek to minimize the size of the set of non-dominated alternatives through more sophisticated comparisons of alternatives, and (2) those that select a single-valued solution based on some semi-arbitrary decision criterion. While strategies of type (1) are preferable for rational decision making, for practical purposes, the DM may need to employ some strategy of type (2) in order to find a single-valued design solution.

The decision policies that seek to minimize the set of non-dominated alternatives differ in the amount of information they take into account. Generally, as more information is considered, the resultant set of non-dominated alternatives will decrease in size. The *maximality* criterion [12] is well-suited for a broad-class of decision problems because it takes into account most of the available relevant information. By introducing differences in expected utility, the DM is able to identify alternatives that are dominated throughout the entire space of possible states of affairs, **X**. Recall by states of affairs, we are referring to possible assumed values of the entire set of uncertain quantities. As an example, consider two design alternatives with overlapping utility bounds. It would appear that neither of these two alternatives is rationally preferred. Further analysis reveals that one of the uncertain quantities affecting the utilities of these designs is shared—that is, this uncertain quantity necessarily assumes the same value regardless of the design alternative chosen. Furthermore, it is found that for all possible values that this shared uncertain quantity can assume that one of the two original design alternatives outperforms the other. A rational DM should select the better performing design alternative. A strict comparison of utility bounds will lose this additional information. The maximality criterion takes into account shared uncertainty. Shared uncertain variables, z_s, are those uncertain quantities that are independent from the design variables—i.e., no matter what design variable is selected the shared uncertain variable will assume the same unknown value. Therefore, when comparing two designs with shared uncertainty, comparisons should only be made between utilities calculated using the same assumed value of the shared uncertain quantities. The maximality criterion prescribes that the DM eliminate all decision alternatives for which, when compared to some other alternative evaluated at the same values for the shared uncertain variables, the upper bound on their expected difference in utility is strictly less than zero. Formally,

$$\mathbf{D}_e = \left\{ \mathbf{D}^j \in \mathbf{D}_h : \left(\exists \mathbf{D}^k \in \mathbf{D}_h\right) \left(\max_{\substack{z_s \in Z_s \\ z_j \in Z_j \\ z_k \in Z_k}} \overline{E}\left[U\left(\mathbf{D}^j, z_j, z_s\right) - U\left(\mathbf{D}^k, z_k, z_s\right) \right] < 0 \right) \right\}$$

where $\mathbf{D}^j$ and $\mathbf{D}^k$ are specific decision alternatives in the set $\mathbf{D}_i$, Z_s is the set of shared uncertain variables, Z_j is the set of uncertain variables specific to $\mathbf{D}^j$, and Z_k is the set of uncertain variables specific to $\mathbf{D}^k$. As an illustration of the use of the maximality criterion, consider again the example in which the DM is trying to select a single value for *d*. Based on past experience, or some other heuristic, the DM believes that d^* will most likely be the preferred solution. In order to eliminate a larger set of design alternatives, the maximality criterion requires that the DM calculate $\bar{E}\left[U(d_i, Z_s, Z_i) - U(d^*, Z_s, Z^*)\right]$ for all $d_i \neq d^*$. A plot of this expected difference in utility is shown in Figure 4.

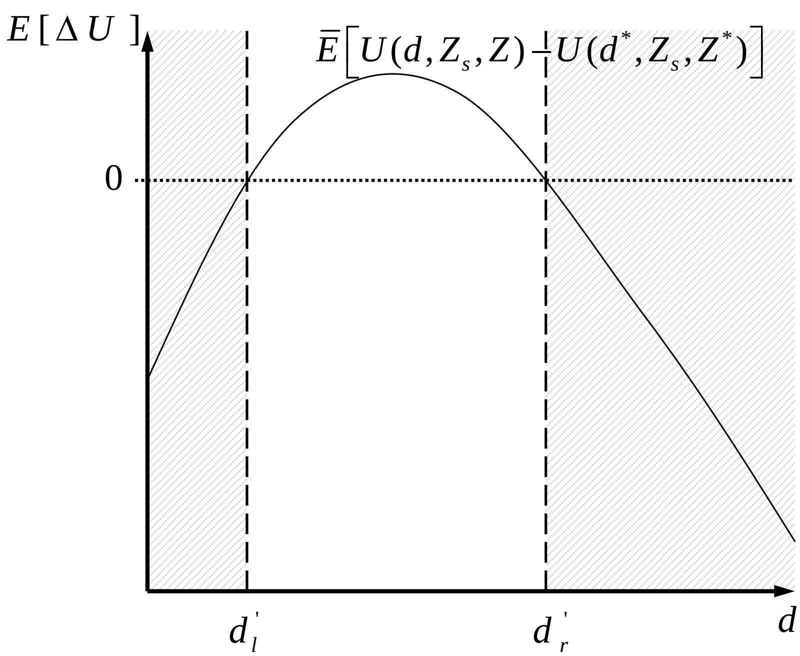

Figure 4. Elimination with the maximality criterion.

For all values of d less than $d_l^{'}$ and greater than $d_r^{'}$, $\bar{E}\left[U(d_i, Z_s, Z_i) - U(d^*, Z_s, Z^*)\right] < 0$. This means that no matter what the actual relevant state of affairs, d^* will outperform those designs, and these regions can be eliminated from consideration. In terms of previous notation, $\mathbf{D}_e = \left\{d : d < d_l^{'} \text{ and } d > d_r^{'}\right\}$. While application of the maximality criterion will identify a smaller set of non-dominated alternatives, the DM will still remain indeterminate between the reduced set of alternatives—in this example the DM is indeterminate between all $d \in \left[d_l^{'}, d_r^{'}\right]$. In general, the bounds found through application of the maximality criterion will be tighter than the bounds arising from the application of the interval dominance criterion—that is, $\left[d_l^{'}, d_r^{'}\right] \subseteq \left[d_l, d_r\right]$ and most often $\left[d_l^{'}, d_r^{'}\right] \subset \left[d_l, d_r\right]$ where d_l and d_r are the bounds determined by interval dominance.

The use of shared uncertain variables is similar to the variance reduction technique of using common random numbers (CRNs) in simulation [38]. The goal of a simulation is usually to compare two scenarios or alternative designs by examining the difference in output for different combinations of control parameters. If different random numbers are used in the simulations for the different alternatives, additional noise is introduced into the model. CRNs are used to induce correlation between scenarios, thereby reducing the variances of the results. In engineering design, shared uncertainty is an inherent characteristic of the problem. Therefore, a DM does not have to add the commonality, he or she merely needs to recognize it and take advantage of that additional property when it exists. The maximality criterion is a means of exploiting this inherent commonality. A detailed discussion of shared uncertainty can be found in the Master's Thesis of Rekuc [39] as well as in [26].

In order to identify a single-valued decision, the DM must employ some semi-arbitrary decision policy. By semi-arbitrary we mean that the decision policy is chosen reasonably, but that it involves additional assumptions beyond those required by strict rationality as defined in terms of avoiding sure loss. A semi-arbitrary policy is not strictly irrational, but it is one of many possible rational policies needed to arrive at a single-valued decision in the presence of imprecision. The most conservative of these types of policies is the Γ-maximin criterion [11]. The Γ refers to the set of prior distributions considered in Berger's robust Bayesian analysis. Very simply, Γ-maximin prescribes that the DM select the alternative that maximizes the lower bound on expected utility. In other words, the DM selects the best worse case solution. Formally, the Γ-maximin solution is found by the expression

$$\mathbf{D}^{\Gamma} = \arg\max_{\mathbf{D}^k \in \mathbf{D}_h} \left(\underline{E}_{\boxed{\mathbf{X}}} \left[U\left(\mathbf{D}^k, \mathbf{x} \right) \right] \right)$$

where the subscript, $\boxed{\mathbf{X}}$, on $\underline{E}$ denotes that the lower expectation is taken over the entire uncertain state space. In Figure 3, the Γ-maximin solution is marked d_{Γ}. Selecting the Γ-maximin solution assures that in the worst-case actualized state of affairs, d_{Γ} will outperform any other design alternative operating in its worst-case actualized state of affairs. This is semi-arbitrary because the DM has no rational reason to believe that the worst-case will be actualized, but the DM can still be certain that performance will at least exceed $\underline{E}\left[U(d_{\Gamma})\right]$. It is possible that a particular problem will not have a unique Γ-maximin solution. In this situation it might be desirable to compare the possible Γ-maximin solutions in terms of their upper expected utilities. However, the choice remains semi-arbitrary since the decision is between a semi-arbitrarily reduced set of non-dominated alternatives.

In the presence of imprecision, the DM will generally need to resort to using some semi-arbitrary decision policy such as Γ-maximin to make a final decision. What value then are the interval dominance and maximality criteria? Should not the DM just compute and select the Γ-maximin solution? The Γ-maximin solution is a function of the body of information available to the DM. Since the design process is not self-contained, this body of information is not static. As the DM progresses through the design process, new information about the structure of the design space and the likelihood of different relevant states of affairs become known. Therefore, the DM should delay making unnecessary (i.e., specific) decisions in the early stages of the design process. The value of proceeding through the design process with sets of design alternatives is discussed in the set-based design literature [26, 40, 41]. The maximality criterion leads to tight, but rational bounds, on the most-preferred solution and so it is therefore useful in the early stages of the design process. Following this process allows the DM to eliminate clearly dominated designs early and to focus his or her resources on the set of clearly non-dominated design alternatives. The DM is then able to seek out more relevant information in order to compare the remaining design alternatives. If the DM were forced to make a final decision based on his or her current knowledge state, it is likely that he or she will not chose the most-preferred decision alternative. Delaying decision making allows the DM to arrive at a more-preferred decision alternative without directing scarce resources towards developing clearly dominated decision alternatives.

2.5 Problem statement

Now that the general issues involved in computing with imprecise information have been explicated, we can now present a concise statement of the problem.

Given:

1. A black box utility function $U = f(\mathbf{D}^k, \mathbf{x})$ where U is the utility of the design $\mathbf{D}^k \in \mathbb{R}^m$ dependent on some $\mathbf{x} \in \mathbb{R}^n$. Generally, f is an interval-valued mapping $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{IR}$ resulting in the lower and upper utilities $\underline{U}(\mathbf{D}^k, \mathbf{x})$ and $\overline{U}(\mathbf{D}^k, \mathbf{x})$.
2. A vector of p-boxes of dimension n, $\boxed{\mathbf{X}} = \left[\boxed{X}_1, \boxed{X}_2, ..., \boxed{X}_n\right]$, describing the uncertainty about the relevant state of affairs, $\mathbf{x}$. This assumes that no *joint* p-box distribution is known which is typically the case in engineering problems. In other words, nothing is known about the dependence relationships between the uncertain quantities.

Find:

1. The lower and upper expected utilities of a design, $\mathbf{D}^k$, with respect to the vector of uncertain quantities, $\boxed{\mathbf{X}}$: $\underline{E}_{\boxed{\mathbf{X}}}\left[U\left(\mathbf{D}^k, \mathbf{x}\right)\right]$ and $\overline{E}_{\boxed{\mathbf{X}}}\left[U\left(\mathbf{D}^k, \mathbf{x}\right)\right]$.
2. The set of dominated solutions under the maximality criterion:
$$\mathbf{D}_e = \left\{ \mathbf{D}^j \in \mathbf{D}_h : \left(\exists \mathbf{D}^k \in \mathbf{D}_h\right) \left(\max_{\substack{z_s \in Z_s \\ z_j \in Z_j \\ z_k \in Z_k}} \overline{E}\left[U\left(\mathbf{D}^j, z_j, z_s\right) - U\left(\mathbf{D}^k, z_k, z_s\right)\right] < 0 \right) \right\}.$$
3. The Γ-maximin solution: $\mathbf{D}^\Gamma = \arg\max_{\mathbf{D}^k \in \mathbf{D}_h} \left(\underline{E}_{\boxed{\mathbf{X}}}\left[U\left(\mathbf{D}^k, \mathbf{x}\right)\right]\right)$.

Any method capable of solving this problem with minimal computational cost will make PBA feasible for engineering design. Therefore, the effectiveness of any alternative computational methods for PBA should be judged according to its ability to solve the problem stated above. Before we propose alternative methods, though, we must first describe and criticize the available methods for PBA.

CHAPTER 3: AVAILABLE METHODS FOR UNCERTAINTY PROPAGATION

To answer Research Question 1, and to test Hypothesis 1, it is necessary to study available methods for PBA. In this chapter, we describe in greater detail the p-box propagation method of dependency bounds convolution (DBC). Because of the structure of this method, it is argued in Section 3.3 that DBC is only compatible with design problems in which the entire mathematical model used for decision making is analytically defined as a sequence of basic arithmetical operations and elementary functions and no uncertain variables are repeated. Therefore, Hypothesis 2 is confirmed.

3.1 Summary of the literature

Several solutions to the problem of computing with uncertain quantities have been proposed in the literature. Although analytical methods based on Laplace and Mellin transforms exist for a limited class of operations on *precise* random variables [42], no work has been done to extend these methods to accommodate *imprecise* random variables. A completely stochastic alternative involves double-loop sampling [43]. The current state-of-the-art methods numerically compute best-possible bounds on the resultant probability distribution of some function of imprecise random variables[27, 35]. While these methods are efficient and accurate, they are not practical for a large class of engineering design problems. The weaknesses of these methods will be discussed in Section 3.3.

Computing with imprecise probabilities is a generalization of the problem of computing the convolution of probability density functions where the probability density functions happen to be imprecise. In this paper, we use the term convolution to mean any operation on some set of random variables[1]. Extensive summaries of analytical methods for computing convolutions of random variables is found in the book by Springer [42] and in the thesis of Williamson [44].

The most straightforward approach for propagating imprecise probabilities through mathematical models is double loop Monte Carlo sampling – this is alternatively called two-dimensional, 2-D, or second-order Monte Carlo. A good review of second-order Monte Carlo methods is found in [43]. In Section 4.1, we present a version of double loop sampling that is more appropriate for the p-box representation of uncertain quantities. Other modifications to pure double loop sampling methods are presented in [45, 46]. Monte Carlo techniques are easy to implement, but for many complex problems, their computational cost becomes prohibitive.

The first efficient numerical approach to the propagation of uncertain quantities was presented by Williamson and Downs in [27, 44]. Williamson's work was motivated by the desire to develop numerical methods for precise probabilistic arithmetic, but his methods are compatible with imprecise probabilistic arithmetic. Williamson's methods are referred to as *dependency bounds convolutions* (DBC) because they result in bounds on the true probability distribution under any possible dependence relation between the uncertain quantities. Dependency bounds are "best-possible" in the sense that the resultant bounds are guaranteed to contain the true resultant distribution, and any

[1] Note that this definition is not in agreement with the precise definition used by probabilists. We adopt a broader definition in order to maintain consistency with the literature on p-box computations.

reduction of the bounds results in the possible exclusion of the true distribution. The commercially available software Risk Calc 4.0 [32] provides an implementation of the dependency bounds methods.

A very similar approach was developed independently by Berleant in [34, 35]. Both Berleant's approach and Williamson's approach discretize probability distribution functions and use maximization and minimization operations to find the best-possible probability bounds on the resultant quantity. Berleant's approach is implemented in the software Statool [36, 47]. Berleant calls his approach *distribution envelope determination* or DEnv. Regan, Ferson, and Berleant [48] have shown that DEnv and dependency bounds convolution are equivalent for cumulative distribution functions on the positive reals. A more detailed description of these methods is given in the next section.

These two approaches are fully sufficient for the propagation of uncertain quantities through functional relationships given explicitly as a sequence of binary operations, but they are insufficient for the computations in most realistic engineering design problems. The weaknesses of DBC will be expanded on in section 3.3.

3.2 Dependency Bounds Convolution (DBC)

Dependency bounds convolution (or DBC) [21, 49-51] is a term used to describe a class of rigorous methods for propagating p-boxes through mathematical models. The results of DBC are *rigorous* in the sense that the resultant probability bounds are guaranteed to contain the true probability distribution of the uncertain quantity for any possible dependence relationship between the inputs—assuming that the input p-boxes were themselves rigorous. These probability bounds can also be described as *best-*

possible in the sense that they are as close together as possible given the information provided in the input p-boxes. Finally, DBC calculations are applicable towards *non-parameterized* p-boxes. This means that no assumptions are made about the true probability distribution other than that it is contained within the p-boxes bounding functions.

Although it is unnecessary to fully describe the methods for DBC, it is helpful to sketch in outline how these methods function. The DBC calculation begins with a bounding discretization of the input p-boxes. This is done by partitioning the p-box into a set of *n* horizontal slices. Each slice is fully described by a probability mass (the vertical height of the slice) and an interval corresponding to lower and upper bounds on a subset of the domain of the uncertain quantity. A discretized p-box is shown in Figure 5. The p-box $\boxed{X}$ is discretized into four slices, each of probability mass 0.25. The discretized p-box contains the true p-box. The second slice from the bottom is associated with the closed interval $\left[\underline{x}_2, \overline{x}_2\right]$.

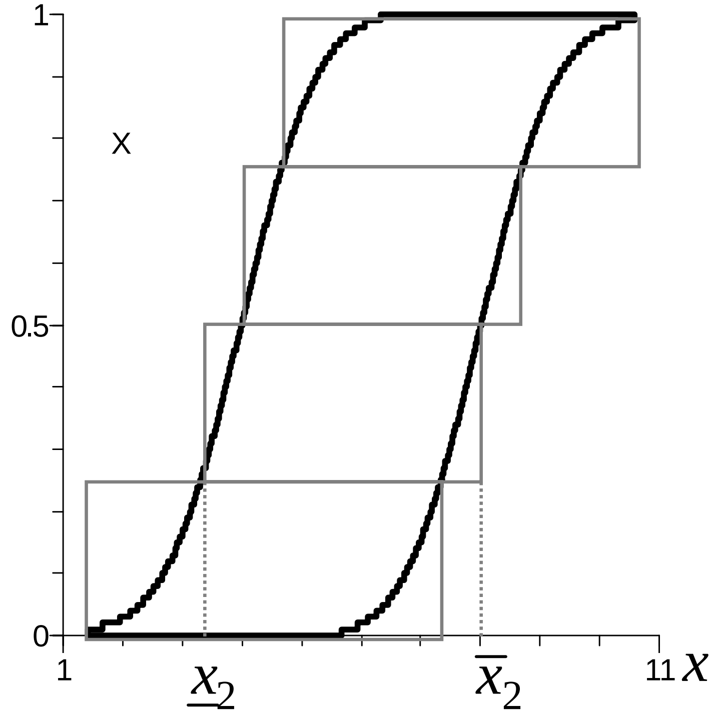

Figure 5. A discretized p-box.

The algorithms of Williamson and Downs [27] provide deterministic and rigorous approximations of binary functions of discretized p-boxes. Any binary function of p-boxes can be analyzed in the context of a Cartesian product of the input p-boxes. This utilization of a Cartesian product was first proposed by Yager [52] for the convolution of Dempster-Shafer structures. Consider some binary function of uncertain quantities, $\boxed{Z} = f\left(\boxed{X}, \boxed{Y}\right)$. If the uncertain inputs, $\boxed{X}$ and $\boxed{Y}$, are discretized into *m* and *n* slices, respectively, then the resultant Cartesian product is an *mn*-element list of interval-mass pairs. Suppose that the discretization slices for the two inputs are evenly distributed along the cumulative probability axis. Then every slice for $\boxed{X}$ has probability mass

$1/m$, and every slice for $\boxed{Y}$ has probability mass $1/n$. Also, we will label the slices of $\boxed{X}$ as the intervals $x_i = [\underline{x}_i, \overline{x}_i], i = 1,...,m$ and the slices of $\boxed{Y}$ as $y_j = [\underline{y}_j, \overline{y}_j], j = 1,...,n$. Then the ij^{th}-element of the Cartesian product for $\boxed{Z} = f\left(\boxed{X}, \boxed{Y}\right)$, assuming statistical independence between the inputs, is $\left(f(x_i, y_j), \frac{1}{m} \times \frac{1}{n}\right)$ where $f(x_i, y_j)$ is the interval extension of the intervals x_i and y_j. The resultant p-box, $\boxed{Z}$, in the case of independence, is then an ordered stacking of the slices $\left(f(x_i, y_j), \frac{1}{m} \times \frac{1}{n}\right)$ for $i = 1,...,m$ and $j = 1,...,n$.

Although this result for the statistically independent case is useful, a more powerful quality of DBC methods is the ability to determine probability bounds in the case of unknown dependence between the inputs. Although engineers often lack knowledge about the true dependence between uncertain quantities, they still tend to assume *independence* in their models—an assumption that likely results in incorrect conclusions. In the case of unknown dependence, the method of Berleant [35] determines dependency bounds by linear programming to maximize and minimize the probability masses of the elements of the Cartesian product. The method of Williamson and Downs [27] is derived using the concept of the copula [53]. A copula is a function $C:[0,1] \times [0,1] \to [0,1]$ that can be used to describe dependence between two random variables. Suppose that the marginal distribution functions of the random variables *X* and *Y* are known to be $u = F_X(x)$ and $v = F_Y(y)$. If the dependence between *X* and *Y* can be modeled with the known copula *C*, then the joint distribution for *X* and *Y* is

$H(x,y)=C\left(F_X(x),F_Y(y)\right)$. All dependencies between random variables are contained within the Frechet-Hoeffding bounds [54, 55]. That is,

$$\max(u+v-1,0)\le C(u,v)\le \min(u,v)$$

for any copula, *C*. Utilizing this fact about dependence, Williamson and Downs derived DBC algorithms for the basic arithmetic operations $\{+,-,\times,\div\}$. For instance, extending the Frechet-Hoeffding bounds to the sum of two p-boxes results in the following relation for the upper and lower bounding curves of $\boxed{Z}=\boxed{X}+\boxed{Y}$

$$\begin{aligned}\underline{F}_Z(z)&=\max_{x+y=z}\left\{\max\left[\underline{F}_X(x)+\overline{F}_Y(y)-1,0\right]\right\}\\ \overline{F}_Z(z)&=\min_{x+y=z}\left\{\min\left[\overline{F}_X(x)+\overline{F}_Y(y),1\right]\right\}\end{aligned}.$$

For a detailed derivation of these formulae, as well as formulae for the other basic arithmetic operations, see [27].

3.3 Inadequacies of the available methods

In Section 3.1, three approaches for propagating uncertainty were mentioned—1) analytical distribution convolutions, 2) dependency bound convolutions, and 3) double loop sampling. At present, the first two of these approaches are incapable of solving the problem stated in Section 2.5. The third approach works but can be computationally expensive. Before an alternative method is proposed, it is necessary to explain why the available methods are insufficient for a large class of engineering design problems. This leads to a confirmation of Hypothesis 1.

The ideal solution would be to formulate and analytically solve the appropriate set of distribution convolutions. The transformation methods described in Springer [42] are limited to basic binary algebraic operations for independent variables with a few distribution shapes. All these methods are impossible or very cumbersome for black box computer models where the functional relationship is not given explicitly as a sequence of binary operations. Analytical methods appear even less tractable in the presence of imprecision where *sets* of distributions must be convolved.

DBC methods are considerably more promising, but they must overcome at least three obstacles before they can be used in engineering design. First, both of these approaches depend strongly on the methods of interval arithmetic for which the presence of repeated variables can result in over-conservative (i.e. not best-possible) solution bounds. While sub-interval reconstitution methods work well for low-dimensional problems [23, 56, 57], they are prohibitively expensive in realistic engineering problems with a large number of imprecise quantities. Although what counts as prohibitively expensive is problem specific, it is of interest to note that Risk Calc only allows sub-interval reconstitution for problems with up to four uncertain inputs. A second challenge for DBC is that black box propagation of intervals is still only workable for quasilinear problems. Trejo and Kreinovich have developed a randomized algorithm for propagating interval uncertainty through black-box models [58, 59], but the method assumes that the black-box model is quasilinear in the region of sampling. It is unclear at this point if this black box method has general applicability towards complex engineering analysis models. The third challenge that must be overcome for DBC to be applicable towards realistic design problems involves the limited class of operations for which DBC algorithms are available. Currently, DBC algorithms are only available for solving binary arithmetic $\{+,-,\times,\div\}$ and elementary functions such as exponentials and trigonometry functions. Therefore, in order to propagate p-boxes through a mathematical model using DBC requires that the model be known (i.e., not a black box) as a series of basic binary operations and elementary functions.

In order for the dependency bounds and distribution envelope methods to be applicable for engineering design, methods for better propagating intervals through black box models in the presence or many repeated variables need to be developed. If these conditions were met, it would then be necessary to convince the producers of the standard engineering analysis software to incorporate these methods into their products. While this seems possible, and is perhaps the most desirable solution, our concern is more immediate: *how can engineers use the tools available to them today to make realistic design decisions under imprecise uncertainty?*

One very simple and easy-to-implement approach is a double loop sampling routine. A formal discussion of a special type of double loop sampling routine will be presented in Section 4.1. Double loop sampling involves random sampling across the two dimensions of an uncertain quantity [43, 45]. Since sampling routines only require evaluations at scalar values of the set of uncertain variables, these approaches meet our requirement of being compatible with black box utility models. For high-dimensional problems, double loop sampling can become prohibitively expensive because the sampling in the outer loop does not retain the computational advantages of Monte Carlo simulation. Specifically, the outer loop uses probabilistic sampling to approximate the minimum and maximum values that result from the inner loop. To approximate these bounds accurately an increasingly large number of samples must be taken as the dimensionality of the problem increases. As a possible solution to this, some authors have suggested a sensitivity analysis approach [46]. In Section 4.2, we present an alternative means of speeding up double loop sampling in which one of the sampling loops is replaced by an optimization algorithm.

In summary, this chapter has lead to two conclusions. First, DBC can be a useful method for PBA in engineering design if and only if the decision model used is given as a sequence of basic arithmetic operations or elementary functions and if there is only a small number of repeated uncertain variables—this is just a simple fact resulting from the available DBC algorithms. This statement confirms Hypothesis 1. The second conclusion of this chapter is that some form of double loop sampling might be useful for engineering design since sampling algorithms are compatible with black box models. In the next chapter, we will begin to address Research Question 2 using double loop sampling as our starting point.

CHAPTER 4: BLACK BOX METHODS FOR UNCERTAINTY PROPAGATION

To answer Research Question 2, we need to find alternative methods for p-box propagation that are black box compatible and are computationally inexpensive. DBC does not meet this requirement. In this chapter, we introduce three alternative black box approaches for propagating uncertain quantities. It is not enough for a method to be black box compatible, though. In addition, the method must be computationally inexpensive. In Chapter 5, we examine the computational feasibility of the three black box methods as compared with DBC.

4.1 Double Loop Sampling (DLS)

Double loop sampling (DLS) is applicable towards black box models with *parameterizable* p-box inputs. DLS involves two layers of sampling: one associated with distribution parameters and the other associated with the distributions themselves. In effect, double loop sampling involves sampling from an analytic distribution whose parameters have been established by sampling. The outer loop will be called the *parameter* loop since it involves sampling different values for the set of distribution parameters for all of the uncertain quantities. The inner loop will be called the *probability* loop since it involves sampling from precise probability distribution functions. The probability loop is essentially a Monte Carlo simulation to determine the expectation of a function of several random variables.

Suppose that we have some black-box model that maps a vector of inputs, $\mathbf{x} \in \mathbb{R}^n$, to some output $z = f(\mathbf{x})$. Suppose further that the inputs are modeled as a vector of parameterized p-boxes, $\boxed{\mathbf{X}}^P$. The first step in DLS is to define a vector containing all distribution parameters for all of the uncertain quantities. Each $\boxed{X}_j^P \in \boxed{\mathbf{X}}^P$ has associated with it a set of imprecise parameters stored in the vector $\boldsymbol{\theta}^j \in \left[\underline{\boldsymbol{\theta}}^j, \overline{\boldsymbol{\theta}}^j\right]$. Specifically,

$$\boxed{X}_j^P = \left\{ F_{X_j}(x_j; \boldsymbol{\theta}^j) : \underline{\boldsymbol{\theta}}^j \le \boldsymbol{\theta}^j \le \overline{\boldsymbol{\theta}}^j \right\}$$

where F_{X_j} is the parameterized distribution function describing the elements of $\boxed{X}_j^P$. The number of parameters associated with a single uncertain quantity, x_j, is denoted $q_j = length\left(\boldsymbol{\theta}_j\right)$. For notational convenience, it is desirable to combine the $q_j \times 1$ vectors for each of the uncertain quantities, x_j, into a single vector representing all relevant distribution parameters. This super-vector will be denoted $\boldsymbol{\Theta}$. Also, by extension from the lower and upper bounds of the sub-vectors, lower and upper bounds of the super-vector can be determined. That is, the vector of distribution parameters is constrained such that $\boldsymbol{\Theta} \in \left[\underline{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Theta}}\right]$. These parameter bounds are important as they represent all of the imprecision in the model. The purpose of the parameter loop is to experiment with these imprecise distribution parameters in order to approximate the smallest and largest expected values of the uncertain quantity *Z*.

In the parameter loop, the space of the parameter vector, $\mathbf{\Theta}$, is explored by random sampling. A sampled point in the parameter space corresponds to a set of precise distributions for all uncertain quantities and will be denoted $\mathbf{\Theta}^a$.

The probability loop uses these precise distribution functions to solve a purely probabilistic sampling problem—a Monte Carlo approximation of the expectation of a function of random variables. Specifically, the probability loop uses samples from the distributions defined by $\mathbf{\Theta}^a$ to compute an expected value of *Z*. The expected value of *Z* given some $\mathbf{\Theta}^a$ is denoted E_a. The process of computing an E_a is repeated for *s* randomly sampled points in the parameter space, $\mathbf{\Theta}$. That is, the DM computes an E_a corresponding to some $\mathbf{\Theta}^a$ for $a = 1,\ldots,s$.

If the sampled parameter vectors sufficiently cover the parameter space, then the largest and smallest values of the set $\{E_a\}$ can be used to approximate the lower and upper expected values of *Z*. It is not possible to know *a priori* if the parameter samples are sufficient. However, some study can be made of convergence by solving the problem with multiple sample sizes. Formally, the lower and upper expected values are approximated by the expressions

$$\underline{E}_{\boxed{\mathbf{X}}^P}\left[\boxed{Z}^P\right] \approx \min_{1\le a\le s} E_a$$

and

$$\overline{E}_{\boxed{\mathbf{X}}^P}\left[\boxed{Z}^P\right] \approx \max_{1\le a\le s} E_a .$$

To summarize, the DLS algorithm is given as follows:

Step 1. Randomly select some point $\mathbf{\Theta}^a \in \left[\underline{\mathbf{\Theta}}, \overline{\mathbf{\Theta}}\right]$.

Step 2. Randomly select some point $\mathbf{x}^{a,b}$ from the distributions defined by $\mathbf{F}_{\mathbf{X}}(\mathbf{x};\mathbf{\Theta}^{a})$ where $\mathbf{F}_{\mathbf{X}}(\mathbf{x};\mathbf{\Theta})$ is the vector of cumulative distribution functions for the uncertain quantities in the vector $\mathbf{X}$ and specified by the parameters in the vector $\mathbf{\Theta}$. Specifically, compute $\mathbf{x}^{a,b}=\mathbf{F}_{\mathbf{X}}^{(-1)}(\mathbf{p}^{b};\mathbf{\Theta}^{a})$ where $\mathbf{p}^{b}\in[0,1]^{n}$.

Step 3. Compute the value of the output for the point $\mathbf{x}^{a,b}$: $z_{a,b}=f(\mathbf{x}^{a,b})$.

Step 4. Go back to Step 2; repeat for $b=1,\ldots,t$.

Step 5. Compute the expected value of the output for $\mathbf{\Theta}^{a}$:

$$E_a=\frac{1}{t}\sum_{b=1}^{t} z_{a,b}$$

Step 6. Go back to Step 1; repeat for $a=1,\ldots,s$.

Step 7. Approximate the lower and upper expected values of the output, *Z*:

$$\underline{E}_{\boxed{\mathbf{X}}^{P}}\left[\boxed{Z}^{P}\right]\approx\min_{1\le a\le s}E_a$$

and

$$\overline{E}_{\boxed{\mathbf{X}}^{P}}\left[\boxed{Z}^{P}\right]\approx\max_{1\le a\le s}E_a.$$

Although DLS is black box compatible, it is by no means computationally inexpensive. This is primarily due to the fact that the parameter loop uses sampling to solve a problem that sampling is not well-suited to solve. Sampling techniques are useful for approximating integrals such as the expectation of a function of several random variables. The parameter loop uses random sampling to approximate the minimum and maximum expected values of the output where these expectations are approximated using sampling from analytic distributions in the inner loop. For high dimensional problems, it becomes increasingly unlikely that a sampled point in the parameter space will result in a close approximation of $\underline{E}_{\boxed{\mathbf{X}}^P}$ or $\overline{E}_{\boxed{\mathbf{X}}^P}$. This is due to the fact that we are trying to locate two small regions corresponding to the points in the parameter space that result in sufficiently close approximations of the lower and upper expected values of the output. As the dimensionality of the problem increases, the size of the sample space increases exponentially, and the probability of sampling the two small regions decreases exponentially.

4.2 Optimized Parameter Sampling (OPS)

In an attempt to overcome the prohibitive cost of double loop sampling for high-dimensional problems, *optimized parameter sampling* (OPS) replaces the sampling in the parameter loop with an optimization algorithm. The parameter loop optimizer is used to locate the points $\mathbf{\Theta}^l$ and $\mathbf{\Theta}^u$ in the parameter space that result in the smallest and largest expected values of the output, E_l and E_u. These expectations are then used to approximate the lower and upper expected values for the output Z:

$$\underline{E}_{\boxed{\mathbf{X}}^P}\left[\boxed{Z}^P\right] \approx E_l$$

$$\overline{E}_{\boxed{\mathbf{X}}^P}\left[\boxed{Z}^P\right] \approx E_u .$$

Essentially, the modified double-loop sampling method is the same as pure double loop sampling except that $\mathbf{\Theta}^a$ is updated intelligently thus saving function evaluations. The modified approach requires the solution of the following two optimization problems:

$$(1)\ \underset{\Theta\in\left[\underline{\Theta},\bar{\Theta}\right]}{\text{minimize}}\ E = g\left(\boldsymbol{\Theta}\right) \Rightarrow E_l$$

$$(2)\ \underset{\Theta\in\left[\underline{\Theta},\bar{\Theta}\right]}{\text{maximize}}\ E = g\left(\boldsymbol{\Theta}\right) \Rightarrow E_u$$

where $\underline{\boldsymbol{\Theta}}$ and $\bar{\boldsymbol{\Theta}}$ are the upper and lower bounds on the parameter space and the $g\left(\boldsymbol{\Theta}\right)$ represents the value returned by the probability loop computation.

Solving these optimization problems numerically poses two challenges related to the objective function, $E = g\left(\boldsymbol{\Theta}\right)$: 1) g is approximated non-deterministically and 2) g could have local extrema. Different random variates in the probability loop will result in different approximations of E_a for a given vector of parameters, $\boldsymbol{\Theta}^a$. That is, since E_a is approximated stochastically, neighboring values of E_a will likely not follow the true continuous surface $E = g\left(\boldsymbol{\Theta}\right)$. For gradient-based optimizers, this is problematic since the approximation to the objective function will develop sharp local gradients. One possible solution to challenge 1) is to use the same set of random variates for each step of the optimization algorithm. This of course introduces a bias into the resulting E_l and E_u, but this bias can be made arbitrarily small by increasing the number of probability loop samples, *t*. The second challenge is due to the nature of the true objective function. For realistic engineering problems, $E = g\left(\boldsymbol{\Theta}\right)$ is often multi-modal. One possible solution to challenge 2) is to repeat the optimizations from multiple starting points, $\boldsymbol{\Theta}^1$. Both of these solutions have proven to be effective in the design of an off-road vehicle gearbox—see Section 5.3 and [26]. However, more examples would be needed to provide sufficient support for the effectiveness of using these two solutions.

For many problems, the OPS algorithm described above will more efficiently locate the minimal and maximal sets of distribution parameters. However, the modified approach retains some of the weaknesses of DLS and even introduces some new difficulties.

Like DLS, OPS assumes a known dependence between the uncertain quantities involved in the computation. If a parameterized joint distribution function of all uncertain quantities were available, it would be compatible with either approach, but for practical problems it is almost never the case that the DM knows a fully characterized joint distribution. The dependency bounds approach, as described in Chapter 3, makes no assumptions about the dependence between uncertain bounds. Indeed, dependency bounds are *best-possible* bounds that contain the results of the computation under any possible case of dependency. For problems in which the computation involves variables with possibly strong but unknown dependency, DBC maintains a distinct advantage over both DLS and OPS.

Also, like DLS, the OPS method might become too computationally expensive for high-dimensional problems. Replacing the *parameter* loop with an optimizer should result in decreased computational cost due to the decreased number of function evaluations, but optimization over a high-dimensional space can itself remain costly. The most that can be claimed of the OPS approach is that it allows for the solution of a wider class of problems than DLS.

Although OPS retains some of the weaknesses of DLS, it also introduces an additional difficulty. Specifically, the functions to be optimized, $E = g\left(\mathbf{\Theta}\right)$, are complex and non-linear and therefore multi-modal. This means that the optimization problems become global optimization problems. Depending on the complexity of the global optimization problem, OPS might be computationally infeasible. Although many sophisticated algorithms for solving global optimization problems have been developed (see [60-62]), for many problems with relatively few local minima, it is often sufficient to repeat the optimization from multiple starting points.

4.3 P-box Convolution Sampling (PCS)

P-box convolution sampling (PCS) is another method for propagating uncertain quantities through black-box models, but, in contrast to DLS and OPS, it is applicable towards non-parameterized p-box inputs. For most applications, compatibility with non-parameterized inputs is a distinct advantage since parameterizing inputs often makes unjustifiable assumptions. In this sense, PCS is similar to DBC. PCS is also similar to OPS but with reversed loops. In other words, PCS arrives at approximations to the lower and upper expectations of the uncertain output Z by sampling in the outer loop, and optimizing in the parameter loop.

Using the same black box model as before, $z = f(\mathbf{x})$, suppose that the uncertain inputs are a vector of non-parameterized p-boxes, $\boxed{\mathbf{X}}$. Each p-box in $\boxed{\mathbf{X}}$ is the set of distribution functions

$$\boxed{X}_j = \left\{ F_{X_j}(x_j) : \underline{F}_{X_j}(x_j) \leq F_{X_j}(x_j) \leq \overline{F}_{X_j}(x_j) \right\}$$

for $j = 1, \ldots, n$. In the outer loop, an interval is sampled from each of the input p-boxes. To determine an interval sample from a p-box, the inverses of the bounding functions must be known. An inverse bounding function for a lower bounding curve on the random variable X is defined by

$$\underline{F}_X^{(-1)}(p) \equiv \left\{x : \underline{F}_X(x) = p\right\}$$

where p is some cumulative probability in the interval [0,1]. For strictly increasing bounding functions, the resultant set will be a singleton. Since the lower bounding function corresponds to larger values of the uncertain quantity, and the upper bounding function corresponds to smaller values of the uncertain quantity, an interval sampled from a p-box is calculated as $\left[\underline{x}, \overline{x}\right] = \left[\overline{F}_X^{(-1)}(p), \underline{F}_X^{(-1)}(p)\right]$. An iteration of the outer loop in a PCS simulation consists of determining such an interval for each of the input p-boxes. This results in an interval vector of the form $\left[\underline{\mathbf{x}}^a, \overline{\mathbf{x}}^a\right]$ for the a^{th} iteration of the outer loop.

The inner loop then needs to propagate this collection of input intervals through the black box model. That is, the interval

$$z_a = \left[\underline{z}_a, \overline{z}_a\right] = \left[f\left(\left[\underline{\mathbf{x}}^a, \overline{\mathbf{x}}^a\right]\right)\right]$$

needs to be computed. If the function f corresponds to a known interval arithmetic operation, then the methods of interval arithmetic can be applied directly. An alternative, and more generally applicable method of computing z_a, is to solve two constrained optimization problems

$$\begin{array}{l} (1)\ \underset{\mathbf{x} \in \left[\underline{\mathbf{x}}^a, \overline{\mathbf{x}}^a\right]}{\text{minimize}}\ f(\mathbf{x}) \Rightarrow \underline{z}_a \\ (2)\ \underset{\mathbf{x} \in \left[\underline{\mathbf{x}}^a, \overline{\mathbf{x}}^a\right]}{\text{maximize}}\ f(\mathbf{x}) \Rightarrow \overline{z}_a \end{array}.$$

The process is repeated for $a=1,\ldots,s$ samples from the input p-boxes, and approximations to the lower and upper expectations of Z is determined by taking the expectations of the resultant set of lower and upper interval bounds. Mathematically,

$$\underline{E}_{\boxed{\mathbf{X}}}\left[\boxed{Z}\right] \approx \tfrac{1}{s}\sum_{a=1}^{s} \underline{z}_a$$

and

$$\overline{E}_{\boxed{\mathbf{X}}}\left[\boxed{Z}\right] \approx \tfrac{1}{s}\sum_{a=1}^{s} \overline{z}_a \ .$$

To summarize, the PCS algorithm is described as follows:

Step 1. Randomly sample an interval from each of the input p-boxes: $\left[\underline{\mathbf{x}}, \overline{\mathbf{x}}\right] = \left[\overline{\mathbf{F}}_{\mathbf{X}}^{(-1)}(\mathbf{p}^a), \underline{\mathbf{F}}_{\mathbf{X}}^{(-1)}(\mathbf{p}^a)\right]$ where $\mathbf{p}^a \in [0,1]^n$ is randomly chosen.

Step 2. Propagate this vector of input intervals through the black box model to approximate an interval on the output: $\left[\underline{z}_a, \overline{z}_a\right]$ found by

$$\text{(1)}\ \underset{\mathbf{x}\in\left[\underline{\mathbf{x}}^a, \overline{\mathbf{x}}^a\right]}{\text{minimize}}\ f(\mathbf{x}) \Rightarrow \underline{z}_a$$

$$\text{(2)}\ \underset{\mathbf{x}\in\left[\underline{\mathbf{x}}^a, \overline{\mathbf{x}}^a\right]}{\text{maximize}}\ f(\mathbf{x}) \Rightarrow \overline{z}_a$$

Step 3. Go back to 1; repeat for $a=1,\ldots,s$.

Step 4. Compute approximations to the lower and upper expected values of the output:

$$\underline{E}_{\boxed{\mathbf{X}}}\left[\boxed{Z}\right] \approx \tfrac{1}{s}\sum_{a=1}^{s} \underline{z}_a$$

and

$$\overline{E}_{\boxed{\mathbf{X}}}\left[\boxed{Z}\right] \approx \frac{1}{s}\sum_{a=1}^{s}\overline{z}_a \; .$$

4.4 Summary of Methods

The three different methods described in this chapter, DLS, OPS, and PCS, are all *black box compatible* in that they can be used to propagate uncertain inputs through structurally unknown or complex models. These three methods stand in contrast to DBC, which rigorously contains the true resultant p-box, but which cannot function through black box models. If the mathematical model relating inputs to outputs does not need to be treated as a black box, the DBC method is generally preferable because of its relatively low computational cost and its assured rigor.

A further classification of the methods can be made in terms of how the inputs can be modeled. The OPS and DLS methods require that the inputs be *parameterized* p-boxes. For some problems, parameterized p-boxes might be the best representation of an uncertain quantity—for instance, in standard statistical parameter estimation where data gives rise to confidence intervals on the true parameter values for some random variable with a known distribution. However, *non-parameterized* p-boxes can more generally characterize imprecision and variability. Both PCS and DBC are capable of propagating non-parameterized p-boxes through mathematical models. A diagram of the classifications of the four methods is shown in Figure 6.

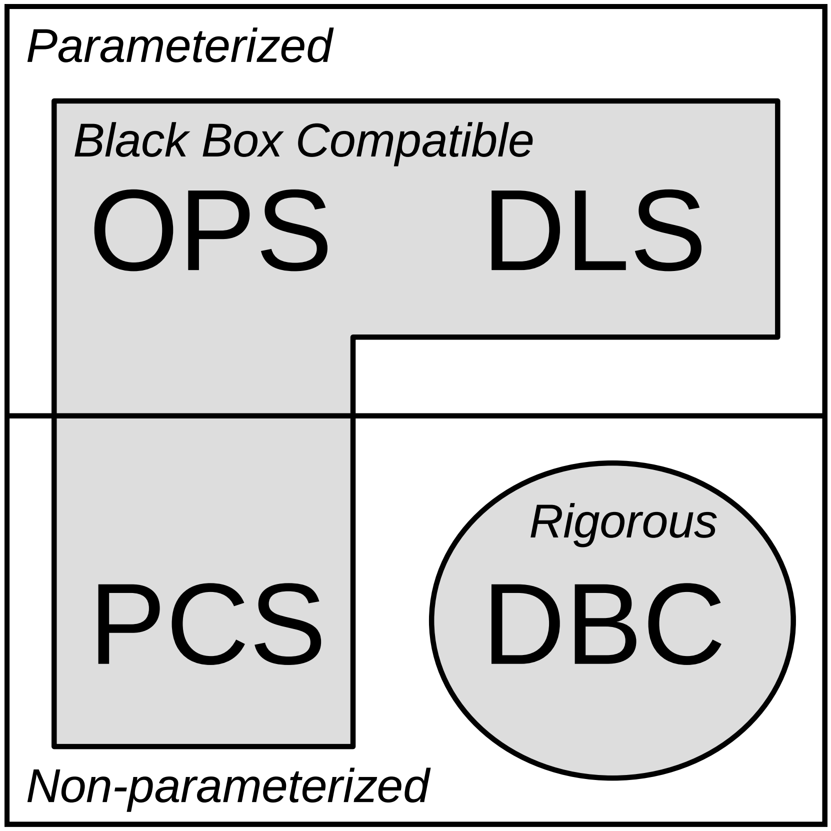

Figure 6. Classification of four methods examined in this paper.

In this chapter, three black box methods have been presented and classified in terms of what types of problems they can be applied towards. This chapter has provided a partial response to Research Question 2. Specifically, we have identified three alternative methods for PBA that function through black box analysis models. Before these methods can be recommended for engineering design practice, it is necessary to study their relative computational costs. In the next chapter, the three black-box methods, in addition to DBC, will be applied towards the solution three uncertainty propagation problems.

CHAPTER 5: NUMERICAL EXAMPLES

The primary purpose of this chapter is to demonstrate the computational cost of the three alternative black box methods for PBA that were proposed in the previous chapter. To this end, the black box methods—in addition to DBC—were applied in the solution of two numerical examples. The first numerical example is a sum of two normally distributed p-boxes. The second example is a transient heat transfer analysis of a thermocouple. A secondary purpose of this chapter is to demonstrate the decision criteria involved in decision making under uncertainty. To this end, a gearbox design problem was solved using the DLS algorithm.

With regards to the research questions, it needs to be determined if the three black box methods are computationally inexpensive enough to be useful for realistic engineering design problems. If it is found that they are, then we will have found further evidence to support Hypothesis 2—specifically, that the alternative methods are compatible with the demands of engineering design.

5.1 Example 1: Simple sum of normal p-boxes

The first numerical example that we study is the simple sum of two uncertain quantities: $Z = A + B$. With the exception of DBC, the propagation methods examined do not currently allow for the case of unknown dependence between uncertain quantities. Therefore, it is assumed that the two addends are stochastically independent. Also, since the two classes of methods, parameterized and non-parameterized, are fundamentally different, we can only directly compare methods of the same class. OPS can be compared to DLS, and PCS can be compared to DBC. To highlight the difference between parameterized and non-parameterized p-boxes, we solved the sum for two sets of input p-boxes, *A* and *B*.

The first set of uncertain inputs is the normal, parameterized p-boxes

$$\boxed{A}^P \sim Normal\left(\mu_A = [4,8], \sigma_A = [1,3]\right)$$

$$\boxed{B}^P \sim Normal\left(\mu_B = [5,6], \sigma_B = [7,8]\right).$$

The bounding functions for these two p-boxes are shown in Figure 7.

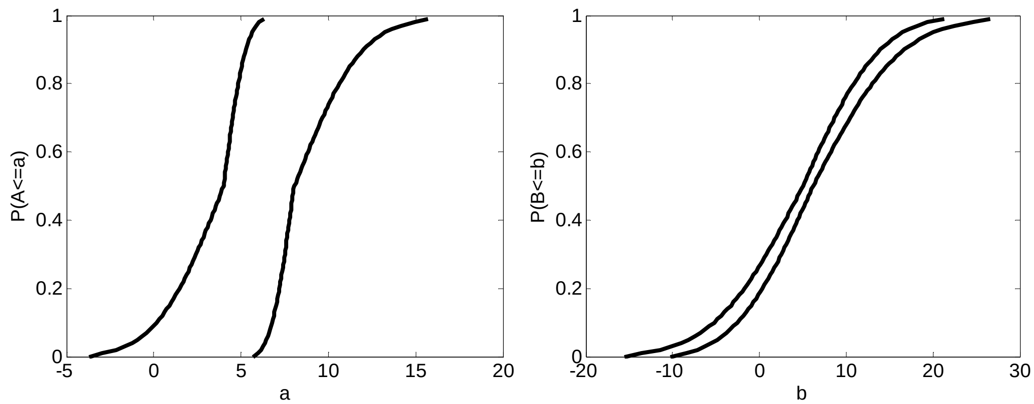

Figure 7. Bounding functions for *A* and *B*.

The resultant p-box $\boxed{Z}^P$ is shown in Figure 8, but the DLS and OPS methods can be used to approximate only the lower and upper expectations of the uncertain quantity *Z*. In decision making, expected utility is the primary metric of interest. Therefore, the different methods are compared with respect to their ability to compute the expected values of the model outputs. Because the problem is linear, an analytical solution to the interval $E^P[Z] = \left[\underline{E}^P[Z], \overline{E}^P[Z]\right]$ is available—note that the superscript *P* is used to denote the fact that the expectation is associated with the parameterized inputs. Generally, the interval of expected utility for $Z = A + B$ is

$$E[Z] = \left[\min_{\substack{F_A \in \boxed{A} \\ F_B \in \boxed{B}}} \left(E\left[F_A\right] + E\left[F_B\right] \right), \max_{\substack{F_A \in \boxed{A} \\ F_B \in \boxed{B}}} \left(E\left[F_A\right] + E\left[F_B\right] \right) \right]$$

where F_A and F_B are distribution functions in the p-boxes $\boxed{A}$ and $\boxed{B}$. For the case of the parameterized p-boxes, evaluating the bounds of the interval $E^P[Z]$ is trivial:

$$\underline{E}^P[Z] = \min_{\substack{F_A \in \boxed{A}^P \\ F_B \in \boxed{B}^P}} \left(E\left[F_A\right] + E\left[F_B\right] \right) = \underline{\mu}_A + \underline{\mu}_B = 4 + 5 = 9$$

$$\overline{E}^P[Z] = \max_{\substack{F_A \in \boxed{A}^P \\ F_B \in \boxed{B}^P}} \left(E\left[F_A\right] + E\left[F_B\right] \right) = \overline{\mu}_A + \overline{\mu}_B = 8 + 6 = 14$$

We therefore have an exact solution for the expectation of the sum of the parameterized p-boxes, $E_e^P[Z] = \left[9,14\right]$. The two parameterized methods, DLS and OPS, can be compared directly with this interval.

The two non-parameterized methods, PCS and DBC, cannot solve the problem with parameterized inputs. Therefore, another set of inputs is needed. The set of non-parameterized p-boxes used will share the same bounding functions associated with $\boxed{A}^P$ and $\boxed{B}^P$. This means that $\boxed{A}^P \subset \boxed{A}$ and $\boxed{B}^P \subset \boxed{B}$, but $\boxed{A}$ and $\boxed{B}$ contain *all* possible non-decreasing functions found within the bounding functions while $\boxed{A}^P$ and $\boxed{B}^P$ contain only normal distributions. Accordingly, $\boxed{A} + \boxed{B}$ will result in a different p-box than is found by $\boxed{A}^P + \boxed{B}^P$. Plots of the bounding functions are shown in Figure 7.

These bounding functions are identical for both the parameterized and general p-box inputs. Plots of $\boxed{Z}^{P}$ and $\boxed{Z}$, the p-box resulting from non-parameterized inputs, are shown in Figure 8.

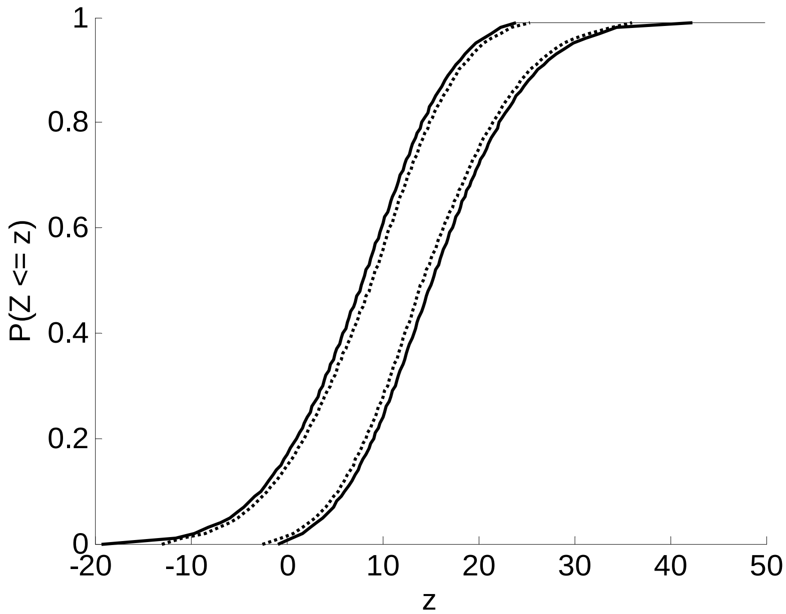

Figure 8. Resultant p-boxes $\boxed{Z}$ (solid line) and $\boxed{Z}^{P}$ (dashed line).

Although the interval of the expected value of Z will be different for the case of the non-parameterized p-boxes, an exact solution can still be approximated accurately by numerically integrating the analytical solution. The minimum and maximum expectations of Z for the non-parameterized p-boxes will be shifted slightly outward as compared to the expectations associated with the parameterized p-boxes. This is due to the fact that the non-parameterized p-boxes contain additional distribution functions.

Therefore, the minimum expectation of any distribution $F_a \in \boxed{A}$ is the expectation of the leftmost bounding function in $\boxed{A}$. Mathematically,

$$F_{a,\min} \sim \begin{cases} Normal(\mu = 4, \sigma = 3), & a \le 4 \\ Normal(\mu = 4, \sigma = 1), & a > 4 \end{cases}.$$

Similar expressions correspond to $F_{a,\max}$, $F_{b,\min}$, and $F_{b,\max}$. The exact solution for the lower expectation on Z with non-parameterized inputs can be found by

$$\begin{aligned} \underline{E}_e[Z] &= E\left[F_{a,\min}\right] + E\left[F_{b,\min}\right] \\ &= \left(\int_{-\infty}^{\underline{\mu}_a} a \cdot f\left(a; \underline{\mu}_a, \overline{\sigma}_a\right) da + \int_{\underline{\mu}_a}^{\infty} a \cdot f\left(a; \underline{\mu}_a, \underline{\sigma}_a\right) da\right) \\ &\quad + \left(\int_{-\infty}^{\underline{\mu}_b} b \cdot f\left(b; \underline{\mu}_b, \overline{\sigma}_b\right) db + \int_{\underline{\mu}_b}^{\infty} b \cdot f\left(b; \underline{\mu}_b, \underline{\sigma}_b\right) db\right) \end{aligned}$$

where the function f is the normal probability density function with mean μ and standard deviation σ. A similar expression can be written for $\overline{E}_e[Z]$. Using numerical integration it is found that $E_e[Z] \cong \left[7.80317, 15.19683\right]$. We take this approximation to be the *exact* solution with which we will compare the results arrived at using the black box methods presented in Chapter 4.

5.1.1 Criteria for Comparison of the Methods

The methods will be compared in terms of accuracy and computational cost. Since the parameterized methods, DLS and OPS, essentially solve a different problem than the non-parameterized methods, they cannot be compared with the DBC and PCS methods.

Because of the symmetry of the simple sum problem, we will limit our error analysis to the lower bound expectation of *Z*. Since exact solutions are known for both the parameterized and non-parameterized versions of the problem, the absolute error can be computed as $\delta \underline{E}_M^P[Z] = \left| \underline{E}_e^P[Z] - \underline{E}_M^P[Z] \right|$ for the parameterized sum and $\delta \underline{E}_M[Z] = \left| \underline{E}_e[Z] - \underline{E}_M[Z] \right|$ for the non-parameterized sum. The subscript M denotes the particular method employed. For the three non-deterministic methods (DLS, OPS, and PCS), 100 simulations will be run to allow for a statistical analysis of the results. The data of the 100 simulations will be summarized by the mean and the 5th and 95th percentiles of $\delta \underline{E}_M^P[Z]$ or $\delta \underline{E}_M[Z]$.

Computational cost is compared in terms of the number of function evaluations of the black box model. For the simple sum problem, the black box model is simply $z = a + b$. That is, given some scalar instantiations, *a* and *b*, of the uncertain quantities *A* and *B*, the black box model returns a scalar instantiation, *z*, of the uncertain quantity *Z*. For a given method, *M*, FE_M denotes the number of function evaluations called by *M* for a particular simulation. If $FE_{M_1} < FE_{M_2}$ for some equivalent degree of accuracy, then we will conclude that method M_1 is less costly than method M_2. The assumption made here is that the costs of auxiliary computations specific to the different methods are negligible when compared to the cost of a single function evaluation. This assumption is justified by the fact that in realistic engineering computations, the limiting cost is most often associated with the computer models.

All three black box methods were implemented in MATLAB, and all optimization was done using MATLAB's quasi-Newton solver, fmincon. The stopping criteria for the optimizations were chosen such the error due to optimization was dominated by the error due to sampling. For example, if the error due to sampling, for a particular problem, were found to be on the order of 10^{-2}, then the objective function tolerance in the optimizer would be set to 10^{-3}. All DBC solutions were found using Risk Calc [32].

5.1.2 Parameterized Methods

DLS simulations were run for inner loop sample sizes of $s = 10,\ 100,\ 500,\ \text{and}\ 1000$ and outer loop sample sizes of $t = 10,\ 100,\ 500,\ \text{and}\ 1000$. For both loops, the number of samples was limited to less than 1000 because larger sample sizes would, for realistic engineering models, require prohibitively expensive DLS simulations.

Table 1. Summary of error results for 100 DLS simulations.

		t			
s		**10**	**100**	**500**	**1000**
10	*5th Perc. Err.*	0.091524	0.23567	0.28015	0.272832
	Mean Error	0.610953	0.633381	0.694595	0.667497
	95th Perc. Err.	1.246041	1.06407	1.056224	1.098929
100	*5th Perc. Err.*	0.028717	0.054525	0.078058	0.069221
	Mean Error	0.277383	0.230073	0.230878	0.220753
	95th Perc. Err.	0.598889	0.42086	0.410804	0.42548
500	*5th Perc. Err.*	0.01545	0.031667	0.026953	0.034267
	Mean Error	0.185985	0.105439	0.103121	0.106333
	95th Perc. Err.	0.399423	0.199126	0.202085	0.192204
1000	*5th Perc. Err.*	0.019565	0.012621	0.020483	0.030474
	Mean Error	0.176262	0.078432	0.073936	0.086487
	95th Perc. Err.	0.388182	0.152141	0.143245	0.163601

The results of the 100 DLS simulations for each of the combinations of *s* and *t* are summarized in Table 1. For the simple sum problem, in the ranges of sample sizes examined, absolute error is much more sensitive to the number of outer loop parameter samples. This can be explained in part by the fundamental difference in how sampling is used in the outer and inner loops. In the inner loop, sampling is used to approximate an expected value which is an integral approximation. It is well known that sampling strategies are well-suited for integral approximations [28]. In the outer loop, sampling is used to approximate extrema values—for this problem, the minimum expected utility. Sampling strategies are not well-suited for this type of approximation. Therefore, it will typically take many more outer loop samples than inner loop samples to arrive at an accurate approximation of $\underline{E}_{DLS}^{P}[Z]$. In fact, when outer loop distribution samples sizes larger than 10 were used, the absolute error remains effectively constant. This indicates that, for this range of distribution sample size, the error due to insufficient parameter samples is dominating the error associated with distribution sampling. Also note that the absolute error of our experiments increases with increasing outer loop sample size for some of the cells in Table 1. Since the lower and upper error bounds (the 5^{th} and 95^{th} percentiles) do converge with increasing *t*, this increase in mean error can be attributed to statistical variation.

For any DLS simulation, the total number of function evaluations is found by $FE_{DLS} = s \times t$. The most accurate approximations to $\underline{E}_{DLS}^{P}[Z]$ were found with *s* = 1000 and $t \geq 100$. To achieve an average absolute error on the order of 0.08 (this corresponds to a relative error of approximately 0.9%), it is necessary to take at least 100,000 function evaluations. If the black-box model takes one minute to run, this would result in 70 days of computer processing time. For most applications, this is infeasible. In order to use DLS cost effectively for a one-minute black-box model, the number of function evaluations should be limited to 1000, or a half day of processing time. For the simple sum example, an average accuracy of approximately 0.28 (corresponding to a relative error of 3.1%) could be achieved with 1000 function evaluations, s = 100 and t = 10. Depending on the application, this degree of accuracy in approximating $\underline{E}_{DLS}^{P}[Z]$ might, or might not, be sufficient.

OPS simulations were run for distribution-sample sizes of 100 to 1000 in increments of 100 samples. The optimization stopping criteria for the MATLAB function fmincon were tweaked from the defaults such that the algorithm stoped if the difference between objective function values for successive iterations ("TolFun") is less than 1e-13 or if the distance between points in the parameter space for successive iterations ("TolX") is less than 1e-9. Absolute error versus t is shown in Figure 9 along with the 5th and 95th percentile curves. As is expected, accuracy increases with an increasing number of distribution samples, t. The smallest average absolute error is approximately 0.0008 (a 0.009% relative error) corresponding to 1000 distribution samples. Even using as few as 100 distribution samples results in an average absolute error of approximately 0.0124 (a 0.138% relative error) which is significantly smaller than the smallest DLS error computed above.

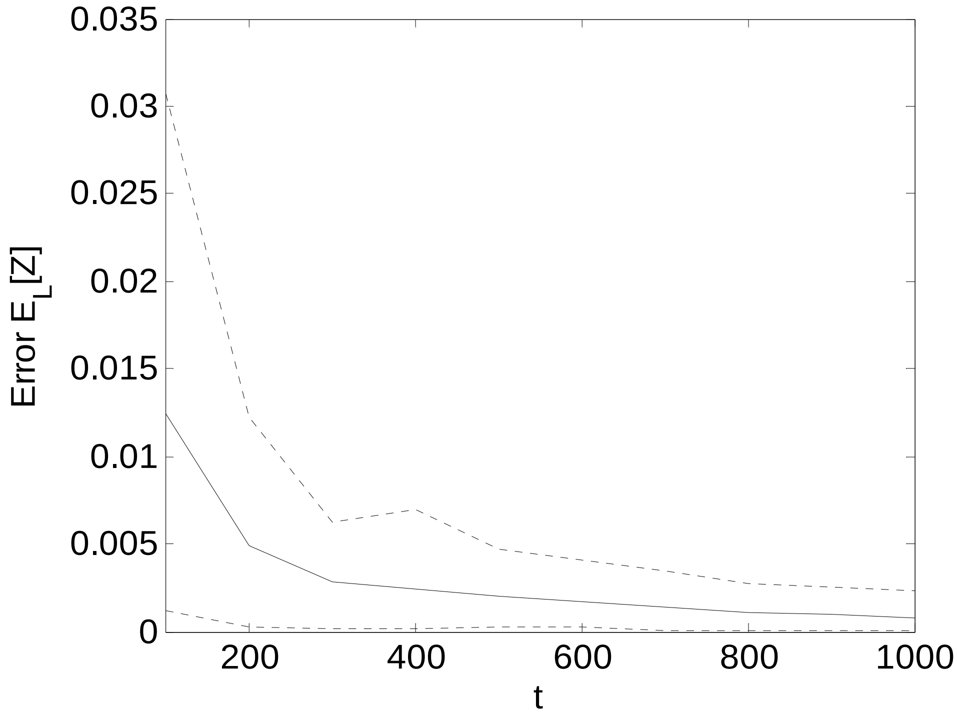

Figure 9. Absolute error versus number of distribution samples.

The number of function evaluations associated with an OPS simulation is equal to the total number of optimization function calls in the outer loop times the number of distribution samples in the inner loop. Depending on the optimization starting point and the set of random deviates used in sampling, different OPS simulations will take a different number of function evaluations. Similar to what was done with absolute error, the number of function evaluations for each of 100 OPS simulations was summarized in terms of the sample mean and the 5th and 95th percentiles of the number of total function evaluations. It was found that the number of optimization function calls is independent of the number of inner loop distribution samples. For the simple sum problem, the average number of optimization function calls per OPS simulation was approximately

125. The number of optimization function calls per OPS simulation will vary in accordance with the characteristics of the black-box model. The range of the average number of total OPS function evaluations varies from 12494 for *t* = 100 to 131420 for t = 1000. For a one-minute black-box model, this translates into 8.7 days to 91.3 days of computation time. For most applications these computational costs are infeasible. Also, realistic models will be more complex than a simple sum, and they will therefore likely require a greater number of optimization function calls. In order to test the accuracy, of OPS for a feasible number of function evaluations, another set of 100 OPS simulations was run for *t* = 10. This resulted in an average absolute error of 0.1684 (a relative error of 1.87%) and 562 average total function evaluations—or 9.4 hours computation time for a one-minute black-box model.

Although both DLS and OPS might be too costly for many applications, OPS appears to be the better method. The relationship between accuracy and cost for the two methods is shown in Figure 10. For the data points taken, OPS can provide a more accurate approximation to $\underline{E}_e^P[Z]$ with less computational cost.

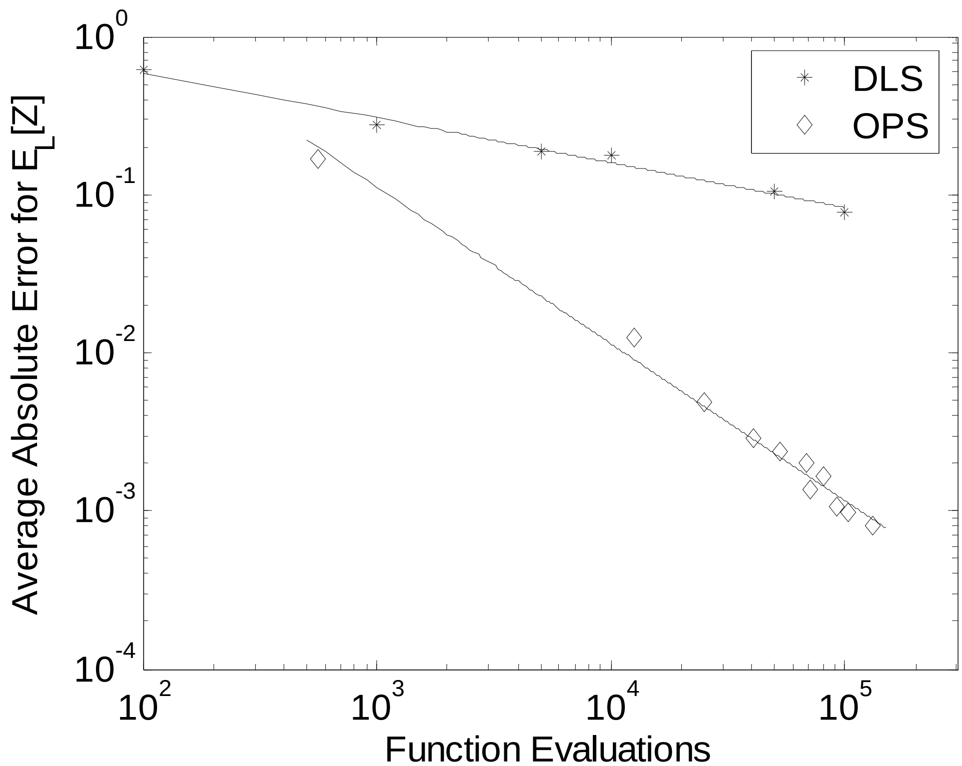

Figure 10. Comparison of OPS and DLS in terms of accuracy and cost.

5.1.3 Non-Parameterized Methods

If the inputs to the simple sum $Z = A + B$ are assumed to be non-parameterized p-boxes, then only DBC and PCS will be capable of approximating the exact expected value of the uncertain quantity, $E_e[Z]$,.

For the example problems in this thesis, DBC is implemented in Risk Calc. Although the bounds computed in Risk Calc are rigorous and are guaranteed to contain the true solution, a certain amount of discretization error will result from the stair-step approximation of the bounding curves. Since an exact solution is known—specifically,

$E_e[Z] \cong [7.80317, 15.19683]$, it is possible to study the size of the discretization error. Taking the expected value of the resultant bounding curves for the p-box $\boxed{Z}$, it is found that the DBC approximation is $E_{DBC}[Z] = [7.318, 15.682]$. As was expected, DBC does provide *conservative* bounds—that is, $E_e[Z] \subset E_{DBC}[Z]$, but the discretization error is significant—a 6.2% error for $\underline{E}[Z]$ and a 3.2% error for $\overline{E}[Z]$. Although discretization error can be reduced by increasing the number of p-box slices used in the analysis, doing so results in a more expensive computation. It is beyond the scope of this thesis to study the effects of DBC discretization error, but it is important to recognize that it can be significant even for simple problems.

In terms of cost, DBC benefits from its reliance on interval arithmetic. In terms of applicability, DBC is limited by its reliance on interval arithmetic. Since interval arithmetic includes interval sums, applicability is not an issue for the p-box sum problem studied here. To compute the resultant bounds on a sum of two intervals requires two function evaluations of the sum. The DBC method of Williamson and Downs [27] requires one interval evaluation for each of the discretized slices. Since Risk Calc uses 100 p-box slices, the total number of function evaluation used in determining the lower and upper bounds of the sum *Z* are $FE_{DBC} = 2 \times 100 = 200$. In general, if the number of function evaluations for a particular interval operation is *n*, and the number of discretization slices is *d*, then $FE_{DBC} = n \times d$. Since DBC is not a black box method like the other three approaches studied in this thesis, it is impossible to compare DBC fairly to the other methods in terms of computational cost.

PCS is a non-parameterized method that is compatible with black-box models. Like the other two black-box compatible methods, PCS is non-deterministic and therefore it was necessary to perform sets of 100 PCS simulations in order to achieve statistically significant approximations of accuracy and cost. Also, for simplicity, the inner loop optimization was replaced by the simple interval arithmetic operation for addition, $\left[\underline{z},\overline{z}\right]=\left[\underline{a}+\underline{b},\overline{a}+\overline{b}\right]$.

The results of 100 PCS simulations were summarized in terms of sample mean and the 5th and 95th percentiles of $\delta\underline{E}_{PCS}[Z]$. This was done for p-box sample sizes of 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000. The average absolute error ranged from 0.39648 (a 5.1% relative error) for 10 p-box samples to 0.0018872 (a 0.024% relative error) for 1000 p-box samples. Note that even with only 10 p-box samples, PCS results in a better approximation of $\underline{E}_e[Z]$ than DBC with 100 discretization slices. The effect of sample size on accuracy is summarized in Figure 11.

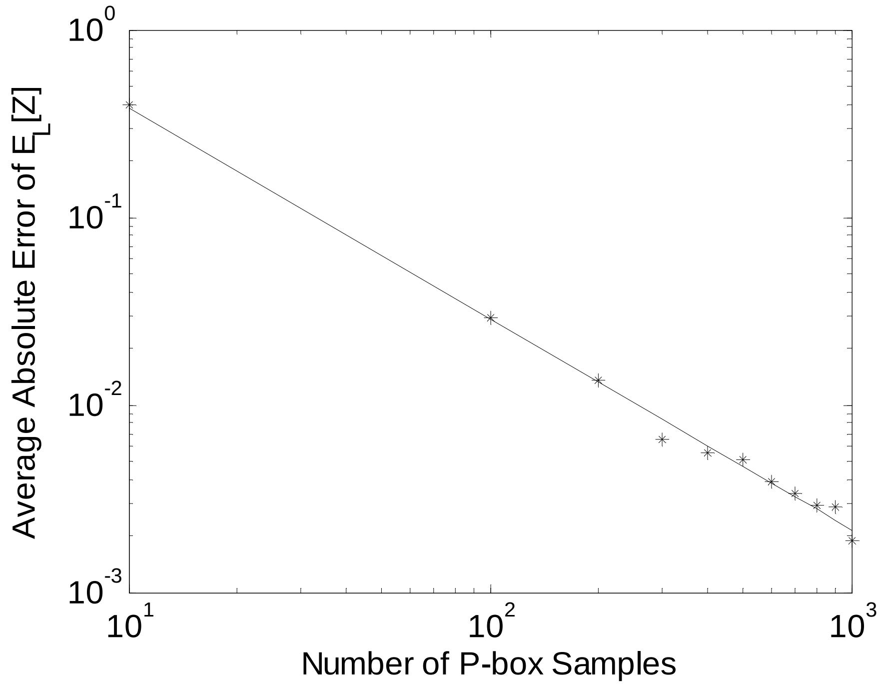

Figure 11. Average value of $\delta \underline{E}_{PCS}[Z]$ versus p-box sample size.

The computational cost of PCS can be calculated similarly to the computational cost of DBC. Specifically, $FE_{PCS} = n \times s$ where n is the number of function evaluations involved in a single black-box interval operation, and s is the number of p-box samples used in the PCS simulation. For the simple sum example, n = 2. Therefore the computational cost of the PCS simulations ranges from $FE_{PCS} = 20$ for s = 10 and $FE_{PCS} = 2000$ for s = 1000.

Although a direct comparison of the DBC and PCS methods is somewhat limited by the fact that only a single data point is available for DBC, it appears that there is reason to believe that PCS might actually be more cost effective for approximating

$\underline{E}_e[Z]$ than DBC. Indeed, for the simple sum problem, to achieve better average accuracy than DBC for 200 function evaluations, it is only necessary to perform 20 function evaluations in a PCS simulation.

Despite the fact that PCS is more easily applicable towards black box models, and apparently is more cost effective, DBC maintains one distinct advantage: the resultant DBC bounds are rigorous and are guaranteed to contain the true result. PCS results in bounds that are closer together than the theoretical expectations. Therefore, DBC is the conservative alternative and should be used if it is important not to underestimate the width of the resultant p-box for *Z*.

5.2 Example 2: Thermocouple junction analysis

In this section, we apply the four methods for PBA towards a transient heat transfer analysis of a spherical thermocouple junction. The thermocouple is designed to measure temperature in a gas stream. It is desired to know how long it will take the thermocouple to reach 99% of the gas stream temperature. Assuming uniform instantaneous temperature, negligible radiation loss, negligible conduction through the lead wires, and constant physical properties, the time to 99% of the gas stream temperature, $t_{0.99}$, can be calculated by

$$t_{0.99} = \frac{\rho D c}{6h} \ln\left(100 - 100\frac{T_i}{T_\infty}\right)$$

where the inputs are defined as follows

$\rho \equiv$ density of junction (kg/m^3)
$D \equiv$ diameter of thermocouple (m)
$c \equiv$ specific heat capacity (J/kg $\cdot$ K)
$h \equiv$ convection coefficient of thermocouple in gas (W/m$^2 \cdot$ K)
$T_i \equiv$ initial temperature of thermocouple (K)
$T_\infty \equiv$ temperature of gas (K)

The uncertainties associated with these inputs are modeled as follows

$$\rho \sim Normal(\mu = [8400, 8700], \sigma = [50, 300])$$
$$D \sim Normal(\mu = 7 \times 10^{-4}, \sigma = [0.1 \times 10^{-4}, 0.7 \times 10^{-4}])$$
$$c \sim Lognormal(\mu = 400, \sigma = 20)$$
$$h \sim Triang(\text{min} = [175, 200], \text{mod} = [280, 300], \text{max} = [380, 430])$$
$$T_i = [296, 300]$$
$$T_\infty \sim Lognormal(\mu = [470, 475], \sigma = [5, 10])$$

The inputs are treated as both parameterized and non-parameterized p-boxes in the following two sections

5.2.1 Parameterized Methods

For the case of parameterized inputs, an "exact" solution to the lower bound expectation was approximated by running OPS simulations for t = 10,000 probability loop samples. It was found that $\underline{E}_e^P[t_{0.99}] \cong 4.6438$. Error for both parameterized methods is computed as $\delta \underline{E}_M^P[Z] = \left| 4.6438 - \underline{E}_M^P[Z] \right|$. Again, 100 repetitions were performed for each specific DLS and OPS simulation.

DLS simulations were run for the sample sizes (s,t) of (10,10), (100,10), (500,10), (500,100), (1000,10), (1000,100), and (1000,500). Simulations were limited to sample size combinations with relatively large number of parameter samples s. This limitation was imposed since it was again found that the accuracy of the DLS method is more sensitive to the number of parameter samples than it is to the number of inner loop distribution samples. The accuracy of the DLS method ranged from an average absolute error of 0.188 (4.06% relative error) for 100 function evaluations to 0.075 (1.62% relative error) for 10,000 function evaluations. A summary of these results in comparison with the OPS methods is presented in Figure 12.

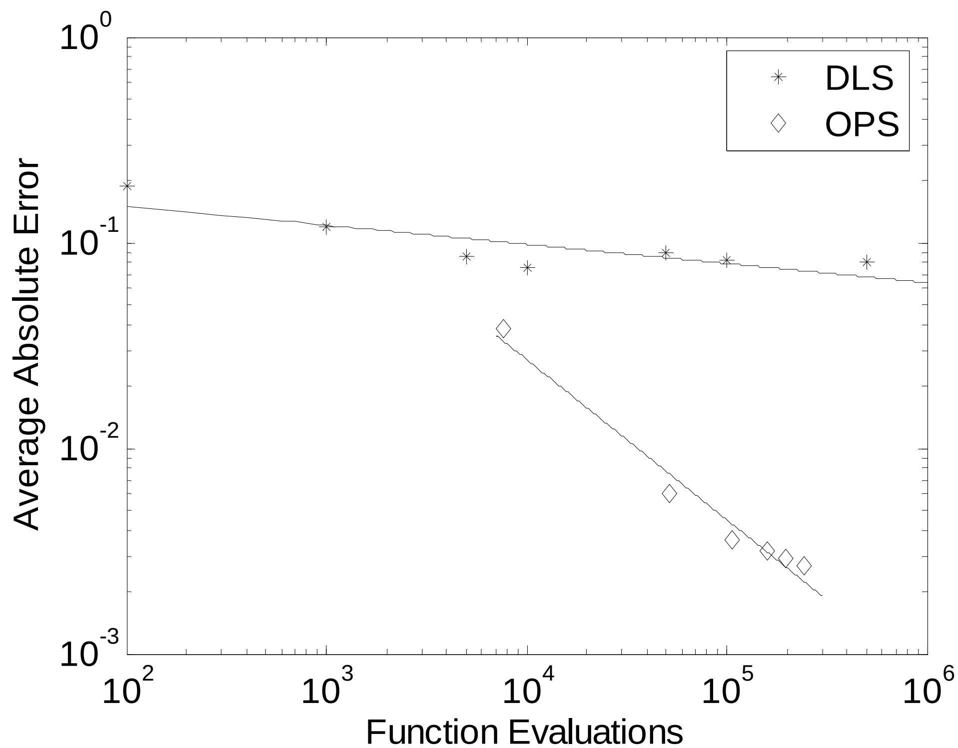

Figure 12. Comparison of DLS and OPS for thermocouple example.

The optimization in the parameter loop of the OPS simulations was run with the stopping criterion "TolFun" (the difference between successive objective function values) set to 1e-9. All other stopping criteria were maintained at their default settings. Like before, using OPS significantly decreased the computational cost of approximating

$\underline{E}_e^P[t_{0.99}]$. For t = 10 inner loop samples, the average absolute error for the OPS simulations was 0.0382 requiring an average of 7684 function evaluations. For a one-minute black-box model, this would require over 5 days of computation time. OPS is compared to DLS in Figure 12. It is shown that for the thermocouple problem, OPS converges towards the "exact" solution much faster than DLS.

5.2.2 Non-Parameterized Methods

For the case of non-parameterized inputs, an "exact" solution to the lower bound expectation was approximated by running PCS simulations for s = 5,000 p-box samples. It was found that $\underline{E}_e[t_{0.99}] \cong 4.5556$. Absolute error for both PCS and DBC were computed in relation to this value. For each specific PCS simulation, 100 trials were performed.

As was done for the simple sum example, a DBC simulation was performed using Risk Calc. It was found that $\underline{E}_{DBC}[t_{0.99}] = 4.3223$. This results in an absolute error of approximately 0.2333 (or a relative error of about 5.12%). DBC contains the true solution, but the discretization error for 100 slices is again significant.

For the PCS simulations, the optimizations in the inner loop (using fmincon) were run with all stopping criteria set to their default values. PCS simulations were performed for s = 10, 100, 200, 300, 400 and 500 p-box samples. A plot of the PCS method's error convergence is presented in Figure 13. Even for the smallest of the experimental sample sizes, s = 10, it was found that PCS results in a smaller error than DBC with 100 discretization slices. Specifically, the average absolute error for 10 p-box samples was found to be 0.0485 (a relative error of only 1.06%). This error corresponds to an average of 1210 function evaluations. For a one-minute black-box model, this would require 20 hours of computation time.

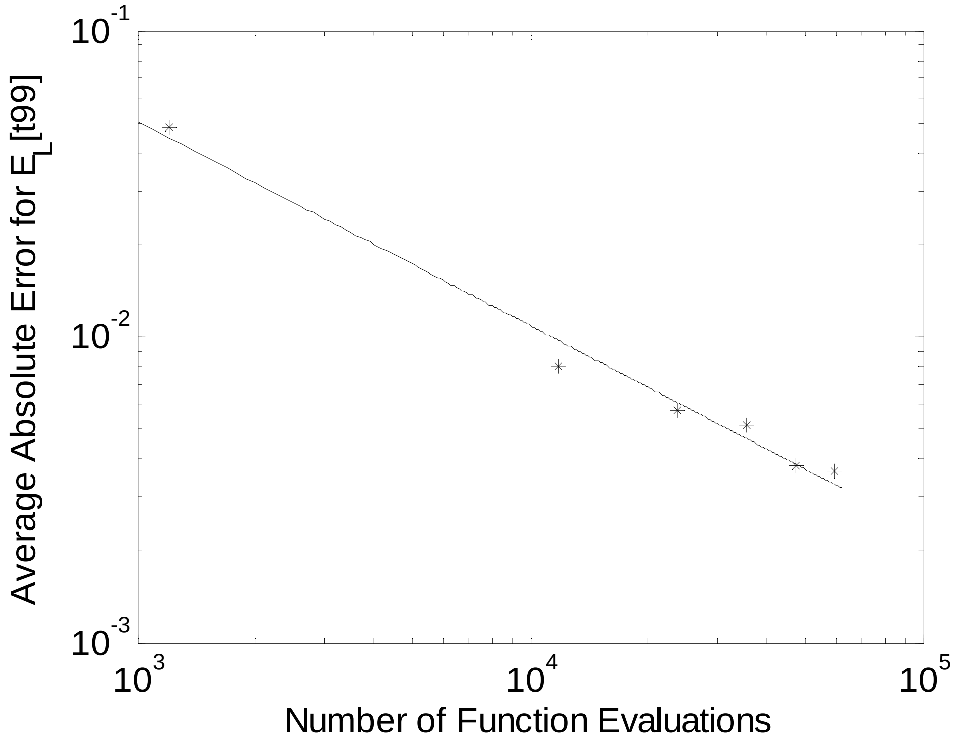

Figure 13. Performance of PCS simulations for thermocouple example.

5.3 Example 3: Design of a gearbox

In this section, we study a gearbox model in order to demonstrate the different elimination criteria in the context of a realistic design problem. The gearbox is intended for use in the drivetrain of an SAE Mini-Baja competition off-road vehicle. The basic configuration of the gearbox is shown in Figure 14. The objective of the design problem is to determine the geometries of the three gears such that expected utility of the design is maximized.

Although the decision policies demonstrated in this section are simply black box models, they are considerably more complex than the models studied in the previous two sections. By demonstrating that problems such as this can be solved using the black box methods proposed in Chapter 4, we will have found further evidence supporting Hypothesis 2. The gearbox design problem also gives support to Hypothesis 1. Specifically, it was determined that DBC was incapable of solving this problem due to the large number of repeated variables inherent in the problem. In other words, even if the mathematical operations of the model were known explicitly (i.e., the model were not a black box), it would still be infeasible to solve by DBC.

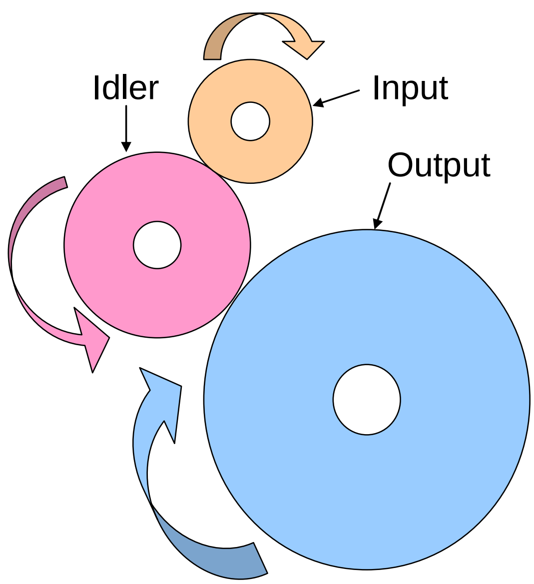

Figure 14. Gearbox configuration schematic

A summary of our problem formulation is presented in Figure 15. Utility is formulated as the expected dollar earnings from constructing and using the gearbox in Georgia Tech's Mini-Baja vehicle for a long-distance race. There are five design variables, and there are ten shared uncertain parameters, with uncertainty modeled as p-boxes, intervals, and precise probability distributions. The same problem was solved in the Master's Thesis of Rekuc [39] assuming interval uncertainty in the inputs.

Maximize

Expected Utility:

$$E[U] = E[U_t] \times R - E[U_C]$$

where

- $U_t = (\text{Prize Money}) * \left(1 - \frac{1}{1+e^{16-4t}}\right)$, a relationship determined by fitting a sigmoid function to past race finish times t
- R is the reliability of the gearbox (the probability that the gearbox completes the race)
- U_C is the cost of constructing the gearbox
- Prize Money is $1,000

Select

Gear Ratio $N_g = [0.5, 5]$ (torque ratio)

Input Gear Diameter $d_{in} = [1.5, 15]cm$

Idler Gear Diameter $d_{id} = [1.5, 15]cm$

Gear Width $w = [1.00, 8.75]cm$

Gear Module $M = [1.27, 8.75]mm/tooth$

Where

Performance is dependent on 10 uncertain system parameters:

Total Mass (kg), m ~ Normal([200,215],[18,20])

External Drag Coefficient (N/(m/s)^2), CD_e = [0.27,0.28]

Internal Drag Coefficient (N/rpm), CD_i = [0,0.0075]

Course Roughness Coefficient, K_c ~ Normal(3,0.5)

Bending Strength Factor, J = [0.38,0.4]

Gear Quality, Q_v ~ Normal([8.25,8.75],1)

Cost Error ($), $Cost_{err}$ = [-5,5]

Uncorrected Bending Strength (N/m^2), S'_{fb} ~ Normal([197,203]e6,[30,35]e6)

Uncorrected Contact Strength (N/m^2), S'_{fc} ~ Normal([197,203]e6,[30,35]e6)

Application Factor, K_a = [1.68,1.70]

Figure 15. Formulation of Mini-Baja Gearbox Problem.

The equation determining the cost of the gearbox (in dollars) is based on the combined volume of the three gears and the cost of material used. Specifically,

$$U_C = 1500 \frac{\pi}{4} w \left(d_{in}^2 + d_{id}^2 + d_{out}^2 \right)^{0.3} + Cost_{err}$$

where d_{out} is the diameter of the output gear.

The reliability of the gear box was set equal to the combined reliability of all three gears. That is, $R = R_{in} R_{id} R_{out}$. Each individual gear reliability is based on that gear's contact stress reliability and its bending stress reliability. For instance, the reliability of the input gear is found by $R_{in} = R_{c,in} R_{b,in}$. Contact stress and bending stress reliability were computed using a fitted exponential function derived from data points relating reliability to a reliability factor as found in Norton [63]. The functions used are

$$R_{b,x} = 1 - e^{-\frac{K_{R_x}}{0.175}}$$

and

$$R_{c,x} = 1 - e^{-\frac{C_{R_x}}{0.175}}$$

where $K_{R,x}$ and $C_{R,x}$ are the reliability factors for bending stress and contact stress, respectively, and x is used to denote a specific gear—i.e. input, idler, or output. Both $K_{R,x}$ and $C_{R,x}$ are calculated based on the equations and data described in Chapter 11 of Norton [63].

Finally, the time to finish was computed based on the gearbox geometry and the dynamics of the Mini Baja vehicle. Course finish time was computed as

$$t = \frac{D}{V_{\max}} + K_c \frac{V_{\max}}{a_{\max}}$$

where D is the course distance, $V_{\max}$ is the maximum velocity of the vehicle, K_c is the course roughness coefficient, and $a_{\max}$ is the maximum acceleration of the vehicle. The course distance, D, was set to 200 km, the distance for the endurance portion of the SAE Mini Baja competition. The course roughness coefficient is the uncertain quantity described in Figure 15. The dynamics of the vehicle were modeled by the following three equations

$$F_d = CD_e\left(\frac{\pi r_w}{30N_c N_g N_{CVT}}\right)^2 \omega_e^{\,2} + CD_i\omega_e + 4f_r mg$$
$$F_w = \left(\frac{N_c N_g N_{CVT}}{r_w}\right)\left((-2e-6)\omega_e^{\,2} + 0.0057\omega_e + 5.9977\right)$$
$$ma = F_w - F_d$$

where the variables are defined in Figure 15 or as follows

$F_d \equiv$ Drag Force (N)
$r_w \equiv$ Wheel Radius (m) $= 0.2794$
$N_c \equiv$ Chain Ratio $= 34/12$
$N_{CVT} \equiv$ CV Transmission Ratio $= [0.77, 3.83]$
$\omega_e \equiv$ Engine Angular Velocity (rpm)
$f_r \equiv$ Coefficient of Rolling Resistance $= 0.0015$
$g \equiv$ Gravitational Acceleration $= 9.81\ m/s^2$
$F_w \equiv$ Force to Wheels (N)

It is necessary to find $V_{\max}$ and $a_{\max}$ in order to find the course finish time. To find $V_{\max}$, set $F_d = F_w$ and N_{CVT} equal to its smallest ratio of 0.77. This equation is then solved for $\omega_{e,\max}$, the maximum engine speed on the Mini Baja vehicle. Then, the maximum velocity of the vehicle is found by

$$V_{\max} = \frac{\omega_{\max} c_w}{60 N_{cvt} N_g N_c}$$

where c_w is the circumference of the wheel. The maximum acceleration is found by

$$a_{\max} = \frac{F_{w,a_{\max}} - F_{d,a_{\max}}}{m}$$

where $F_{w,a_{\max}}$ and $F_{w,a_{\max}}$ are the forces acting on the vehicle at the maximum engine torque with N_{CVT} equal to its largest ratio of 3.83. It is assumed that the maximum engine torque is produced at 1000 rpm.

For this example, all uncertainty propagation was done using the OPS algorithm. OPS was appropriate since the problem assumes parameterized inputs and OPS is the most efficient of the available parameterized methods. The optimizations run in the OPS parameter loop were done using the MATLAB function fmincon. All stopping criteria were set to default values with the exception of "TolFun" (the difference between successive objective function values) which was set equal to 1e-9. It should be emphasized that the primary point of this example is to demonstrate the mechanics of the decision criteria for decision making under imprecise uncertainty. It was also of interest to show that solutions to realistic design problems can be found with reasonable computational time.

5.3.1 Demonstration of elimination criteria

The first part of the example demonstrates the reduction of the design space for the first design variable—the gear ratio. In this problem, it is ranges of values that are eliminated, rather than discrete alternatives. The initial problem statement specifies the design space of gear ratios is in the interval [0.5, 5.0]. In the first step of the sequential decision process, the DM seeks to reduce this interval as much as possible while retaining the most preferred—though currently unidentifiable—solution in the range.

We first consider the application of interval dominance by the DM. Figure 16 contains a plot of expected utility versus gear ratio. The two curves represent the upper and lower bounds on expected utility at a given gear ratio. In the plot, we can see that the highest point on the lower-bound, or the Γ-maximin solution, occurs at a ratio of about 1.5. The DM draws a horizontal line at the lower expected utility at this gear ratio. By the condition of interval dominance, any gear ratio with an upper-bound on expected utility that is below this line should be eliminated. For example, two expected utility intervals are indicated in Figure 16. The leftmost interval is located at the Γ-maximin solution. The DM compares all other decision alternatives to this interval. The Γ-maximin solution clearly dominates any of the expected utility intervals in the shaded regions. Therefore, the DM can eliminate both shaded regions from the design space.

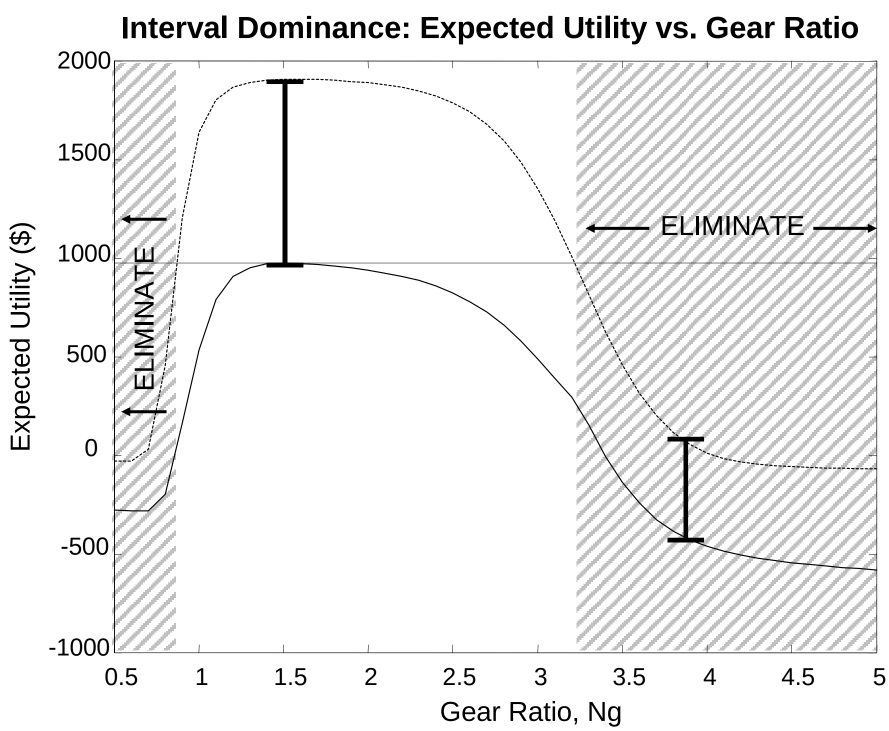

Figure 16. Elimination using interval dominance.

By taking into account the shared uncertainty of the model, we can make further eliminations in the design space using the maximality criterion. In theory, the DM would need to make pairwise comparisons between all possible designs throughout the entire design space to eliminate *all* decisions dominated under maximality. Of course, this is impossible for design problems with continuous design variables. In practice, a DM should therefore perform comparisons between a well-chosen discrete set of design alternatives across the entire design space.

In this example, the DM computes the bounds on the expected difference in utility between each gear ratio and the Γ-maximin gear ratio of 1.5. Recall that the maximality elimination criterion specifies that the DM should eliminate any alternative (in this case, gear ratio) with an upper bound on expected difference less than zero. Figure 17 contains a demonstration of maximality elimination over the range of gear ratios. The two curves represent lower and upper expected differences in utility.

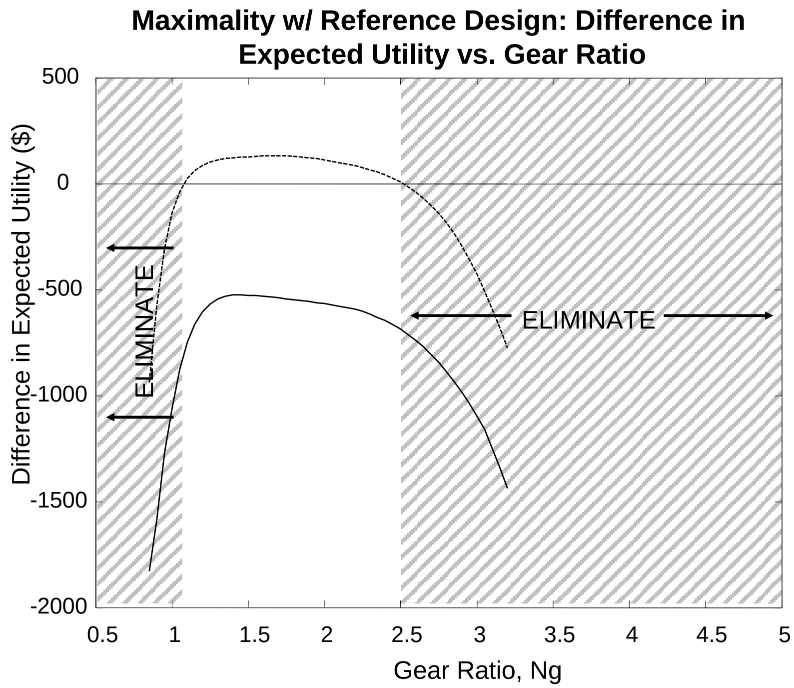

Figure 17. Eliminating using maximality.

The calculation of a difference in expected utility requires two alternatives—the one being tested, and a reference design. In order to increase the efficiency of elimination, we choose a detailed reference design [39]. This approach, part of a larger branch-and-bound strategy, takes one promising alternative and develops it in more detail, thereby reducing the imprecision for that alternative.

Any gear ratio that results in a negative upper bound on expected difference in utility will always perform worse than the reference design. The DM draws a horizontal line at an expected difference in utility of zero. The shaded regions correspond to gear ratios that are always dominated by designs with the reference gear ratio of 1.5. Therefore, the DM can eliminate all decision alternatives that fall in the shaded regions in Figure 17.

5.3.2 Sequential reduction of the design space

We conclude our examination of the gearbox example problem with a sequential decision-making process, sketched in Figure 18, to reduce the set of feasible designs. A single step in this process was described in the last section in which we reduced the design interval for the gear ratio. Now we proceed to reduce the set of non-dominated design alternatives sequentially through each of the remaining four design variables.

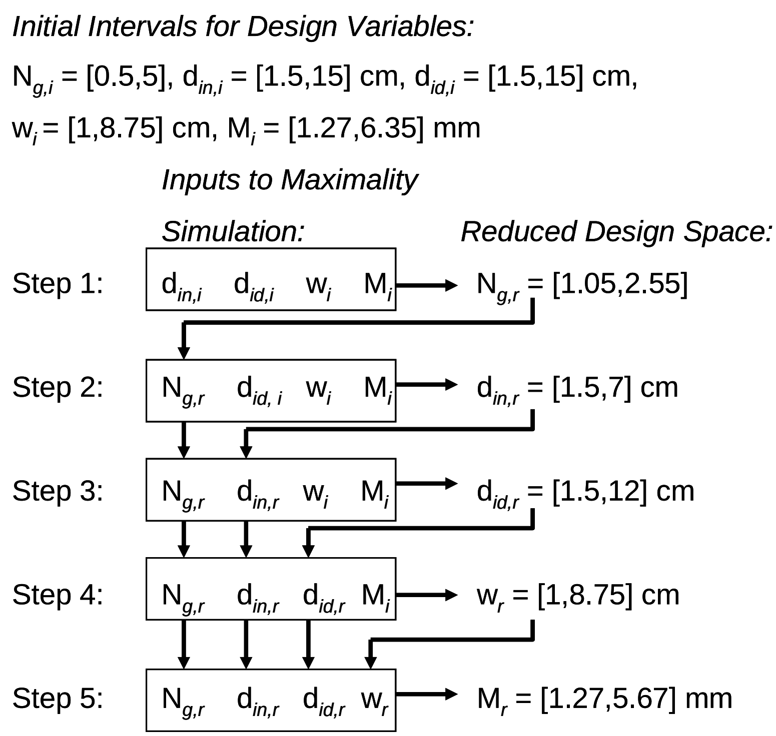

Figure 18. Sequential reduction process.

The advantage of sequential elimination is that with each reduction in the uncertainty associated with a single design variable, the uncertainty in expected utility is reduced. This, in turn, allows the DM to identify a larger set of dominated decision alternatives in the future decisions. The DM can therefore make greater eliminations with future design variables.

In step 1, the DM reduces the interval for gear ratio based upon the initial design uncertainty for the other four design variables. In step 2, the DM reduces the uncertainty for input gear diameter based upon the reduced uncertainty for gear ratio and the initial uncertainty for the other three design variables. The DM repeats this process sequentially until the design space for each design variable has been reduced. The right column contains the intervals representative of the reduced design space. This process can be repeated with this set of reduced intervals. Eventually, further repetition will converge to some irreducible set of intervals for the design variables. At this point, in order to reduce the design space further, either more information about the uncertain inputs must be gathered, or a final design decision must be determined using a semi-arbitrary decision policy such as Γ-maximin as described in Section 2.4.

5.3.3 Summary of example

This section has demonstrated the process of eliminating design alternatives in the context of gearbox design for a SAE Mini-Baja competition off-road vehicle. The problem is relatively rich in that it contains five decision variables and 10 uncertain parameters that illustrate a range of possible uncertainty characterizations. The goal of the example was to illustrate how decisions are made under imprecise uncertainty and that the OPS method is capable of performing the necessary computations without excessive computational cost. Hypothesis 2 is now better supported by the fact that the parameterized methods are capable of solving a design example with realistic complexity. Also, Hypothesis 1 is further supported by the fact that at least this realistic example is incompatible with the available methods for PBA.

CHAPTER 6: SUMMARY AND FUTURE WORK

The research presented in this thesis has addressed the problem of propagating imprecise uncertain quantities through engineering design models. This consisted of an evaluation of current computational methods for PBA as well as the presentation of three alternative computational methods for PBA. In this final chapter, the research is summarized. A special emphasis is placed on highlighting and elucidating the contributions made. The contributions are then critically evaluated with reference to the research questions and hypotheses presented in Chapter 1. Finally, several avenues for future research are indicated.

6.1 Summary

From the decision-based design perspective, decision making is the critical element of the design process. All practical decision making occurs under some degree of uncertainty. Subjective expected utility theory is a well-established method for decision making under uncertainty; however, it assumes that the DM can express his or her beliefs as precise probability distributions. For many reasons, both practical and theoretical, it can be beneficial to relax this assumption of precision. One possible means for avoiding this assumption is the use of imprecise probabilities. Imprecise probabilities are more expressive of uncertainty than precise probabilities, but they are also more computationally cumbersome. Probability Bounds Analysis (PBA) is a compromise between the expressivity of imprecise probabilities and the computational ease of modeling beliefs with precise probabilities. In order for PBA to be implemented in engineering design, it is necessary to develop appropriate computational methods for propagating probability boxes (p-boxes) through black box engineering models. The goal of this thesis is to explore two research questions related to this problem:

Research Question 1: *To what classes of engineering design problems can current computational methods in PBA be applied?*

Research Question 2: *Do there exist alternative computational methods for PBA that are compatible with the demands engineering design?*

In response to these research questions, the following hypotheses were proposed:

Hypothesis 1: *Current computational methods for PBA are only applicable to engineering design problems in which*

i) the entire mathematical model used for decision making is analytically defined as a sequence of basic arithmetical operations and elementary functions and

ii) no uncertain variables are repeated—i.e., all variables appear only once in the model expressions.

Hypothesis 2: *The PBA formalism can be made compatible with a broader class of engineering design problems by*

i) a double loop sampling algorithm using parameterized uncertain quantities,

ii) a modified double loop sampling algorithm called optimized parameter sampling, or

iii) a generalized version of probabilistic sampling called p-box convolution sampling.

In Chapter 2, the context was set for the problems posed by the research questions. Specifically, the design process was modeled as a sequence of decisions in which the space of design alternatives is systematically reduced. Each reduction in the space of design alternatives is accomplished according to some rational decision policy. A rational decision policy maps beliefs and preferences to some set of non-dominated decision alternatives. Therefore, in order to proceed through the design process, it is necessary to model mathematically the beliefs and preferences of the DM. Because of the reasons outlined previously, the p-box was chosen as the appropriate model of the DM's beliefs. Preferences were specified to be black box models mapping a realizable state of the world to a utility or a set of utilities (i.e., an imprecise utility function). Before specific aspects of computation were addressed, it was first necessary to comment on possibilities for decision policies in the presence of imprecise beliefs and preferences. In particular, the decision policies of interval dominance and maximality were discussed with examples for a single design variable. These decision policies naturally lead to *sets* of non-dominated decision alternatives. Chapter 2 concludes with a precise statement of a design decision problem in which beliefs are modeled by p-boxes. Essentially, the problem examined in this thesis is how to propagate p-boxes through engineering black box models.

Chapter 3 introduces the state of the art for PBA. The method of dependency bounds convolution (DBC) is both explained and criticized in terms of its applicability to engineering design. The specific criticisms resulting from this evaluation serve as a starting point for the development of alternative computational methods for PBA.

Three alternative computational methods are introduced in Chapter 4. Double loop sampling (DLS) is compatible with black box models but requires that the input beliefs be modeled as parameterized p-boxes. This means that DLS is considerably less general than DBC. A similar, but less computationally expensive, alternative to DLS is optimized parameter sampling (OPS). OPS is similarly not applicable for general p-box inputs. The final method introduced, p-box convolution sampling (PCS), is capable of propagating general p-boxes through black box models. Unlike DBC, the three methods introduced in Chapter 4 are not rigorous, that is they are not guaranteed to contain the true p-box output of the black box model. However, it can be argued that, in engineering design, rigor is always balanced by cost, and assured rigor is not necessarily the most valuable approach.

The trade-off between accuracy and cost is examined empirically in Chapter 5 through a series of numerical examples. First, a simple sum of normal p-boxes is evaluated using the four methods of DBC, DLS, OPS, and PCS. The methods that assume parameterized p-box inputs and those that assume general p-box inputs cannot be directly compared. The second example involves a transient heat transfer analysis of a thermocouple junction. The thermocouple example involves more uncertain inputs and requires the propagation of these inputs through a more mathematically complex model. The final example studied involved an actual design decision—the design of a gearbox for an off-road vehicle. The point of the gearbox example was to demonstrate an actual sequential design process and to illustrate that the methods can be used in design decision making. The three examples indicated that

1) OPS is less computationally expensive that DLS,

2) PCS is less computationally expensive that DBC for determining upper and lower expectations of the black box model output, and

3) OPS (and by extension DLS) are compatible with at least some realistic design problems.

These three conclusions indicate that Hypothesis 2 is correct. After much effort, it was also determined that DBC could not be effectively applied to the gearbox design problem due to the large number of repeated variables present in the model. This indicates that Hypothesis 1 is correct. In the next section, the correctness of the hypotheses will be argued for in greater detail.

6.2 Critical evaluation of hypotheses

Hypothesis 1 was given support in Sections 3.2, and 3.3. Part *i)* of Hypothesis 1 proposes that the current methods for PBA are only applicable for problems in which the decision model is "analytically defined as a sequence of basic arithmetical operations and elementary functions." In Section 3.2, DBC was introduced as the state-of-the-art of the available methods for PBA. Because DBC operations are only available for basic arithmetic and elementary functions [23], current computational methods for PBA are only compatible with analytically defined sequences of such operations. Part *ii)* of Hypothesis 1 proposes further that current methods can only handle models in which no uncertain variables appear more than once. As was explained in Section 3.3, DBC relies on the methods of interval arithmetic. Since interval arithmetic becomes over-conservative in the presence of repeated variables [56], DBC becomes over-conservative in the presence of repeated variables. Depending on the degree of over-estimation of the resultant p-box bounds, DBC results might be entirely useless or unreliable in the presence of repeated variables.

Hypothesis 2 was given theoretical support in Chapter 4 and numerical support in Chapter 5. Three alternative computational methods for PBA were presented in Chapter 4. These were the three methods proposed as alternatives in Hypothesis 2. It was explained in Chapter 4 that these three methods do indeed approximate the quantities that we PBA in design is looking for—specifically, lower and upper expectations of some resultant quantity. Therefore, alternative computational methods for PBA do exist. Furthermore, these methods are capable of functioning within black box decision models. The argument that these methods are compatible with the demands of engineering design was given support by the numerical examples in Chapter 5. Of the parameterized methods, OPS appears to be more efficient. Of the non-parameterized methods, PCS appears to be more efficient. The gearbox design example showed that for at least one realistic design problem that OPS is computationally feasible. Therefore, the statement in Hypothesis 2 that alternative methods can make PBA compatible with a broader class of engineering design problems is given empirical support. Further work remains to be done to determine the limits of the three proposed black box methods.

6.3 Future work

The three methods presented in this thesis have only been implemented in their most basic form. Possibilities exist for improving the efficiency of these methods, and these possibilities point in several new directions for future research.

Although the parameterized methods, DLS and OPS, are only applicable to a less general class of p-boxes, they are still useful in problems where uncertainty should be modeled with parameterized p-boxes. Therefore, it might be valuable to examine possible areas of improvement for the two parameterized methods. Three general areas for improvement can be identified.

First, since OPS must solve a global optimization problem, it would be valuable to determine what optimization algorithms are most appropriate for using OPS in engineering design. Although using multiple starting points proved to be effective for the numerical examples in this paper, perhaps a more efficient strategy is available.

A second possibility for improving the parameterized methods is to find ways of parameterizing more general p-boxes. As was discussed previously, parameterized p-boxes are special cases of general p-boxes. It was argued that parameterized p-boxes arise frequently in practice, but not all realistic belief states can be easily represented as parameterized p-boxes. For instance, Dempster-Shafer structures are general p-boxes that result from the methods of evidence theory [64, 65], but there appears to be no straightforward way in which to model a Dempster-Shafer structure as a distribution with imprecise parameters. If any version of DLS or OPS is to be generally applicable, a means of parameterizing more general p-boxes needs to be discovered.

A third possibility for improving the parameterized methods concerns modeling dependence between the uncertain inputs. One of the primary advantages of DBC is that it allows for the determination of theoretical best-possible bounds on the resultant p-box under any state of dependence between the uncertain quantities. By comparison, the black box methods presented in this thesis assume independence between the uncertain quantities. This is in violation of the problem statement presented in Section 2.5. It is therefore desirable to modify the DLS and OPS methods such that they provide approximations to the best-possible bounds in case of unknown dependence.

Based on the results presented in this thesis, the PCS method is the most promising of the black box methods since it is already applicable to fully general p-boxes. We will close this chapter with three possible avenues for improving the PCS method.

First, there are many opportunities for improving the efficiency of the inner loop of PCS. Recall that the inner loop uses an optimizer to find the bounds on a function of several intervals. For many applications, additional information about the black box model can be used to speed up this optimizer. For instance, if it is known that the black box model is monotonic with respect to a certain set of input variables, then the dimensionality of the inner loop optimizer can be reduced by the number of monotonic inputs. Also, some information about the sensitivity of the model to the model inputs can be used to reduce the complexity of the optimizer. Specifically, if it is found that the black box model is sufficiently insensitive to a given input, then that input can be effectively eliminated from the analysis, and the dimensionality of the optimization problem can be further reduced. Another means of improving efficiency of PCS is to choose the optimizer starting point intelligently. A very simple means of doing this is to start the optimizer from the solution of the previous optimization.

Second, the efficiency of the p-box sample selection can be improved. In this paper, all sampling was done using Latin Hypercube Sampling. More efficient sampling procedures such as Hammersley sequence sampling [66] could reduce the number of p-box samples required to achieve a desired accuracy. In addition, importance sampling techniques could be used to sample the multidimensional input space more intelligently.

A final area of improvement for the PCS method involves the utilization of dependency information. In this thesis, it was assumed that the inputs were independent, but in realistic problems, this assumption is not always justified. In fact, *no knowledge* of dependence is available in many problems. A strong advantage of the DBC method is that it can compute the tightest possible p-box bounds for the case of unknown dependence. It would be desirable to augment the PCS method for it to be applicable in cases of unknown dependence.

REFERENCES

[1] Mistree, F., Smith, W. F., Bras, B. A., Allen, J. K., and Muster, D., 1990, "Decision-Based Design. A Contemporary Paradigm for Ship Design," *1990 SNAME Annual Meeting, Oct 31-Nov 3 1990*, Publ by Soc of Naval Architects & Marine Engineers, Jersey City, NJ, USA, San Francisco, CA, USA, **98**, pp. 565-597.

[2] Hazelrigg, G. A., 1998, "A Framework for Decision-Based Design," *Journal of Mechanical Design*, **120**(4), pp. 653-658.

[3] Marston, M., Allen, J. K., and Mistree, F., 2000, "The Decision Support Problem Technique: Integrating Descriptive and Normative Approaches in Decision Based Design," *Engineering Valuation and Cost Analysis*, **3**, pp. 107-129.

[4] Parry, G. W., 1996, "The Characterization of Uncertainty in Probabilistic Risk Assessment of Complex Systems," *Reliability Engineering and System Safety*, **54**(2-3), pp. 119-126.

[5] Kreinovich, V., Ferson, S., Ginzburg, L., Schulte, H., Barry, M., and Nguyen, H., 1999, "From Interval Methods of Representing Uncertainty to a General Description of Uncertainty," *International Conference on Information Technology*, McGraw-Hill, Bhubaneswar, India, pp. 161-166.

[6] Muhanna, R. L., and Mullen, R. L., 2004, "Interval Methods for Reliable Computing," *Engineering Design Reliability Handbook*, Nikolaidis, E., Ghiocel, D. M., and Singhal, S. eds., CRC Press, New York, NY, pp. 12.11-12.24.

[7] Aughenbaugh, J. M., and Paredis, C. J. J., 2006, "Why Are Intervals and Imprecision Important in Engineering Design?" *Reliable Engineering Computing Workshop*, Savannah, GA, USA.

[8] de Finetti, B., 1980, "Foresight. Its Logical Laws, Its Subjective Sources (Translated)," *Studies in Subjective Probability*, Kyburg, H. E. and Smokler, H. E. eds., E. Krieger Publishing Company.

[9] Savage, L. J., 1972, *The Foundation of Statistics*, Dover Publications, New York.

[10] von Neumann, J., and Morgenstern, O., 1944, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ.

[11] Berger, J. O., 1985, *Statistical Decision Theory and Bayesian Analysis*, Springer.

[12] Walley, P., 1991, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, London.

[13] Keeney, R., and Raiffa, H., 1993, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, Cambridge, UK.

[14] Scott, M. J., 2004, "Utility Methods in Engineering Design," *Engineering Design Reliability Handbook*, eds., CRC Press, New York.

[15] Ferson, S., and Ginzburg, L., 1996, "Different Methods Are Needed to Propagate Ignorance and Variability," *Reliability Engineering and System Safety*, **54**(2-3).

[16] Hofer, E., 1996, "When to Separate Uncertainties and When Not to Separate," *Reliability Engineering & System Safety*, **54**(2-3), pp. 113-118.

[17] Winkler, R. L., 1996, "Uncertainty in Probabilistic Risk Assessment," *Reliability Engineering & System Safety*, **54**(2-3), pp. 127-132.

[18] Aughenbaugh, J. M., and Paredis, C. J. J., 2005, "The Value of Using Imprecise Probabilities in Engineering Design," *ASME DETC 2005*, Long Beach, California, USA.

[19] Howard, R. A., 1988, "Decision Analysis: Practice and Promise," *Management Science*, **34**(6), pp. 679-695.

[20] Clemen, R. T., 1996, *Making Hard Decisions: An Introduction to Decision Analysis*, Duxbury Press, New York.

[21] Ferson, S., and Donald, S., 1998, "Probability Bounds Analysis," *International Conference on Probabilistic Safety Assessment and Management (PSAM4)*, Springer-Verlag, New York, NY, pp. 1203-1208.

[22] Ferson, S., and Ginzburg, L. R., 1996, "Different Methods Are Needed to Propagate Ignorance and Variability," *Reliability Engineering & System Safety*, **54**(2-3), pp. 133-144.

[23] Ferson, S., and Hajagos, J., 2004, "Arithmetic with Uncertain Numbers: Rigorous and (Often) Best Possible Answers," *Reliability Engineering & System Safety*, **85**(1-3), pp. 135-152.

[24] Ferson, S., Ginzburg, L., Kreinovich, V., Nguyen, H. T., and Starks, S. A., 2002, "Uncertainty in Risk Analysis: Towards a General Second-Order Approach Combining Interval, Probabilistic, and Fuzzy Techniques," *2002 IEEE International Conference on Fuzzy Systems: FUZZ-IEEE'02, May 12-17 2002*, Institute of Electrical and Electronics Engineers Inc., Honolulu, HI, United States, **2**, pp. 1342-1347.

[25] Levi, I., 1980, *The Enterprise of Knowledge, an Essay on Knowledge, Credal Probability, and Chance*, MIT Press, Cambridge, MA.

[26] Rekuc, S. J., Aughenbaugh, J. M., Bruns, M., and Paredis, C. J. J., 2006, "Eliminating Design Alternatives Based on Imprecise Information," *Society of Automotive Engineering World Congress*, Detroit, Michigan.

[27] Williamson, R. C., and Downs, T., 1990, "Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds"," *International Journal of Approximate Reasoning*, **4**, pp. 89-158.

[28] Fishman, G. S., 1996, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer, New York.

[29] Dubois, D., and Prade, H., 1988, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press.

[30] Zadeh, L. A., 1965, "Fuzzy Sets," *Information and Control*, **8**, pp. 338-353.

[31] Cooke, R., 2004, "The Anatomy of the Squizzel - the Role of Operational Definitions in Representing Uncertainty," *Reliability Engineering & System Safety*, **85**(1-3), pp. 313.

[32] Ferson, S., 2002, "RAMAS Risk Calc," 4.0 ed., Lewis Publishers, New York.

[33] Berleant, D., and Zhang, J., 2004, "Representation and Problem Solving with Distribution Envelope Determination (Denv)," *Reliability Engineering & System Safety*, **85**(1-3), pp. 153-168.

[34] Berleant, D., 1993, "Automatically Verified Reasoning with Both Intervals and Probability Density," *Interval Computations*(2), pp. 48-70.

[35] Berleant, D., and Goodman-Strauss, C., 1998, "Bounding the Results of Arithmetic Operations on Random Variables of Unknown Dependency Using Intervals," *Reliable Computing*, **4**(2), pp. 147-165.

[36] Berleant, D., Xie, L., and Zhang, J., 2003, "Statool: A Tool for Distribution Envelope Determination (Denv), an Interval-Based Algorithm for Arithmetic on Random Variables," *Reliable Computing*, **9**(2), pp. 91-108.

[37] Troffaes, M. C. M., 2004, "Decision Making with Imprecise Probabilities: A Short Review," in *The Society for Imprecise Probability Theory and Applications*, vol. 2, pp. 4-7.

[38] Law, A. M., and Kelton, D. W., 2000, *Simulation Modeling and Analysis*, McGraw-Hill, St. Louis.

[39] Rekuc, S. J., 2005, *Eliminating Design Alternatives under Interval-Based Uncertainty*, Thesis, Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

[40] Sobek, D. K. I., and Ward, A. C., 1996, "Principles from Toyota's Set-Based Concurrent Engineering Process," *ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, California.

[41] Sobek, D. K., Ward, A., and Liker, J. K., 1999, "Toyota's Principles of Set-Based Concurrent Engineering," *Sloan Management Review*.

[42] Springer, M. D., 1979, *The Algebra of Random Variables*, John Wiley & Sons, New York.

[43] Hoffman, F. O., and Hammonds, J. S., 1994, "Propagation of Uncertainty in Risk Assessments: The Need to Distinguish between Uncertainty Due to Lack of

Knowledge and Uncertainty Due to Variability," *Risk Analysis*, **14**(5), pp. 707-712.

[44] Williamson, R. C., 1989, *Probabilistic Arithmetic*, Thesis, Electrical Engineering, University of Queensland.

[45] Helton, J. C., and Davis, F. J., 2003, "Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems," *Reliability Engineering & System Safety*, **81**, pp. 23-69.

[46] Hofer, E., Kloos, M., Krzykacz-Hausmann, B., Peschke, J., and Woltereck, M., 2002, "An Approximate Epistemic Uncertainty Analysis Approach in the Presence of Epistemic and Aleatory Uncertainties," *Reliability Engineering & System Safety*, **77**(3), pp. 229-238.

[47] Berleant, D., and Cheng, H., 1998, "A Software Tool for Automatically Verified Operations on Intervals and Probability Distributions," *Reliable Computing*, **4**(1), pp. 71-82.

[48] Regan, H. M., Ferson, S., and Berleant, D., 2004, "Equivalence of Methods for Uncertainty Propagation of Real-Valued Random Variables," *International Journal of Approximate Reasoning*, **36**(1), pp. 1-30.

[49] Ferson, S., 2000, "Probability Bounds Analysis Solves the Problem of Incomplete Specification in Probabilistic Risk and Safety Assessments," *Ninth Conference on Risk-Based Decisionmaking in Water Resources*, Santa Barbara CA, USA.

[50] Tucker, W. T., and Ferson, S., 2003, "Probability Bounds Analysis in Environmental Risk Assessment," Applied Biomathematics, Setauket, New York, http://www.ramas.com/pbawhite.pdf.

[51] Ferson, S., Hajagos, J., and Tucker, W. T., 2004, "Probability Bounds Analysis Is a Global Sensitivity Analysis," *International Conference on Sensitivity Analysis of Model Output (SAMO)*, Sante Fe, New Mexico.

[52] Yager, R. R., 1986, "Arithmetic and Other Operations on Dempster-Shafer Structures," *International Journal of Man-Machine Studies*, **25**, pp. 357-366.

[53] Nelsen, R. B., 1999, *An Introduction to Copulas*, Springer, New York.

[54] Hoeffding, W., 1940, "Scale-Invariant Correlation Theory," *Collected Works of Wassily Hoeffding*, Fisher, N. I. and Sen, P. K. eds., Springer-Verlag, New York.

[55] Frechet, M., 1951, "Sur Les Tableaux De Corelation Dont Les Marges Sont Donnees," *Ann. Univ. Lyon, Sect. A*, **9**, pp. 53-77.

[56] Moore, R., 1979, *Methods and Applications of Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia.

[57] Ferson, S., Nelsen, R. B., Hajagos, J., Berleant, D., Zhang, J., Tucker, W. T., Ginzburg, L., and Oberkampf, W. L., 2004, "Dependence in Probabilistic Modeling, Dempster-Shafer Theory, and Probability Bounds Analysis," Sandia National Laboratories, Albuquerque

[58] Trejo, R., and Kreinovich, V., 2001, "Error Estimations for Indirect Measurements: Randomized Vs. Deterministic Algorithms for 'Black-Box' Programs," *Handbook on Randomized Computing*, Rajasekaran, S., Pardalos, P., Reif, J., and Rolim, J. eds., Kluwer, pp. 673-729.

[59] Kreinovich, V., and Ferson, S., 2004, "A New Cauchy-Based Black-Box Technique for Uncertainty in Risk Analysis," *Reliability Engineering & System Safety*, **85**, pp. 267-279.

[60] Pinter, J. D., 1996, *Global Optimization in Action*, Kluwer Academic Publishers, Dordrecht.

[61] Horst, R., Pardalos, P. M., and Thoai, N. V., 2000, *Introduction to Global Optimization*, Kluwer, Dordrecht.

[62] Hansen, E., and Walster, G. W., 2004, *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., New York.

[63] Norton, R. L., 2000, *Machine Design: An Integrated Approach*, Prentice-Hall Inc., Upper Saddle River, N.J.

[64] Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D. S., and Sentz, K., 2002, "Constructing Probability Boxes and Dempster-Shafer Structures," SAND2002-4015, Sandia National Laboratories, Albuquerque, New Mexico, http://www.sandia.gov/epistemic/Reports/SAND2002-4015.pdf.

[65] Ferson, S., Hajagos, J., Myers, D. S., and Tucker, W. T., 2005, "Constructor: Synthesizing Information About Uncertain Variables," Applied Biomathematics for Sandia National Laboratories, http://www.ramas.com/construct.pdf.

[66] Kalagnanam, J. R., and Diwekar, U. M., 1997, "An Efficient Sampling Technique for Off-Line Quality Control," *Technometrics*, **39**(3), pp. 308-319.